Міністерство освіти і науки України
Відокремлений структурний підрозділ
«Ковельський промислово-економічний фаховий коледж
Луцького національного технічного університету»

«Web-технології та Web-дизайн»

Методичні вказівки до практичних занять для здобувачів освітньо-професійного ступеня фаховий молодший бакалавр IV-курсу
спеціальность 122 Комп’ютерні науки
денної форми здобуття освіти

Ковель 2025

УДК 004.41 (07)
В-26

До друку _____ Надія КОВАЛЬЧУК. Перший проректор. Голова Навчально-методичної ради ЛНТУ

Затверджено Навчально-методичною радою ЛНТУ,

протокол №_____ від « ___» _____ 2025 року

Затверджено до видання методичною радою ВСП «КПЕФК ЛНТУ»

протокол №_____ від « ___» _____ 2025 року

Голова методичної ради ВСП «КПЕФК ЛНТУ» _____ Ігор ІЛЮШИК

Розглянуто і схвалено на засіданні циклової комісії з комп'ютерних наук ВСП«КПЕФК ЛНТУ»,

Протокол № ___ від «____» _____ 2025 року.

Голова циклової комісії_____Олександр ПРИСАДА

Електронна копія друкованого видання передана до книжкового фонду бібліотеки ВСП «КПЕФК ЛНТУ»
Завідувачка бібліотеки_____ Неля ТЕЛЮЧИК

Укладач: _____Людмила МЕЛЕЩУК, викладач ВСП «КПЕФК ЛНТУ»
(підпис)

Рецензент:_____ Тетяна СЕЛІВОНЧИК, канд. техн. наук, доцент, директорка (підпис) ВСП «КПЕФК ЛНТУ»

Відповідальний за випуск: _____ Леся ПРОКОПЧУК, методист ВСП «КПЕФК ЛНТУ»
(підпис)

В-26 Методичні вказівки до практичних занять для здобувачів освітньо-професійного ступеня фаховий молодший бакалавр IV курсу спеціальності 122 Комп'ютерні науки денної форми здобуття освіти / уклад. Л.В. МЕЛЕЩУК. – Ковель : ВСП «КПЕФК ЛНТУ», 2025. – 154.

Методичні вказівки до практичних занять, призначені висвітленню теоретичних та практичних аспектів вебтехнологій та вебдизайну, а також містять типові приклади вебпрограмуваня. Розглянуто різні принципи побудови та функціонування вебсайтів, використання сучасних вебтехнологій та мов вебпрограмування, будування вебсторінок із заданими характеристиками і алгоритмами функціонування.

Методичні вказівки призначені для здобувачів освіти спеціальності 122 Комп'ютерні науки денної форми здобуття освіти ВСП «КПЕФК ЛНТУ».

ЗМІСТ

ВСТУП

Методичні вказівки до практичних занять, призначені висвітленню теоретичних та практичних аспектів вебтехнологій та вебдизайну, а також містять типові приклади вебпрограмуваня. Розглянуто різні принципи побудови та функціонування вебсайтів, використання сучасних вебтехнологій та мов вебпрограмування, будування вебсторінок із заданими характеристиками і алгоритмами функціонування.

У методичних вказівках також розглядається широкий спектр протоколів, стандартів і технологій, що мають безпосереднє відношення до розробки вебзастосувань.

Методичні вказівки використовуються здобувачами освіти спеціальності 122 Комп'ютерні науки при здачі практичних робіт, підготовці до складання іспиту, при узагальненні та систематизації отриманої інформації, при закріпленні теоретичного матеріалу, а також при вивченні матеріалів інших освітніх компонентів, пов'язаних з інтернет програмуванням.

Даний курс базується на вивченні наступних тематичних напрямків:

- Основи Web-дизайну та графіка для Web;
- Теоретичні і практичні основи HTML 5.0;
- Теоретичні і практичні основи CSS;
- Базовий синтаксис JavaScript;
- Технологія AJAX;
- Мова програмування PHP.

Метою освітнього компонента є забезпечення теоретичної підготовки здобувачів освіти з основ вебтехнологій, вебдизайну та вебпрограмування, а також набуття практичних навичок з проєктування, розробки та дизайну вебсайтів і вебдодатків на боці клієнта/сервера. Програма розрахована на здобувачів, які засвоїли освітні компоненти «Основи програмування та алгоритмічні мови», «Алгоритми та структури даних», «Бази даних», «Комп'ютерні мережі», «Операційні системи».

У результаті вивчення освітнього компонента здобувач освіт повинен знати:

- ✓ основні принципи створення Web-сторінок;
- ✓ засоби макетування та верстки засобами HTML;
- ✓ засоби макетування та верстки засобами CSS;
- ✓ правила запису HTML, CSS та PHP кодів;
- ✓ функції, що обробляють строкові данні в PHP;
- ✓ цикли PHP;
- ✓ основи роботи серверних програм;
- ✓ історію мови програмування PHP та програм, які її супроводжують (Apache, PHP Edit, Denver, Openserver);
- ✓ структуру динамічного сайту.
- ✓ схему роботи мереж клієнт/сервер.

У результаті вивчення освітнього компонента здобувач освіт повинен вміти:

- ✓ створювати окремі програми для динамічних сайтів на базі HTML5.0, стилів CSS та PHP;
- ✓ працювати з розміткою HTML;
- ✓ працювати зі стилями CSS;
- ✓ обробляти строкові дані;
- ✓ створювати програми, що можуть управляти зовнішнім виглядом сайту;
- ✓ інсталювати програми PHP Edit, OpenServer;
- ✓ розробляти алгоритми та програми на мові програмування PHP;
- ✓ оперувати базами даних на сервері;

У процесі викладання освітнього компонента необхідно навчати зацікавленість здобувачів застосовувати комп'ютерні технології в обраній спеціальності, організовувати інформаційні дані різних видів, широко налагоджувати міждисциплінні зв'язки, особливо зі спецдисциплінами.

Практичне заняття №1

Тема. Встановлення Visual Studio Code, налаштування. Створення першої web-сторінки

Мета: навчитися встановлювати VS та створювати примітивні web-сторінки з використанням мови опису HTML.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Теоретичні відомості

1. Встановлюємо Visual Studio Code – це професійний «легкий» редактор для створення коду програм різних мов програмування.

2. Встановлюємо мовний пакет та «html css support» і «live Server».

Редактор VSC є програмною оболонкою до якої можна додавати різні програмні розширення. Назвемо ці розширення плагіни. Вони покращують певні сторони роботи програми. Встановимо два плагіна: «html css support» і «live Server».

- *html css support* - полегшує набір html і css код при створенні web-сторінок.
- *live Server* - допомагає переглядати виконання програмного коду у браузері.

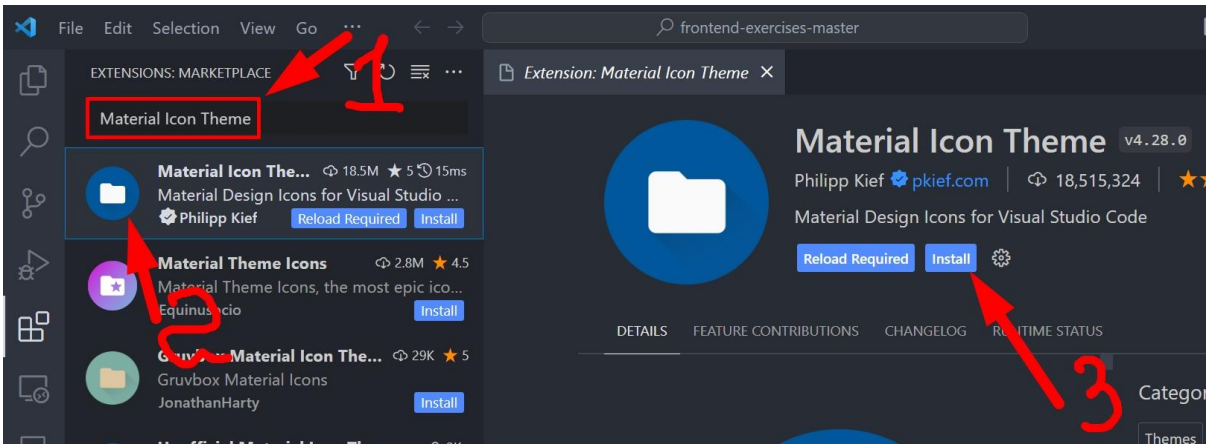
3. Встановлення додаткових плагінів та налаштувань.

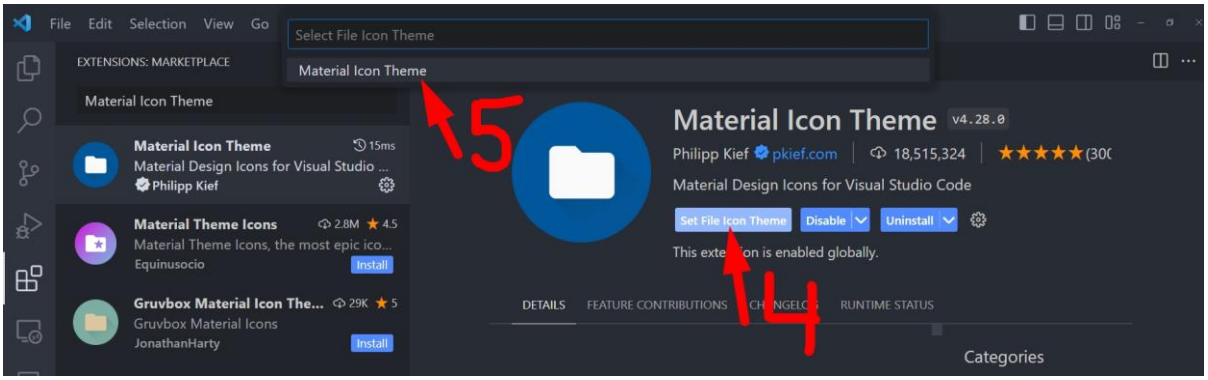
Для покращеної роботи VS Code необхідно встановити ще ряд інший плагінів:

(встановлення відбувається таким самим способом як ви і робили до цього? тобто у пошуку пишете назву плагіну, зазвичай він з'являється першим у рядку, та натискаєте кнопку Install)

- **auto rename tag**
- **auto close tag**
- **auto complete tag**
- **css peek**
- **css navigation**
- **html/css/javascript snippets**
- **IntelliSense for CSS class names in**
- **path intellisenss**
- **Prettier - Code formatter**
- **eslint**

А також для покращеного відображення програми, а саме різноманітних значків папок, файлів у ній потрібно встановити на налаштувати плагін: **Material Icon Theme,** дотримуйтеся фото-інструкції:





4. Добавляємо папку у робочу область.

Вся інформація яку ви створюєте і якою користуєтесь має бути збережена у відповідних папках. Це робиться для того, щоб можна було швидко знайти ту чи іншу програму. Все має бути структуровано. Ваша робоча папка, у яку ви будете поміщати файли з своїми програмами, спочатку створюється на вашому комп'ютері. Така папка має бути додана і до редактора VSC, це для синхронізації роботи Windows і VSC.

На диску Z:\ твого комп'ютера створи папку «Прізвище». У паці» Прізвище» створи папку «html css». Додай папку «html css» у робочу область програми VSC.

5. Що таке WEB-сторінка?

Вебсторінка - це інформаційний ресурс, який доступний в мережі Інтернет і проглядається за допомогою браузера. Вона створюється як текстовий файл в якому є текст і теги, а формат такого файлу - html.

HTML в перекладі з англійської - мова гіпертекстової розмітки. Так само вона дозволяє вставляти в документи зображення, звук, відео і т.д. HTML - це тегова мова розмітки документа. Будь-який документ на мові HTML являє собою набір елементів, при цьому початок і кінець кожного елемента позначаються спеціальними позначками - *тегами*.

Теги - це команди браузера і з їх допомогою створюється структура вебсторінки - параграфи, розділи, абзаци, списки, малюнки, таблиці, колонтитули, індекси, зміст і т.д. В середині кожного блоку можна змінювати шрифт тексту, його формат, колір та інше.

6. Структура WEB-сторінки:

<!DOCTYPE html > - *цей тег вказує браузеру, що це є вебсторінка. Тоді браузер розуміє як правильно її читати в показувати людям.*

<html> – *це тег який містить в собі всю вебсторінку.*

<head> – *цей тег призначений для зберігання інших елементів, мета яких — допомогти браузеру в роботі з даними*

<meta charset=«UTF-8»>

<meta http-equiv=«X-UA-Compatible» content=«IE=edge»>

<meta name=«viewport» content=«width=device-width, initial-scale=1.0»>– *це метатеги, вони використовуються для зберігання інформації призначеної для браузерів і пошукових систем. Також без них комп'ютер не вмітиме читати українською мовою показуватиме каралючки.*

<title>Тема 1**</title>** – *встановлює заголовок вікна вебсторінки.*

</head> – *обов'язково має бути закритий*

<body>

все, що хочемо бачити на сторінці пишемо тут

</body>

</html>- *цей закритий тег повинен завжди стояти в документі останнім.*

** ** тут текст ** ** – створює текстовий абзац. Тег завжди починається з нового рядка, абзаци тексту йдуть один за одним, розділяються між собою пустим рядком. Якщо закритого тега немає, вважається, що кінець абзацу збігається з початком наступного абзацу.

**
** – утворює новий рядок з того місця де він знаходиться.

<h1> … </h1> (та відповідно <h2>,<h3>,<h4>,<h5>,<h6>) – теги заголовків.

Тег <h1> представляє собою найбільш важливий заголовок першого рівня, а тег <h6> служить для позначення заголовка шостого рівня і є найменш значущим. За умовчанням, заголовок першого рівня відображається найбільшим шрифтом жирного накреслення, заголовки подальшого рівня за розміром менше.

Завдання 1.

1. У робочій області VSC в раніше створеній папці «html css» створіть папку t1z1 - для першого завдання.

2. У папці t1z1 створіть файл з назвою «index.html».

3. Включіть «Автозбереження файлів», для цього виберіть «Файл» - «Автосохранение».

4. Створіть вебсторінку за таким взірцем.

Всім привіт.

Ми з України!

Людмила МЕЛЕЩУК

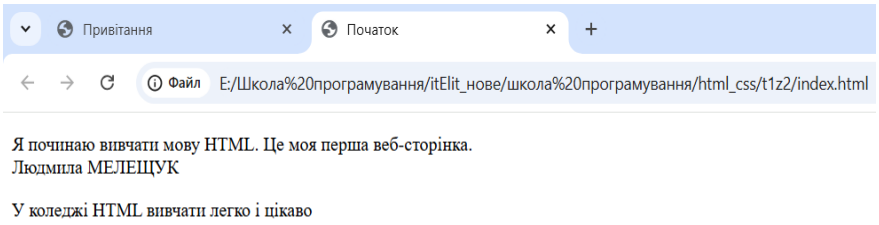
```
html_css > t1z1 > index.html > html
1   <!DOCTYPE html>
2   <html lang="uk">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Привітання</title>
7   </head>
8   <body>
9
10          <h2>Всім привіт.</h2>
11          <h1>Ми з України!</h1>
12          <h6>Людмила МЕЛЕЩУК</h6>
13
14      </div>
15  </body>
16  </html>
```

Завдання 2.

Для цього завдання у папці “html css” створіть нову папку з назвою “t1z2”. В подальшому для кожного завдання у папці “html css” створюйте нові папки у яких назви були б: t1z3, t2z1, t2z2, t2z3, t3z1 і так далі.

У новостворенній папці створіть початковий (основний чи запускний) файл з назвою “index.html”. В подальшому для кожного завдання створюйте основний файл index.html.

Використовуючи теги абзацу- ... і тег завершення рядка
 створіть вебсторінку за наведеним зразком.



```
html_css > t1z2 > index.html > ...
1   <!DOCTYPE html>
2   <html lang="uk">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Початок</title>
7   </head>
8   <body>
9       <p>Я починаю вивчати мову  HTML. Це моя перша веб-сторінка. <br>
10          Людмила МЕЛЕЩУК</p>
11      <p>У коледжі HTML вивчати легко і цікаво</p>
12  </body>
13  </html>
```

Завдання 3.

Зробіть додаткові зміни у завданні1, а саме:

Змініть фото на будь-яку картинку.

Ось приклад виконання цього завдання.


```
html_css > t1z1 > index.html > ...
1   <!DOCTYPE html>
2   <html lang="uk">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Привітання</title>
7   </head>
8   <body>
9       <div style="text-align: center;">
10          <img src="leopard.jpg" alt="фото леопарда" width="300">
11          <h2>Всім привіт.</h2>
12          <h1>Ми з України!</h1>
13          <h6>Людмила МЕЛЕЩУК</h6>
14
15      </div>
16  </body>
17  </html>
```

Завдання для самостійної роботи

1. Оформити у вигляді HTML-документа титульну сторінку практичного заняття №1 (На сторінку помістити логотип коледжу).

Звіт до практичного заняття №1

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Яким чином встановити VSC?
- Яким чином створюється і відображається вебсторінка?
- Що таке тег? Для чого теги використовуються?
- Яку структуру має HTML-документ?
- Як вирівняти текст по центру, по ширині, по лівому або правому краю?

Практичне заняття №2

Тема. Форматування тексту. Підключення зовнішнього CSS. Фон вебсторінки, фон елементів.

Мета: навчитися підключати зовнішні CSS, форматувати текст та змінювати фон вебсторінки і фон елементів з використанням мови опису HTML та CSS.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Теоретичні відомості

Тег **<link>** - прив'язує до html файлу каскадні таблиці стилів (CSS). Обов'язково записується всередині контейнера head

Синтаксис

***<link** rel="stylesheet" href="тут назва файлу.css">*

href="*адреса***"** (від англ. hypertext reference - гіпертекстове посилання) – задає адресу документа, на який слід перейти. В нашому випадку застосовуємо його у тезі link для прив'язки файлу css до вебсторінки.

class="…" - задає стильовий клас, який дозволяє пов'язати тег із стильовим оформленням (css). У значенні допускається вказувати відразу кілька класів, розділяючи їх між собою пропуском

Імена класів можуть містити в собі латинські літери (A-Z, a-z), цифри (0-9), символ дефісу (-) та підкреслення (_). *Використання українських літер заборонено!*

У CSS, щоб вибрати елемент з певним класом, напишіть символ крапки (.), а після назва класу. Наприклад для елемента <p **class="text1"**> Привіт світ! </p> звернення у стилях виглядатиме:

.text1 {
[параметри]
}

id="…" – вказує на унікальний ідентифікатор для елемента HTML. Значення ідентифікатора може використовуватися CSS для виконання певних завдань для унікального елемента із зазначеним значенням ідентифікатора.

У CSS, щоб вибрати елемент з певним ідентифікатором, напишіть символ хеша (#), а після назва ідентифікатора елемента.

Наприклад для елемента <p **id=“text2”**> Привіт світ! </p> звернення у стилях виглядатиме:

#text2 {
[параметри]
}

Нові властивості стилів:

CSS(абревіатура від Cascading Style Sheets, що в перекладі означає каскадні таблиці стилів) - це спеціальна мова (мова стилів), за допомогою якої описують вигляду документів (як і де відображати елементи вебсторінки).

Для того щоб задати для певного елемента (чи багатьох елементів) певні налаштування стилів, то потрібно звернутися до нього (або написати просто назву тегу(без дужок! або у випадку звернення до ідентифікатора чи класу слідкувати правилам їх запису)) → після поставити відкриту та закриту фігурні дужки **{ }**, та між ними вписувати параметри, виглядає це так:

назва параметру : значення ;

Зразок

body {
color: yellow ;
}

А тепер стосовно властивостей:

color - задає колір тексту елемента. Колір можна записувати як просто англійськими словами (black, green, blue, red ….) , так і іншими способами.

Синтаксис:

color: blue;

font-family – встановлює шрифт яким буде написаний текст.

Синтаксис:

font-family: ім'я шрифту;

font-size – вказує розмір шрифту елемента. Вказувати розмір можна різними способами, але ми будемо використовувати найбільш поширений це вказання конкретним значенням (може бути у пікселях (px) або піками (pt))

Синтаксис:

font-size: значеннярх(або pt) ;

font-weight – Встановлює значення (товщину) шрифту. Значення встановлюється від 100 до 900 з кроком 100. Надтонке накреслення, яке може відобразити браузер, має значення 100, а над товсте — 900. Нормальне

накреслення шрифту (яке встановлене за умовчанням) дорівнює 400, стандартний напівжирний текст — значення 700.

text-align – визначає горизонтальне вирівнювання тексту в межах елементу. Варто вирівнювати текст саме через цей стильовий параметр, а не атрибут у тезі, так код виглядає більш професійним!

Синтаксис:

text-align: center | justify | left | right ;

Універсальний атрибут **background** дозволяє встановити одночасно до п'яти характеристик фону. Значення можуть йти в будь-якому порядку, браузер сам визначить, яке з них відповідає потрібному властивості.

В CSS3 допустимо вказувати параметри відразу декількох фонів, перераховуючи їх через кому.

background-color – задає колір фону елемента.

Синтаксис:

background-color: <колір> ;

Градієнти - тип фону коли відбувається плавне перетікання від одного кольору до іншого. Для швидкого створення градієнтів використовуйте сервіс, а після просто скопіюйте готовий код та вставляйте у background.

Сайт сервісу: https://cssgradient.io/

Синтаксис:

background: linear-gradient(кут нахилу, колір 1, колір 2, колір N);

background-image – встановлює фонове зображення для елемента. Якщо одночасно для елемента задано колір фону, він буде показаний, поки фонова картинка не завантажується повністю. Те ж відбудеться, якщо зображення не доступні або їх показ в браузері відключений. У разі наявності в малюнку прозорих областей, через них буде проглядатися фоновий колір.

Синтаксис:

background-image: url(шлях до файлу) ;

background-position – задає початкове положення фонового зображення, встановленого за допомогою властивості **background-image**

Синтаксис: (цю лінію "|" читайте як слово АБО)

background-position: [left | center | right | <відсотки> | <значення>] || [Top | center | bottom | <відсотки> | <значення>]

background-attachment – встановлює, чи буде прокручуватись фонове зображення разом з вмістом елементу. Зображення може бути зафіксовано і залишатися нерухомим, або переміщатися разом з документом.

Синтаксис:

background-attachment: fixed | scroll | inherit ;

Пояснення значень:

fixed - робить фонове зображення елемента нерухомим.

scroll - Дозволяє переміщатися фону разом з вмістом.

inherit - Успадковує значення батька.

background-repeat – визначає, як буде повторюватися фонове зображення, встановлене за допомогою властивості background-image . Можна встановити повторення малюнка тільки по горизонталі, по вертикалі або в обидві сторони.

Синтаксис:

background-repeat: no-repeat | repeat | repeat-x | repeat-y | inherit ;

Пояснення значень:

no-repeat - Встановлює одне фонове зображення в елементі без його повторень, положення якого визначається властивістю background-position (за умовчанням в лівому верхньому куті). Аналогічно no-repeat no-repeat .

repeat - Фонове зображення повторюється по горизонталі і вертикалі. Аналогічно repeat repeat .

repeat-x - Шпалери повторюється тільки по горизонталі. Аналогічно repeat no-repeat .

repeat-y - Шпалери повторюється тільки по вертикалі. Аналогічно no-repeat repeat .

inherit - Успадковує значення батька.

background-size – масштабує фонове зображення відповідно до заданих розмірах.

Синтаксис:

background-size: [<значення> | <відсотки> | auto] {1,2} | cover | contain ;

Пояснення значень:

<значення> - Задає розмір в будь-яких доступних для CSS одиницях - пікселі (px), сантиметри (cm), em та ін.

<відсотки> - Задає розмір фонової картинки в процентах від ширини або висоти елемента.

auto - Якщо задано одночасно для ширини і висоти (auto auto), розміри фону залишаються вихідними; якщо тільки для одного боку картинки (100px auto), то розмір обчислюється автоматично виходячи з пропорцій картинки.

cover- Масштабує зображення зі збереженням пропорцій так, щоб його ширина або висота дорівнювала ширині або висоті блоку.

contain - Масштабує зображення зі збереженням пропорцій таким чином, щоб картинка цілком помістилася всередину блоку.

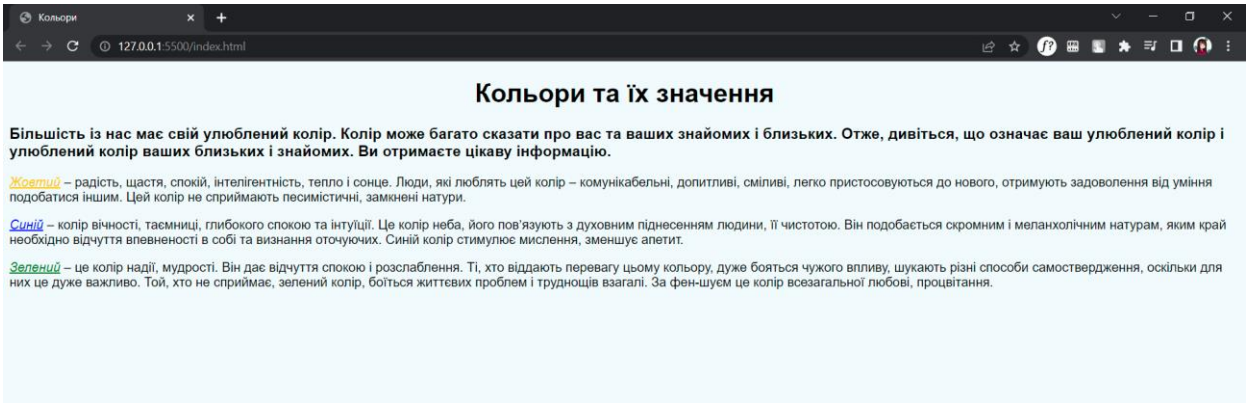
Якщо встановлено одне значення, воно задає ширину фону, друге значення приймається за auto. Пропорції картинки при цьому зберігаються. Використання двох значень через пробіл задає ширину і висоту фонової картинки.

Завдання для самостійної роботи

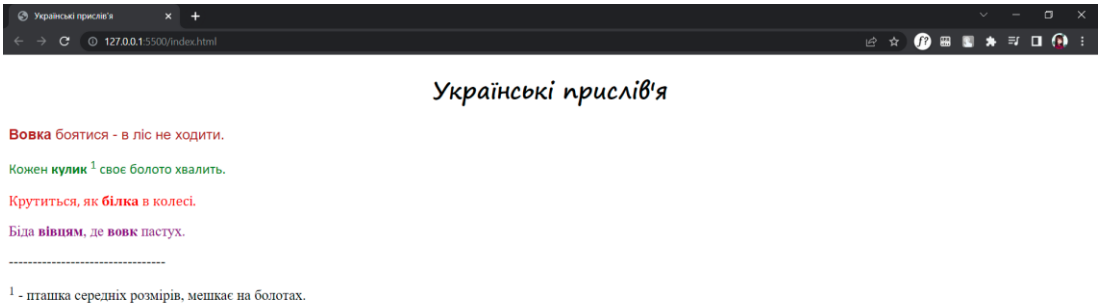
Завдання 1

Шрифт тіла документа - **sans-serif**

Колір тіла документа - **#f3fbfd**
Колір слова “жовтий” - **#ffca00**
Колір слова “синій”- **#0004fd**
Колір слова “зелений”- **#00821a**
Назви кольорів зробити курсивом та підкресленими
“Більшість із нас...” огорнути в заголовок третього рівня.



Завдання 2
Шрифт заголовку - **Segoe Print**
Шрифт і колір першого прислів'я - **Arial, brown**
Шрифт і колір другого прислів'я - **Calibri, green**
Шрифт і колір третього прислів'я - **Cambria, red**
Шрифт і колір четвертого прислів'я - **Consolas, purple**
Вага товстого шрифта - **600**
Усі назви звірів виділити жирним шрифтом;
Цифру “**1**” в примітці відформатувати як **верхній індекс.**

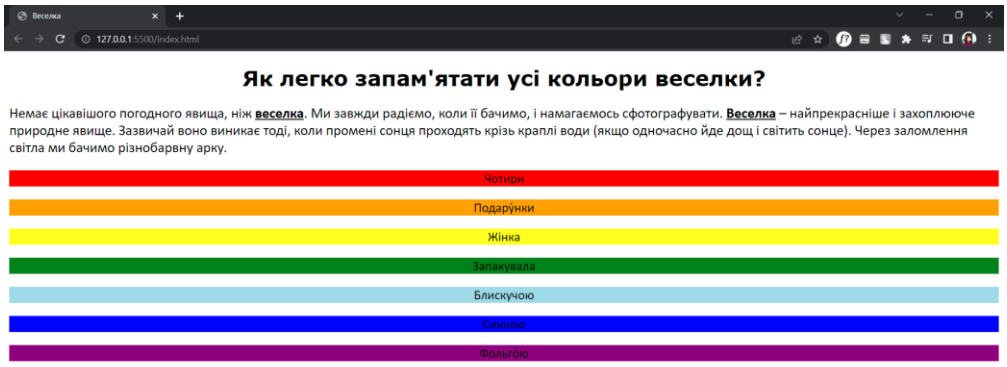


Завдання 3
Заголовок:
шрифт - **Verdana;**
для розміщення по центрі використати властивість **text-align: center;**
Усі абзаци:
шрифт - **Calibri**,
розмір шрифту - **20px**,
розміщення по центру за допомогою **text-align: center;**

Абзац про веселку:
розмір шрифту - **22px**,
розміщення по лівій стороні за допомогою **text-align**

Кожне слово **“веселка”** в тексті зробити:
жирним та підкресленим за допомогою тегів HTML

Для кожного з абзаців відповідно задати фон кольору,
який їм відповідає: Чотири(червоний), Подарунки(помаранчевий) і т.д.



Завдання 4
Для **тіла документу**:
фон-картинка - **задати картинку з тигром,**
заборонити повторюваність фону (background-repeat),
розмір фону - 100%,
колір шрифту - білий;

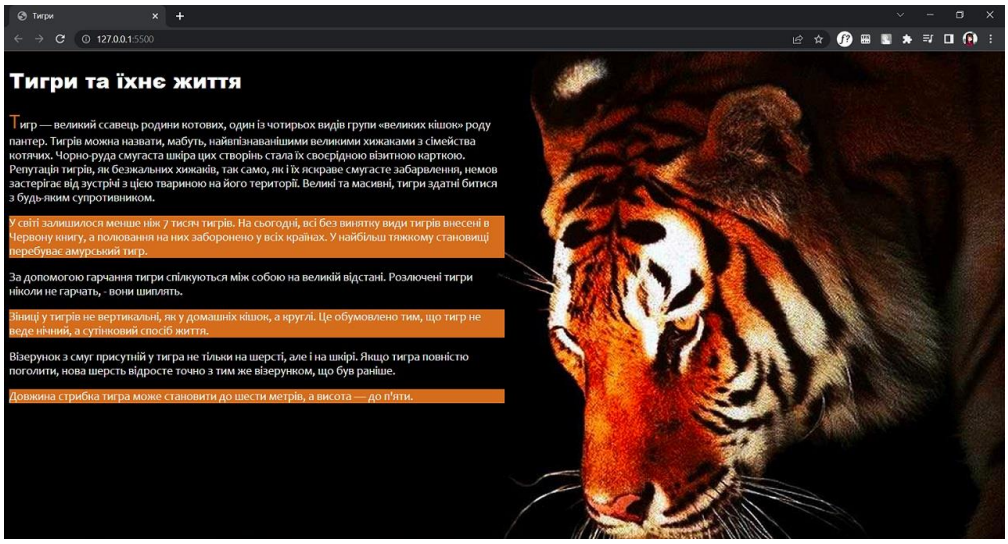
Заголовок:
Шрифт - **Arial**,
вага шрифту - **900**

Усім параграфи:
шрифт **Candara**,
розмір шрифту - **18px**,
ширина усіх абзаців(width) - **50%**

Першу букву тексту змінити за допомогою тегу **span** з атрибутом **class='letter':**
Колір - **#c6722a**,
розмір шрифту - **32px**

До параграфів які мають мати помаранчевий фон дописати атрибут **class='paragraph'**:

задати колір фону - **#c6722a**



Звіт до практичного заняття №2

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №3

Тема. Списки. Горизонтальні лінії.

Мета: навчитися створювати списки та горизонтальні лінії на web-сторінки з використанням мови опису HTML.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Теоретичні відомості

<ul> –встановлює маркований список. Кожен елемент списку повинен починатися з тега **<li>** .

Якщо до тегу <ul> застосовуються стилів CSS, то елементи <li> успадковують ці властивості.

Синтаксис:

 елемент маркованого списку

<ol> – встановлює нумерований список. Кожен елемент списку повинен починатися з тега **<li>** . Якщо до тегу <ol> застосовуються стилів CSS, то елементи <li> успадковують ці властивості.

Синтаксис:

 елемент нумерованого списку

 елемент нумерованого списку

<hr> – малює горизонтальну лінію, яка за своїм виглядом залежить від використовуваних параметрів, а також браузера. Тег <hr> відноситься до блокових елементів, лінія завжди починається з нового рядка, а після неї всі елементи відображаються на наступному рядку.

Атрибут **type** до тега ** –** встановлює вигляд маркера.

Синтаксис:

<ul **type=“**disc | circle | square**”**>

...

Для маркованого списку маркери можуть приймати один з трьох видів: замальоване коло (**disc**), коло (**circle**) і квадрат (**square**).

Атрибут **type** до тега ** –** встановлює вигляд маркера.

Синтаксис:

<ol type=“A | a | I | i | 1”>

...

Пояснення значень:

A - великі латинські літери;

a - рядкові латинські літери;

I - заголовні римські цифри;

i - рядкові римські цифри;

1 - арабські цифри (значення за замовчуванням)

Атрибут **reversed** до тега **<ol>** – змінює нумерацію в списку на зворотній порядок, замість 1,2,3 буде виводитися 3,2,1.

Синтаксис:

<ol reversed>

...

Атрибут **start** до тега **<ol>** – встановлює номер, з якого буде починатися перелік. При цьому не має значення, який тип списку встановлений за допомогою type, атрибут **start** однаково працює і з римськими і з арабськими числами.

Синтаксис:

<ol start=“число”>

 Приклад

Властивості стилів:

list-style-type*: none;* - прибирає будь яке маркування в усіх типах списків. Також замість none, можна писати назви типів маркованих списків (disc, circle, square) і задавати їх саме через css а не атрибут в тезі

Щоб налаштувати ширину та висоту лінії потрібно використати: **width –**встановлює ***ширину*** елементів

Height – встановлює ***висоту*** елементів

Синтаксис їх однаковий:

width/height: значення | відсотки | auto

Як значення приймаються будь-які одиниці довжини, прийняті в CSS — наприклад, пікселі (px), дюйми (in), пункти (pt) та ін. При використанні процентної запису ширина елемента обчислюється залежно від ширини батьківського елементу. Якщо батько явно не вказано, то в його якості виступає вікно браузера.

auto - встановлює ширину виходячи з типу і вмісту елемента.

text-transform – керує перетворенням тексту елемента у великі або прописні символи.

Синтаксис:

text-transform: capitalize | lowercase | uppercase | none | inherit

Пояснення значень:

capitalize – перший символ кожного слова в реченні буде заголовних. *Решта символів свій вигляд не змінюють.*

lowercase – все символи тексту стають малими (нижній регістр).

uppercase – все символи тексту стають прописними (верхній регістр).

none – не змінює регістр символів.

inherit – успадковує значення батька.

Псевдокласи – це ключові слова, які додаються до CSS селекторів і змінюють їх стан та положення внаслідок дій користувачів.

Псевдоклас**:before** – застосовується для додавання бажаного контенту **перед** вмістом елемента, до якого він додається. Працює спільно з властивістю **content**.

Синтаксис:

```
елемент:before{
        content:" текст/символ"
       }
```

Ідентично працює псевдоклас**: after** – додає бажаний контент **після** елементу.

Завдання для самостійної роботи

Завдання 1.

Тілу документа:

задати фон градієнтом **background: linear-gradient(133deg, rgba(251, 255, 69, 1) 45%, rgba(106, 149, 215, 1) 55%) ;**

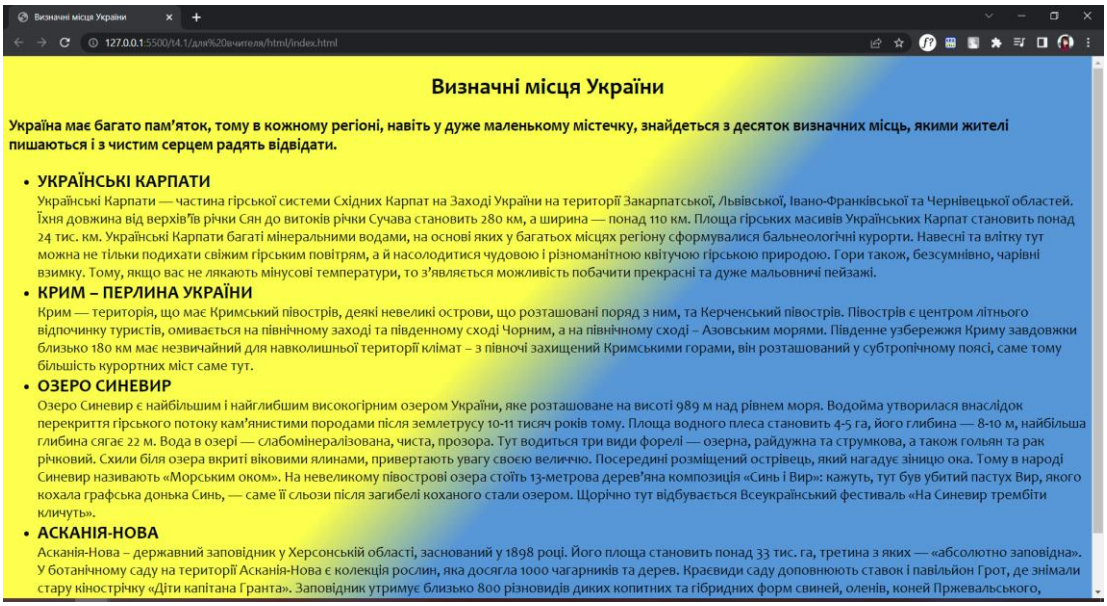
заборонити повторюваність заднього фону(**no-repeat**)

шрифт - **Candara**

Заголовок вирівняти по центру за допомогою **text-align: center**
Підзаголовок виділити жирним шрифтом будь-яким зручним способом (тег b, strong або за допомогою span та CSS)

Розмір шрифт **підзаголовку - 22px**
Розмір шрифта пунктів списку (**li**) - **20px**

Назви пунктів списку виділити за допомогою тегу span та задати такі параметри:
Зробити великими буквами,
розмір шрифта - 25px,
вага шрифта - 600



Завдання 2.
Тіло сайту:

колір - #D7BDE2

Шрифт - Verdana
Розмір шрифту - 22px

Заголовок:

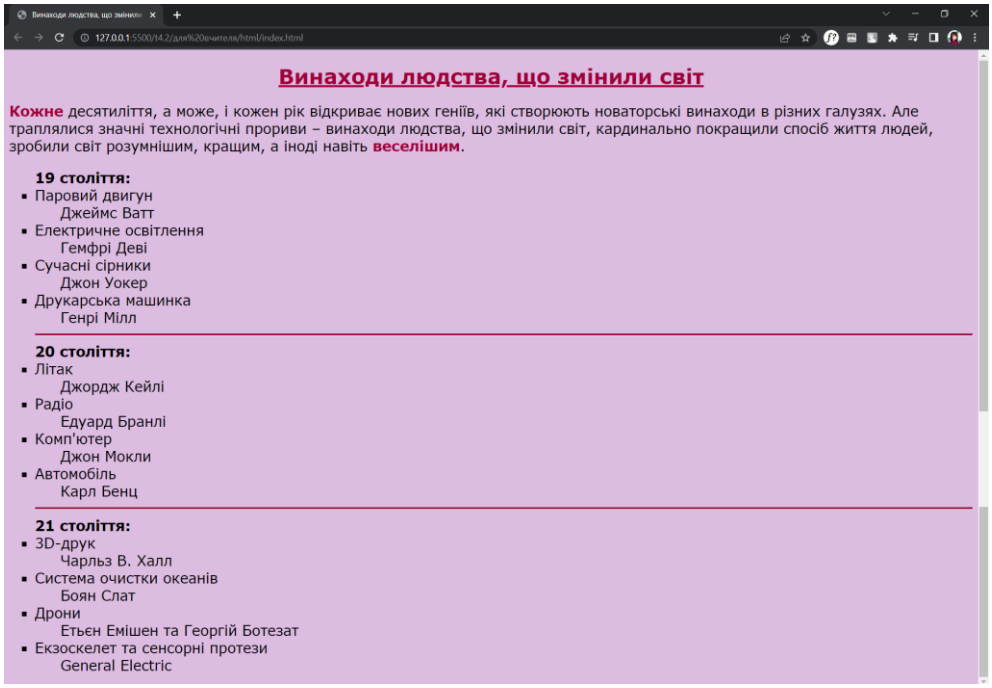
Колір - #900C3F,

розмір шрифту - 32px,

розміщений по центру.

Першому та останньому слову задати такі стилі *(виділити їх за допомогою тегу span)*:
колір - **#900C3F**
вага шрифту - **900**

Горизонтальна лінія задати такі стилі:
висота - **2px**
колір фону - **#900C3F**



Завдання 3.

Тіло сайту:

фон-зображення - прапор Японії (japang.png).
позиціонування фонового зображення - по верху *(background-position: top;)*
шрифт - 'Sitka Text'.

Заголовок:

розмістити посередині,
виділити великими буквами
колір - **#bc002d**.

Елементам лінії задати такі параметри:
ширина - **400px**,
висота - **5px,**
колір фону - **#bc002c18**.

Елементам **параграф** та **ненумерований список** задати розмір шрифту **22px**.
В елементів списку(**li**) забрати маркування за допомогою властивості
list-style-type: none
Після цього, задати елементам списку нове маркування наступним способом з використанням псевдоелементу **before:**

li::before {
content: '(ᵔᵕᵔ)';
}

Текст з поясненням японських назв виділити жирним шрифтом будь-яким зручним способом та зробити курсивом.
Кожній першій букві фактів задати такі стилі:
розмір шрифта - **32px**
колір - **#bc002d**

Звіт до практичного заняття №3

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Якими тегами створюються маркеровані списки в HTML?
- Якими тегами створюються нумеровані списки в HTML;
- Як змінити тип маркеру в маркірованому списку?
- Як визначаються вкладені списки?
- Для чого використовується графіка в HTML?
- Опишіть псевдокласи?

Практичне заняття №4

Тема. Підсумкова робота 1.

Мета: узагальнити навички форматування елементів на web-сторінки з використанням мови опису HTML та CSS.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

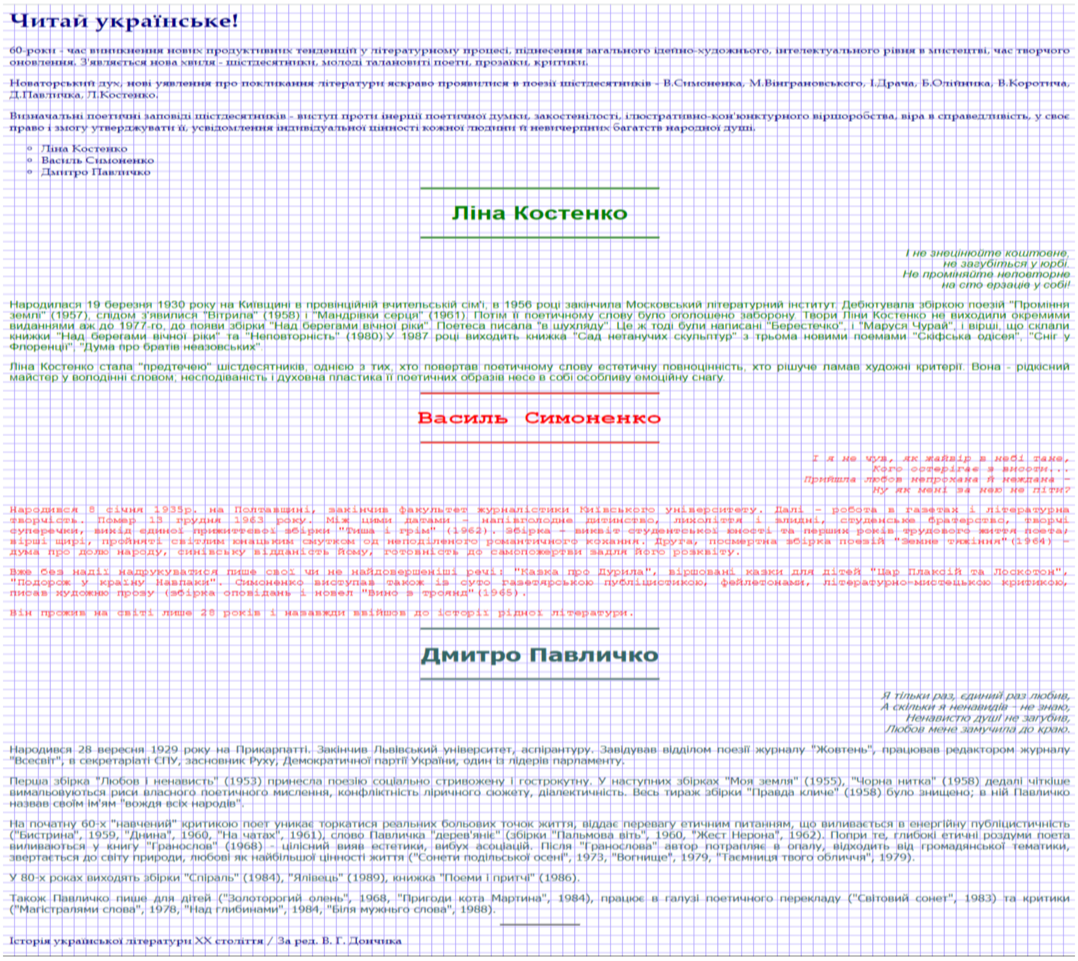
Завдання для самостійної роботи

Завдання 1

Створити сторінку *ukr.html* (див. мал.1), всі матеріали шукайте в інтернеті, виконати наступні завдання:

Вимоги до Web-сторінки:

- основний шрифт: гарнітура - Book Antiqua; колір - темно-синій (navy);
- стаття про Ліну Костенко: гарнітура - Arial; колір - зелений (green);
- стаття про Василя Симоненка: гарнітура - Courier New; колір - червоний (red);
- стаття про Дмитра Павличка: гарнітура - Tahoma; колір - #336666;
- всі уривки віршів потрібно виділити курсивом.
- Фонове зображення для web-сторінки створити самостійно (розмір клітинки 15x15 пікселів).



Мал.1

Завдання 2

Створити сторінку *tesla.html* (див. мал.2).

Вимоги до Web-сторінки:

HTML :

Короткий опис структури сайту *(те що у body)***:**

<**h1** class=“title”> *... ...тут заголовок. . .*</**h1**>
<**p**>
<**span** class=“first__letter”>*тут перша буква*</**span**> *. .. . . .весь інший текст. . . .. ..*
</**p**>
<**h3** class=“subtitle”> *тут назва розділу (наприклад Tesla Roadster)*</**h3**>
<**p**>
<**span** class=“first__letter”>*тут перша буква*</**span**> *. .. . . .весь інший текст. . ..*
</**p**>
<**ol**... *..тут потрібно дописати певний атрибут (дальше у поясненні знайдете).... . ..>*
тут пункти списку
</**ol**>
і так дальше наступні частини сайту продовжуються із *заголовків 3-рівня, абзаців із виділиними першими літерами* та інших елементів.

CSS :
Тіло сайту:
Фоном-зображення - **tesla.jpg**,
повторюваність по осі Y *(background-repeat: repeat-y;),*
позиція фону - права сторона *(background-position: right;),*
розмір фону - **20%**.
шрифт - Sylfaen,
розмір шрифта - 25px.

Заголовок *(має бути першого рівня):*
колір тексту - **#b04545**,
розмір шрифту - **32px**;

Підзаголовок *(має бути третього рівня):*
колір фону - **#b04545**,
колір тексту - **білий**;

Заголовку та **підзаголовку:**
шрифт - **Corbel**,
позиціонування тексту по центру.

Перші букви абзаців *(мають бути у тезі span)*:
Колір - **#b04545**
Розмір шрифта - **32px**

Вага шрифта - **600**

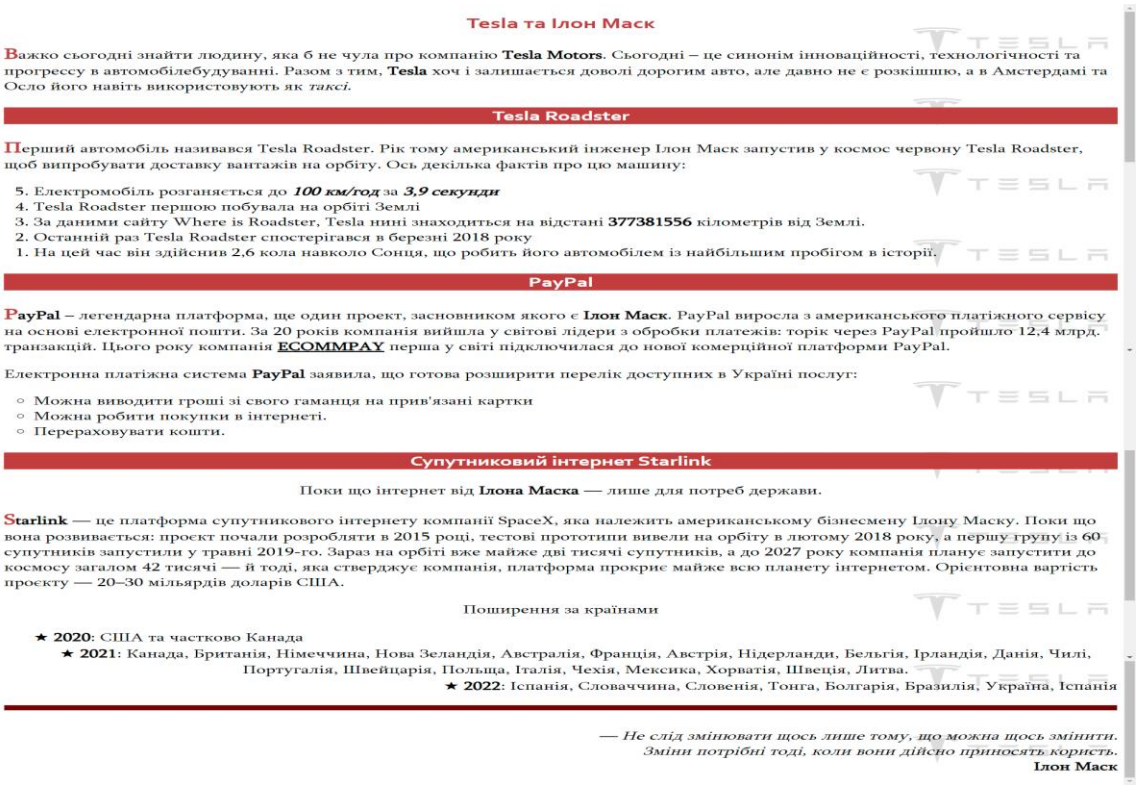
У номерованому списку зробити зворотню нумерацію додавши атрибут **reversed** до тегу списку.

Горизонтальна лінія:
висота - **35px**,
колір фону - **#b04545**

Останній список із ★:
кожному пункту списку додати атрибут **id=' ' із різними назвами**, та **задати розміщення**:
для першого - зліва
для другого - по центрі
для третього - справа

Цитата:
Написати **курсивом**,
позиціонування тексту п**о правій стороні**,
ширина - 50%.
розміщення - по правій стороні (**float: right**)

Усі елементи в тексті виділити необхідним типом шрифта (курсив, жирний) **відповідно до скріншноту.**



Tesla та Ілон Маск

Важко сьогодні знайти людину, яка б не чула про компанію **Tesla Motors**. Сьогодні – це синонім інноваційності, технологічності та прогресу в автомобілебудуванні. Разом з тим, **Tesla** хоч і залишається доволі дорогим авто, але давно не є розкішшю, а в Амстердамі та Осло його навіть використовують як *таксі*.

Tesla Roadster

Перший автомобіль називався Tesla Roadster. Рік тому американський інженер Ілон Маск запустив у космос червону Tesla Roadster, щоб випробувати доставку вантажів на орбіту. Ось декілька фактів про цю машину:

5. Електромобіль розганяється до ***100 км/год*** за ***3,9 секунди***
4. Tesla Roadster першою побувала на орбіті Землі
3. За даними сайту Where is Roadster, Tesla нині знаходиться на відстані **377381556** кілометрів від Землі.
2. Останній раз Tesla Roadster спостерігався в березні 2018 року
1. На цей час він здійснив 2,6 кола навколо Сонця, що робить його автомобілем із найбільшим пробігом в історії.

PayPal

PayPal – легендарна платформа, ще один проект, засновником якого є **Ілон Маск**. PayPal виросла з американського платіжного сервісу на основі електронної пошти. За 20 років компанія вийшла у світові лідери з обробки платежів: торік через PayPal пройшло 12,4 млрд. транзакцій. Цього року компанія **ECOMMPAY** перша у світі підключилася до нової комерційної платформи PayPal.

Електронна платіжна система **PayPal** заявила, що готова розширити перелік доступних в Україні послуг:

- Можна виводити гроші зі свого гаманця на прив'язані картки
- Можна робити покупки в інтернеті.
- Перераховувати кошти.

Супутниковий інтернет Starlink

Поки що інтернет від **Ілона Маска** — лише для потреб держави.

Starlink — це платформа супутникового інтернету компанії SpaceX, яка належить американському бізнесмену Ілону Маску. Поки що вона розвивається: проєкт почали розробляти в 2015 році, тестові прототипи вивели на орбіту в лютому 2018 року, а першу групу із 60 супутників запустили у травні 2019-го. Зараз на орбіті вже майже дві тисячі супутників, а до 2027 року компанія планує запустити до космосу загалом 42 тисячі — й тоді, яка стверджує компанія, платформа прокриє майже всю планету інтернетом. Орієнтовна вартість проекту — 20–30 мільярдів доларів США.

Поширення за країнами

★ **2020**: США та частково Канада
★ **2021**: Канада, Британія, Німеччина, Нова Зеландія, Австралія, Франція, Австрія, Нідерланди, Бельгія, Ірландія, Данія, Чилі, Португалія, Швейцарія, Польща, Італія, Чехія, Мексика, Хорватія, Швеція, Литва.
★ **2022**: Іспанія, Словаччина, Словенія, Тонга, Болгарія, Бразилія, Україна, Іспанія

— Не слід змінювати щось лише тому, що можна щось змінити. Зміни потрібні тоді, коли вони дійсно приносять користь.
Ілон Маск

Мал. 2

Звіт до практичного заняття №4

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №5

Тема. Розміщення зображень на Web-сторінці. Межі елементів. Зовнішні відступи. Різновиди вставка відео та аудіо на сайті.

Мета: навчитися вставляти зображення, відео та аудіо на web-сторінки з використанням мови опису HTML та CSS.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Теоретичні відомості

<img> - призначений для відображення на вебсторінці зображень в графічному форматі GIF, JPEG або PNG. Адреса файлу з картинкою задається через **атрибут src** . Якщо необхідно, то малюнок можна зробити посиланням на інший файл, помістивши тег <img> в контейнер <a>, але це будемо робити уже на наступних заняттях.

При цьому навколо зображення відображається рамка, яку можна налаштувати з допомгою стилів.

Синтаксис:

<img src="URL" alt="альтернативний текст">

<section> - Задає розділ документа, може застосовуватися для блоку новин, контактної інформації, розділів тексту, вкладок в діалоговому вікні та ін. Розділ зазвичай містить заголовок. Допускається вкладати один тег <section> всередину іншого.

Синтаксис:

<section> ТЕКСТ </section>

** ** - цей елемент є блоковим елементом і призначений для виділення фрагмента документа з метою зміни виду вмісту. Як правило, вигляд блоку управляється за допомогою стилів додаванням до цього тега атрибут class або id з ім'ям селектора.

Як і при використанні інших блочних елементів, вміст тега завжди починається з нового рядка. Після нього також додається перенесення рядка.

Синтаксис:

...

Атрибут **src** до тегу **<img>** - адреса до фото, яке буде відображатися на веб-сторінці.

Синтаксис:

<img src=“URL”> якщо фото знаходиться поруч файлі index, то потрібно записувати: <img src=“**./**”> і обирати картинку із меню програми, що з'явилося.

Атрибут **alt** до тегу **<img>** - атрибут alt встановлює альтернативний текст для зображень. Такий текст з’явиться на місці зображення, якщо через якісь причини браузер його не завантажив.

Синтаксис:

<img alt=“текст про зображення”>

Width - встановлює ширину блокових та інших елементів (до них, наприклад, відноситься тег <img>)

Синтаксис:

width: значення | відсотки | auto | inherit

height - встановлює ширину блокових та інших елементів (до них, наприклад, відноситься тег <img>)

Синтаксис:

height: значення | відсотки | auto | inherit

border - універсальний атрибут дозволяє одночасно встановити товщину, стиль і колір межі навколо елемента. Значення можуть йти в будь-якому порядку, розділяючись пробілом, браузер сам визначить, яке з них відповідає потрібному властивості.

Для установки межі тільки на певних сторонах елементу, скористайтеся властивостями border-top, border-bottom, border-left, border-right.

СТИЛІ РАМОК:

Синтаксис:

border: товщина стиль колір

Наприклад: border: 3px solid black

border-radius - встановлює радіус скруглення куточків рамки. Якщо рамка не задана, то округлення також відбувається і з фоном.

Синтаксис:

border-radius: <радіус>

audio - додає, відтворює і керує налаштуваннями аудіозаписів на веб-сторінці.

Шлях до файла задається через атрибут src або вкладений тег <source>. Всередині контейнера <audio> можна написати текст, який буде виводитися в браузерах, які не працюють з цим тегом.

Синтаксис:

<audio src="URL">

</audio>

або

<audio>

<source src="URL">

</audio>

Video - додає , відтворює і управляє настройками відеоролика на веб-сторінці. Шлях до файлу задається через атрибут **src** або вкладений **тег <source>**.

Синтаксис:

<video>

<source src="URL">

</video>

Атрибути до тегу audio:

autoplay - звук починає грати відразу після завантаження сторінки.

controls - додає панель керування до аудіофайлу.

loop - повторює відтворення звуку з початку після його завершення.

preload - використовується для завантаження файлу разом із завантаженням веб-сторінки.

src - вказує шлях до відтворюваного файлу.

Атрибути до тегу video:

autoplay - Відео починає відтворюватися автоматично після завантаження сторінки.

controls - Додає панель управління до відеоролика.

height - Задає висоту області для відтворення відеоролика.

loop - Повторює відтворення відео від початку до його завершення.

poster - Вказує адресу картинки, яка буде від зображують, поки відео недоступне або не чути.

preload - Використовується для завантаження відео разом із завантаженням веб-сторінки.

src - Вказує шлях до відтворюваного відеоролику.

width- Задає ширину області для відтворення відеоролика.

display - багатоцільова властивість, яка визначає, як елемент повинен бути показаний в документі.

Можливі значення:

display: block | inline | inline-block | inline-table | list-item | none | run-in | table | table-caption | table-cell | table-column-group | table-column | table-footer-group | table-header-group | table-row | table-row-group

Завдання для самостійної роботи

Завдання 1.

Забрати усі стандартні зовнішні та внутрішні відступи за допомогою коду:

```
* {
   margin: 0;
   padding: 0;
}
```

Тіло документа:

шрифт - Calibri

Блок із фоном квітів зверху:

фонове зображення - flower.jpg

висота - 400px

заборонити повторюваність фону

розмір фону - покриття (background-size: cover)

позиціонування фону - центер

Заголовок:

вирівнювання тексту по центру

зовнішній відступ зверху - 20px

Для того щоб обмежити ширину усіх блоків одразу, необхідно створити зовнішній контейнер, у який ми будемо поміщати усі блоки.

Контейнер:

максимальна ширина - 1200px (max-width: 1200px)

ширина - 100%

вирівнювання усього контенту по центру за допомогою margin: 0 auto

Усім параграфам сторінки задати розмір шрифта 24px

Вступний параграф під заголовком:

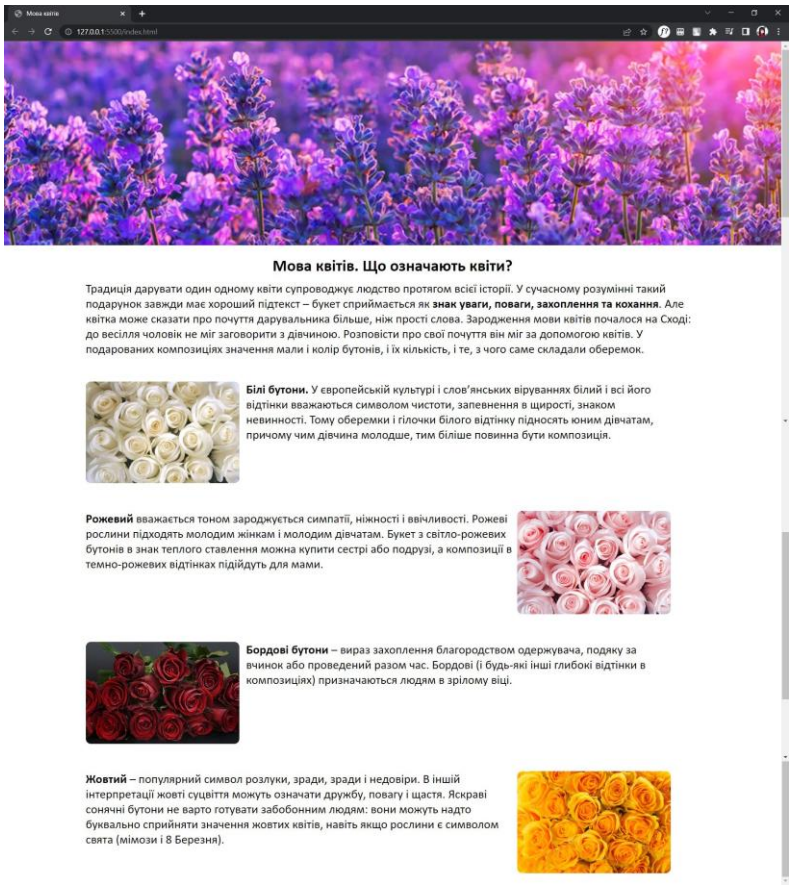
зовнішній відступ зверху та знизу - 10px

Блок(одне зображення і один текст в одному блоці):

зовнішній відступ зверху - 50px

зовнішній відступ знизу - 20px

Зображення всередині блоку:
ширина - 25%
висота - авто
дисплей - лінійний блок
зовнішній відступ справа - 10px
радіус кордонів - 10px
Параграфи всередині блоку:
ширина - 70%
дисплей - лінійний блок
вертикальне вирівнювання - верх (vertical-align: top)



Завдання 2.
Тілу сайта задати такі параметри:
колір фону - #FFD857
шрифт - 'Georgia', розмір - 22px

Заголовок:
розмір шрифту - 32px
розміщення по центру
колір - #d60303

Для елементу ul обнулити зовнішні відступи (margin: 0)

Створити два блоки (div). В одному буде розміщено текст, в іншому - картинку.
Лівому блоку задати такі параметри:
ширина - 70%
притиснути елемент до лівого краю за допомогою float: left

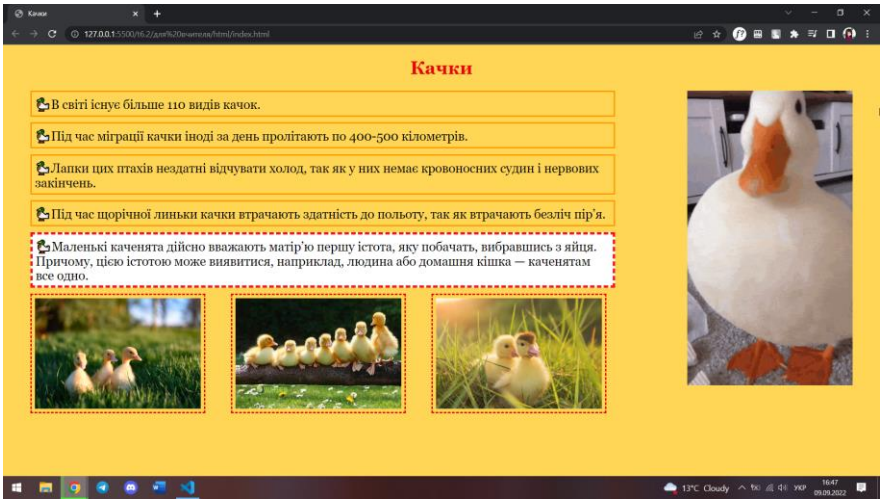
Правому блоку задати такі параметри:
притиснути елемент до правого краю
зовнішній відступ справа - 40px

Елементам списку задати такі параметри:
кордони - 3px, суцільні, колір - #ff9e02
зовнішній відступ знизу- 10px
внутрішні відступи- 5px
типи списку- none

задати нове маркування - **i**

Розмір усіх зображенн на сторінці - 300px
Зображенням під списком задати такі налаштування:
кордони: 3px штрихові колір- #d60303
внутрішні відступи- 5px
зовнішній відступ зліва - 40px

Найцікавіший, на Вашу думку, факт виділити такими параметрами за допомогою id:
колір фону - білий
кордони - 4px, штрихові, колір - #dd3232



Завдання 3.
Для відео та фото створити папку “**assets**”, у ній ще три папки - “**img**”, “**video**” та “**audio**” та помістити в них файли.

Тіло документа:задати зображення background.png, заборонити повторюваність, позиція фону - право, низ (right, bottom). Висота тіла 91vh
HTML задати фоновий колір aqua.
Заголовок:
колір - blue,
шрифт - Verdana
розміщення по центру

Відео вставити з такими параметрами:

```
<video class=“video” controls=“controls” autoplay muted loop>
    <source src=“./assets/video/dolphin1.mp4” type=“video/mp4”>
</video>
```

Це надасть нашим відео такі можливості:
клас - для подальшого редагування зовнішнього вигляду відео (*class=“video”*)
відображення із можливістю контролювати відтворення - звук, пауза, швидкість відтворення (*controls=“controls”*)
автовідтворення разом із завантаженням сторінки (*autoplay*)
автовідтворення в більшості браузерів неможливе без атрибуту “*muted*”
зациклене відео (*loop*)

Вставити аудіо bgsound.mp3 з параметрами autoplay та loop

Відео помістити в блок та задати йому такі параментри:
зовнішній відступ зверху - 50px
розміщення по центру за допомогою text align

Відео:
дисплей - inline-block
ширина - 400px
висота - авто
зовнішній відступ зліва - 30px

Список:
зовнішній відступ - 0
внутрішній відступ - 0

Елемент списку:
забрати стиль нумерування списку
шрифт - 'Constantia'
розмір шрифту - 22px

Псевдоелемент before для елементів списку:
контент -

Вибрати найцікавійший, на Вашу думку, факт та задати йому такі параметри:
зовнішній відступ зверху та знизу - 10px
фоновий колір - aliceblue
кордони зверху та знизу - 4px, подвійні, колір - forestgreen

Звіт до практичного заняття №5

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №6

Тема. Внутрішні та зовнішні гіперпосилання. Псевдоклас :hover. Кнопки із гіперпосилань. Використання зображень в якості гіперпосилань. Використання іконок Font Awesome. Основи Flex.

Мета: освоїти основні методи використання внутрішніх та зовнійшніх гіперпосилань, іконок Font Awesome. Навчитися вказувати властивості для елементів гіпертекстової розмітки HTML-документа. Ознайомитися з основами Flex.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Теоретичні відомості

Тег <a> - є одним з важливих елементів HTML і призначений для створення посилань. Атрибут href тег <a> встановлює посилання або якір.

Синтаксис:

...

...

Тег <nav> - задає навігацію по сайту. Якщо на сторінці кілька блоків посилань, то в <nav> зазвичай поміщають пріоритетні посилання. Також допустимо використовувати кілька тегів <nav> в документі.

Атрибути до тегу <a>:

href - задає адресу документа, на який варто перейти.

target - вказує, де відкрити документ при переході за посиланням. За замовчуванням, при переході за посиланням документ відкривається в поточній вкладці браузера. Значення _blank - відкриває пов'язаний документ в новому вікні або вкладці.

Псевдоклас :hover- визначає стиль елементу при наведенні на нього курсора миші, але при цьому елемент ще не активований, іншими словами кнопка миші не було натиснуто.

Синтаксис: елемент :hover {...}

transition- встановлює ефект переходу між двома станами елемента, вони можуть бути визначені за допомогою псевдоелемента :hover.

Синтаксис:

transition: час переходу (у s-секунди чи ms-мілісекунди (1 секунда = 1000 мілісекунд))

text-shadow- додає тінь до тексту, а також встановлює її параметри: колір тіні, зміщення щодо напису і радіус розмиття.

Синтаксис:

text-shadow: none | тінь [, тінь] * де тінь: <зрушення по x> <зрушення по y> <радіус розмиття> <колір>

Значення

None - Скасовує додавання тіні.

Колір - Колір тіні у будь-якому доступному CSS форматі. За умовчанням колір тіні збігається з кольором тексту. Необов'язковий параметр.

зрушення по x - зсув тіні по горизонталі щодо тексту. Позитивне значення цього параметра задає зсув тіні вправо, негативне - вліво. Обов'язковий параметр.

зрушення по y - зсув тіні по вертикалі щодо тексту. Також допустимо використовувати від'ємне значення, яке піднімає тінь вище тексту. Обов'язковий параметр.

Радіус - Задає радіус розмиття тіні. Чим більше це значення, тим сильніше тінь згладжується, стає ширше і світліше. Якщо цей параметр не заданий, за умовчанням встановлюється рівним 0.

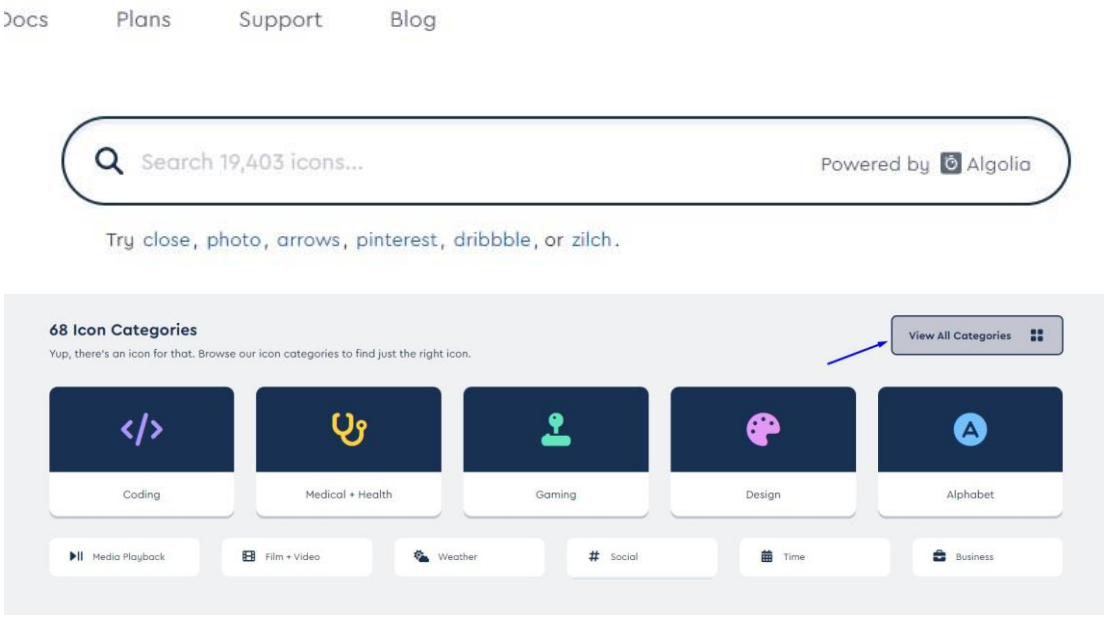
Підключення Font Awesome :

1. Скопіюйте даний рядок коду та вставте в head *(але перед підключенням основних стилів style.css)*

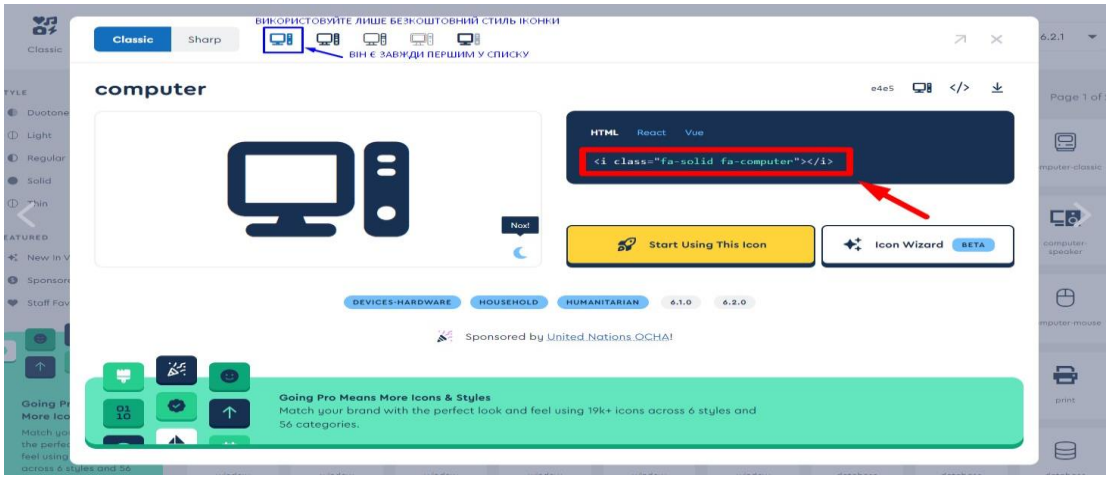
<script src="https://kit.fontawesome.com/af737f80ce.js" crossorigin="anonymous"></script>

2. Відкрийте сайт ***https://fontawesome.com/icons***

3. Знайдіть необхідну вам іконку скориставшись рядком пошуку, або вибравши один із запропонованих розділів.



4. Скопіюйте код іконки і вставте у необхідне місце в body



5. Додайте для іконки потрібні вам стилі.

Transform -трансформує елемент, зокрема, дозволяє його масштабувати, обертати, зрушувати, нахиляти, а також комбінувати види трансформацій.

Синтаксис:

transform: <функція>

Функції трансформації основні:

Rotate - поворот елемента на заданий кут відносно точки трансформації, що задається властивістю transform-origin .

Зразок: transform: rotate (< кут >)

Scale - масштаб елемента по горизонталі і вертикалі.

Зразок: transform: scale (sx [, sy]);- значення більше 1 збільшує масштаб елемента, менше 1 - зменшує масштаб.

scroll-behavior - визначає поведінку прокрутки для будь-якого елемента на сторінці

Значення:

Auto - стандартна поведінка прокрутки без ефекту плавності.

Smooth - елемент прокручується плавно;

Синтаксис:

scroll-behavior: auto або *smooth*

Основна (найпростіша) структура використання Flex-ів :

У файлі .html

```
<body>
    <section class="container">
        <div class="inner">
            <!-- 1 елемент, що буде флексом -->
            <!-- 2 елемент, що буде флексом -->
            <!-- n елемент, що буде флексом -->
        </div>
    </section>
</body>
!УВАГА! Флексами можуть бути такі елементи,
як просто блоки (теги div), покликання (теги a),
зображення (теги img) та інші
```

У файлі .css

```
/* Базові налаштування секції: */
.container {
    max-width: 1200px; /* налаштування максмальної ширини (якщо потрібно) */
    width: 100%; /* розтягування вмісту на всю ширину */
    margin: 0 auto; /* вирівнювання по центрі (якщо потрібно) */
}
/* Головні налаштування, щоб запрацювали флекси */
.inner {
    display: flex; /* перетворення елементів на флекси */
    flex-direction: row; /* щоб флекси стояли у рядок,
    якщо потрібно у стовпчик то значення буде: column */
    align-items: center; /*  вирівнювання по висоті */
    justify-content: space-between; /* вирівнювання по ширині  */
}
.елемент {
    /* стилізація елементів */
}
```

Властивості стилів:

flex-direction - напрямок розташування флексів.

Можливі значення:

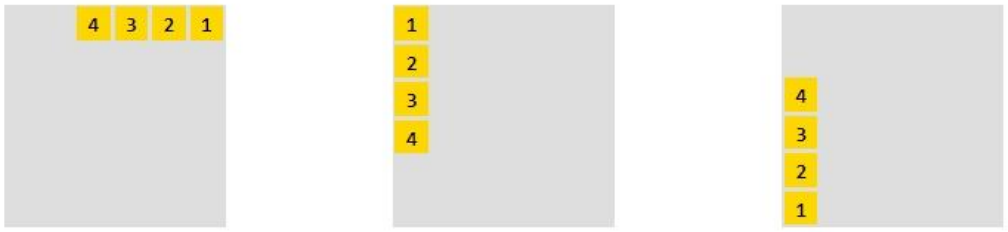
row - розташування у рядок (зліва на право)

row-reverse - розташування у рядок дзеркально (справа на ліво)

column - розташування у стовпчик (вертикально) (зверху вниз)

column-reverse - розташування у стовпчик дзеркально (вертикально) (знизу вверх)

Ви бачите три ідентичних блоки, але для них задані різні напрямки осі: *row-reverse*, *column* й *column-reverse* відповідно.



align-items - вирівнювання флексів по вертикалі

Можливі значення:

stretch - елементи розтягуються та займають весь доступний простір контейнера.

center - елементи розташовані по центру контейнера.

flex-start - елементи розташовані на початку контейнера.

flex-end - елементи розташовані в кінці контейнера.

baseline - елементи розташовані на базовій лінії контейнера.

Ви бачите два ідентичні контейнери, але для них задані різні значення для вирівнювання: *flex-start*, *center* й *flex-end* відповідно.

justify-content - вирівнювання флексів по горизонталі

Можливі значення:

flex-start - елементи розташовані на початку контейнера.

flex-end - елементи розташовані в кінці контейнера.

center - елементи розташовані в центрі контейнера.

space-between - флекси рівномірно розподіляються по всьому рядку. Перший і останній флекс притискаються до відповідних країв контейнера.

space-around - флекси рівномірно розподіляються по всьому рядку. Порожній простір перед першим і після останнього елементів дорівнює половині простору між двома сусідніми елементами.

Ви бачите три ідентичних блоки, але для них задані різні значення для justify-content: *flex-end*, *center* й *space-around* відповідно.

Завдання для самостійної роботи

Завдання 1.

Створити папку “**assets**” та ще одну в ній - “**img**”, і помістити туди усі зображення,завантажити їх можна у classroom.

Забрати усі стандартні відступи.

Тілу документа задати фонове зображення back.jpg

Усім посиланням задати такі параметри:

забрати текстову декорацію

колір - yellow

вага шрифта - 600

При наведенні на посилання:

колір - #ffa600

Заголовок:

вирівнювання тексту по центру

шрифт - Verdana

колір - білий

зовнішні відступи зверху та знизу - 20px

Створити три блоки (div class=“box”), в які буде поміщатись уся інформація.

box:

колір фону - #4d4d8e

внутрішні відступи - 25px

зовнішні відступи - 20px

радіус корднонів - 25px

В кожному блокові повинні бути: назва планети, фото та опис. Елемент фото та опису змінювати місцями відповідно до їхнього положення на скріні - якщо спочатку йде картинка, то її необхідно вставляти першою, якщо текст - навпаки.

Назва планети:

дисплей - блоки

вирівнювання тексту по центру

ширина - 100%

колір - білий

шрифт - Cambria

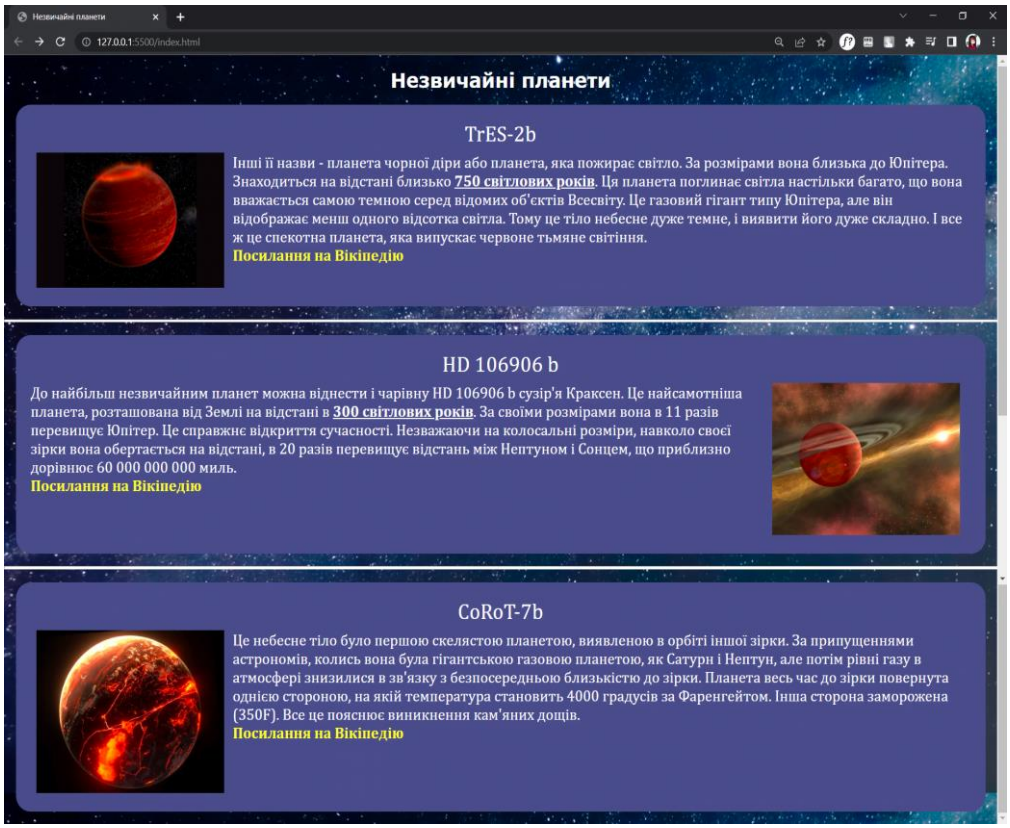
розмір шрифта - 35px

зовнішні відступи знизу - 10px

Зображення всередині box:
ширина - 20%
дисплей - лінійний блок (inline-block)
зовнішні відступи справа та зліва - 10px

Опис планети:
ширина - 78%
вертикальне вирівнювання - верх
дисплей - лінійний блок
колір - білий
шрифт - Cambria
розмір шрифта - 25px

Лінії, що розділяють блоки:
висота - 3px
колір фону - білий
Текст для вебсторінки можна завантажити у classroom.



Завдання 2.

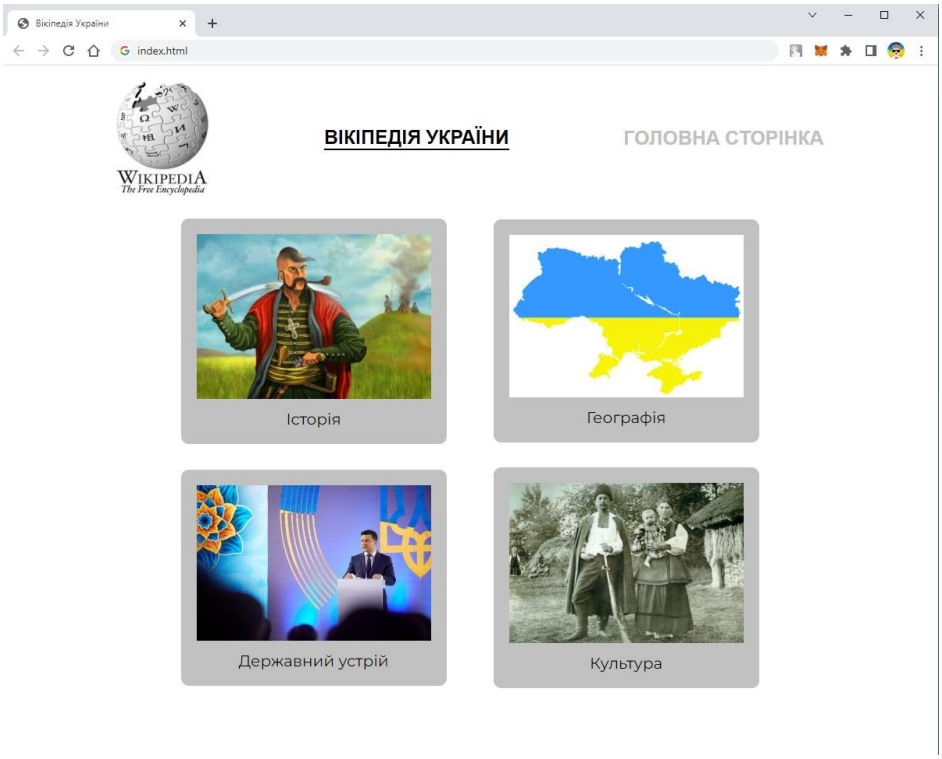
Розмістити посилання в потрібних місцях та зробити так, щоб кожне посилання відкривалось в новому вікні.

Головна сторінка - https://cutt.ly/FXuH1rh

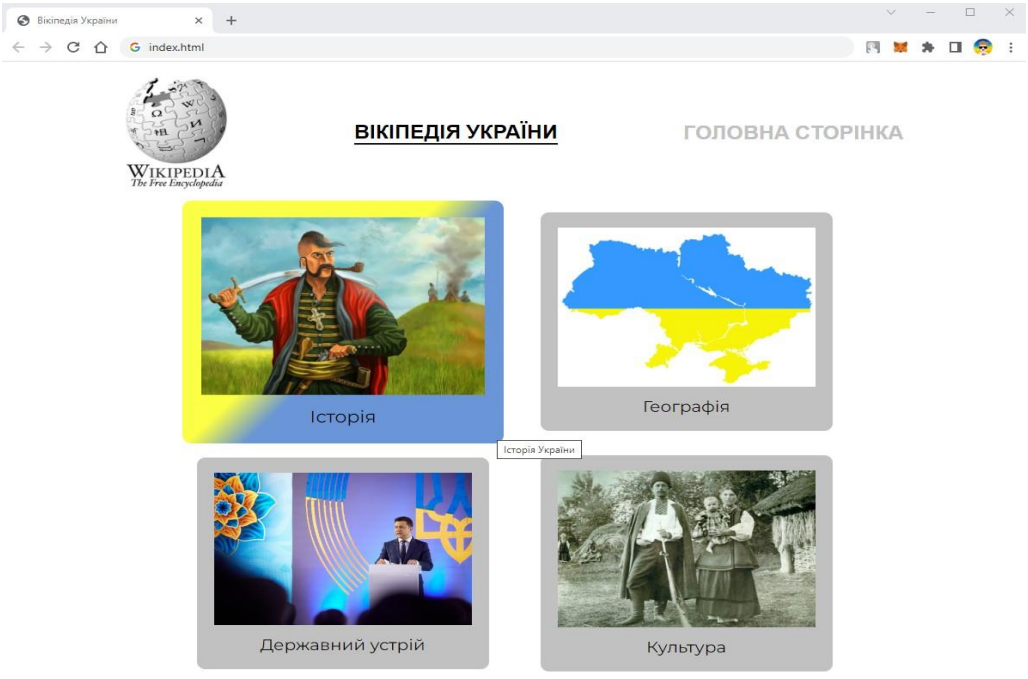
Історія - https://cutt.ly/8XuJrYJ
Географія - https://cutt.ly/iXuJiZ3
Державний устрій - https://cutt.ly/uXuJgcZ
Культура - https://cutt.ly/iXuJjB4

Для роботи скористайтеся уже готовим шаблоном сайту - завантажити всю папку можна у classroom.

Вигляд сайту при відкритті:



Вигляд сайту при наведенні на кнопку:



Завдання 3. Створити сторінку контактів із посиланнями на різні соцмережі.

Для усіх *посилань* на сторінці задати такі параметри:
відкриття у новій вкладці
прибрати стандартні текстові декорації

Ширину обмежети за допомогою *контейнера*:
максимальна ширина - 1000px
ширина - 100%
вирівнювання по центру за допомогою margin

Внутрішньому блоку, який містить в собі картки із посиланнями:
дисплей - флекс
розміщення коненту по блоку - простір між блоками
розміщення елементів - центр
направлення флекс-елементів - рядок
висота - 100vh

Картки *(містять іконку та підпис)*:
розміщення тексту по центру
ширина - 25%
висота - авто
кордони - #b4c2ed 2px суцільні
радіус кордонів - 20px
внутрішні відступи - 30px
перехід - 0.4s

При наведенні на картку:
колір фону - #829EEF
колір кордонів - #829EEF

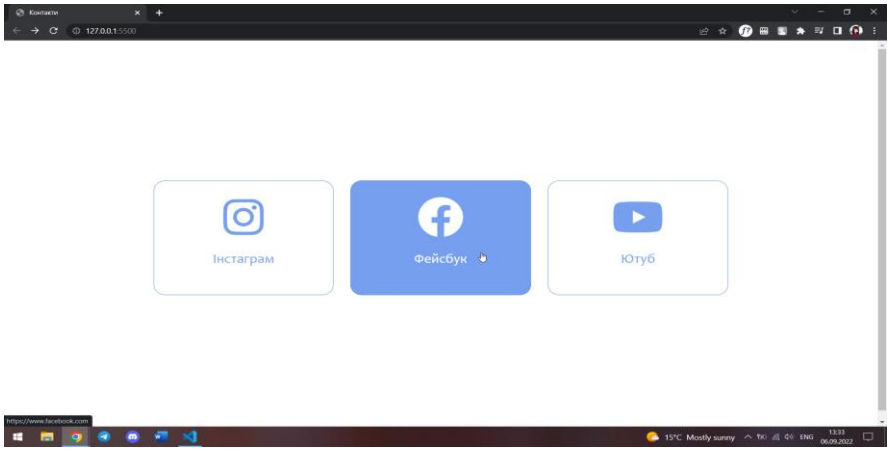
При наведенні на картку змінити іконку:
колір - білий

```
.card:hover>.icon::before {
    color: #fff;
}
```

***При наведенні* на картку змінити назву соцмережі за аналогією:**
колір - білий

Іконка соцмережі:
колір - #829EEF
розмір шрифта - 80px

Назва соцмережі:
шрифт - Candara
розмір шрифта - 25px
колір - #829EEF



Завдання 4.
Підключити шрифт **Raleway** за допомогою *Google Fonts*.
Підключити за допомогою скрипта сервіс *Font Awesome*.
Забрати усі стандартні зовнішні та внутрішні відступи усіх елементів.

Тіло документа:
шрифт - 'Raleway', sans-serif

HTML:
прокручування - повільне (*scroll-behavior: smooth;*)

Перший розділ сайту:
дисплей - флекс
розміщення контенту по центру (justify-content: center)
розміщення елементів по центру (align-items: center)
положення flex-блоків в flex-контейнері - стовпчик (flex-direction: column)
фон - зображення back.png
заборонити повторювання фону
розмір фону - покриття (cover)
висота - 100vh
розміщення тексту по центру
Заголовок та підзаголовок:
колір тексту - білий

Заголовок:
розмір шрифта - 40px

Підзаголовок:
розмір шрифта - 30px
зовнішній відступ зверху - 20px

Текст, виділений синім:
колір тексту - #829EEF
вага шрифта - 600

Кнопка:
дисплей - блок
ширина - 15%
вирівнювання по центру за допомогою margin
внутрішні відступи зверху та знизу - 10px
прибрати стандартні текстові декорації
колір тексту - білий
колір фону - #829EEF
радіус кордонів - 15px
зовнішній відступ зверху - 30px
перехід - 0.3 секунди

***При наведенні* на кнопку:**
колір фону - #5872c1

Для наступного розділу потрібно створити два блоки - *container* та *inner*. **Контейнер** обмежить ширину нашого контенту, **Іннер** дасть можливість поставити блоки так, як нам треба.
Контейнер:
максимальна ширина - 1200px
ширина - 100%
вирівнювання по центру за допомогою margin

Іннер:
висота - 50vh
вирівнювання тексту по центру
дисплей - флекс
розміщення контенту по центру (justify-content: center)
розміщення елементів по центру (align-items: center)
положення flex-блоків в flex-контейнері - рядок (flex-direction: row)

Картки, у яких розміщаються іконки**, головний текст та допоміжний текст**:
ширина - 30%

Псевдоелемент *:before* для іконок:
розмір шрифту - 80px

Головний текст карток:
зовнішні відступи зверху та знизу - 20px
розмір шрифту - 24px

Допоміжний текст карток:
колір - rgb(71, 71, 71)

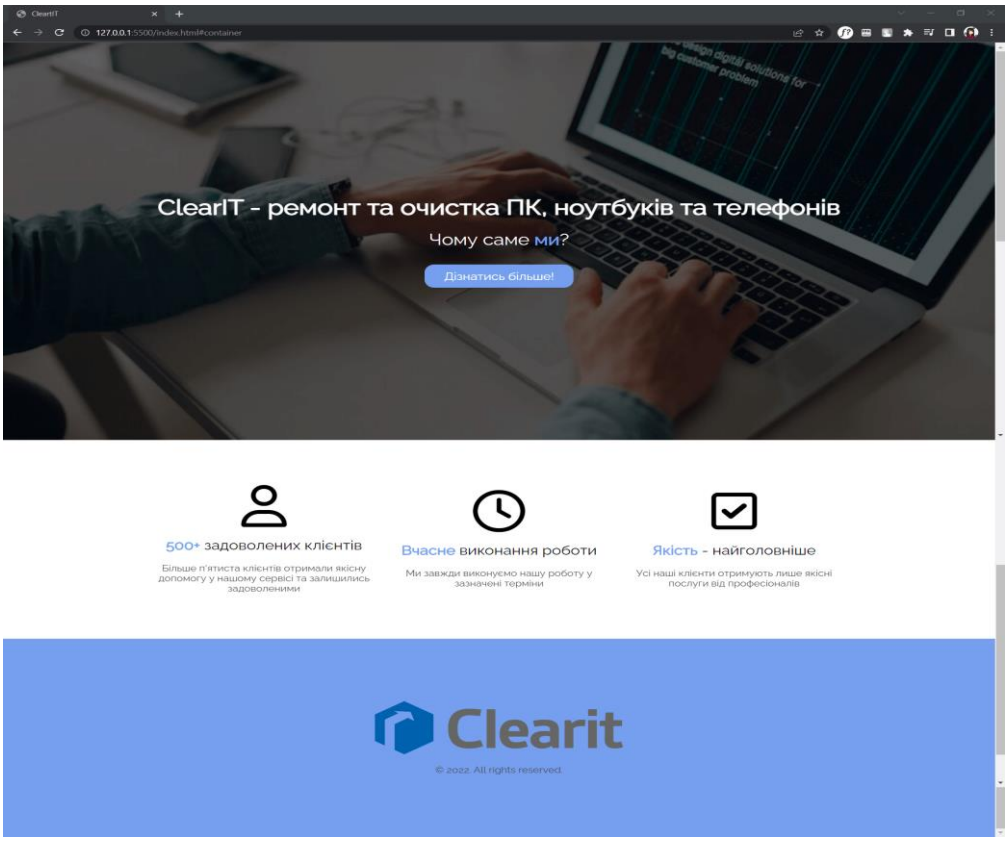
Футер *(останній розділ, що знаходиться внизу сторінки)*:
колір фону - #829EEF
висота - 50vh

Логотип:
ширина - 25%
зовнішній відступ знизу - 20px

Іннер всередині блоку Футер:
положення flex-блоків в flex-контейнері - рядок (flex-direction: column)

```
.footer>.inner {
    flex-direction: column;
}
```

Фото та текст для вебсторінки можна завантажити у classroom.



Звіт до практичного заняття № 6

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття № 7

Тема. Підсумкова робота 2.

Мета: узагальнити навички форматування елементів на web-сторінки з використанням мови опису HTML та CSS, технологія Flex.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Завдання для самостійної роботи

Завдання.

Вам потрібно створити головну сторінку *(index.html)* та сторінки для статтей *(post1.html, post2.html, post3.html).*

(для підсумкової роботи достатньо створити 1 статтю про Скелі Довбуша, текст та фото для неї уже є, та відео у classroom, а на домашнє завдання можете доробити й інші статті знайшовши інформацію про них в мережі інтернет самостійно).

Підключити ***Font Awesome***.

За допомогою онлайн-сервісу ***Google Font*** підключити шрифти '***Open Sans***' та '***Prosto One***'.

На ГОЛОВНІЙ СТОРІНЦІ є 4 основних секції з контентом.
Перша - головна, де міститься меню, заголовок і кнопка “Детальніше”
Друга - містяться три іконка та блок з текстом
Третя - пости на сайті
Четверта - блок з контактами.

Структура першої секції:

```
<section class=“intro”>
   навігація з посиланнями.nav__link

   <div class=“intro__inner”>
      заголовок
      кнопка (.intro__button href='#about')
</section>
```

Структура другої секції:

```
<section class=“about” id=“about”>
   <div class=“container”>
      <div class=“inner”>
         блоки із іконками.card
      </div>

      блок з текстом.about__text
   </div>
</section>
```

Структура третьої секції:

```
<section class=“blog” id=“blog”>
   <div class=“container”>
      підзаголовок.blog__subtitle
      <div class=“inner”>
         3 блоки.post із постами блогу
      </div>
   </div>
</section>
```

Структура четвертої секції:

```
<section class=“footer” id=“footer”>
   <div class=“container”>
      <div class=“inner”>
```

```
            посилання на соцмережі.link
          </div>
        </div>
    </section>
```

Усі стилі розділені на елементи сайту, до яких вони відносяться, щоб у коді було легше орієнтуватись.

***** Базові стилі *****

для усіх елементів забрати стандартні зовнішні та внутрішні відступи

Тіло документа:
шрифт - 'Open Sans', sans-serif

HTML:
прокручування - повільне

усім посиланням на сайті забрати стандартні текстові декорації

container:
максимальна ширина - 1200px
ширина - 100%
вирівнювання по центру за допомогою margin

inner:
висота - 50vh
дисплей - флекс
розділення флекс контенту по блоку - простір між блоками (space-between)
розміщення елементів - центр
направлення флекс контенту - рядок

***** Навігація *****

nav:
дисплей - флекс
направлення флекс контенту - рядок
розділення флекс контенту по блоку - рівномірний простір (space-evenly)
внутрішній відступ зверху - 20px
ширина - 50%
вирівнювання по центру за допомогою margin

Посилання в навігації:
вертикальне вирівнювання - верх

зовнішні відступи справа та зліва - 15px
позиція - відносна (position: relative)
колір шрифта - білий
розмір шрифта - 20px
перехід - для кольору 0.1 секунда, лінійний (transition: color 0.1s linear)

Посилання в навігації при наведенні на нього:
колір - #b2db96
***** Кнопки *****
Кнопки у блоках статтей (.button**):**
колір фону - #496f2f
внутрішні відступи:
зверху та знизу - 5px
справа та зліва - 10px
колір шрифта - білий
перехід - 0.4 секунди
Кнопка .intro__button**:**
внутрішні відступи:
зверху та знизу - 20px
справа та зліва - 30px
розмір шрифту - 25px
При наведенні на всі кнопки (.button та .intro__button)**:**
колір фону - #749261

***** Перша секція *****
intro:
фонове зображення - back.png
розмір фону - покриття

intro__inner:
дисплей - флекс
розміщення елементів - центр (align-items: center)
розділення флекс контенту по блоку - рівномірний простір (space-evenly)
висота - 100vh

Заголовок:
текстова трансформація - верхній регістр
шрифт - 'Prosto One', cursive
колір шрифта - білий
ширина - 40%
розмір шрифта - 80px

зовнішній відступ зліва - 100px
кордон справа - білий, 2px, суцільний

***** Друга секція *****
about:
висота - 100vh
фонове зображення - fly.png
заборонити повторювання фону
позиція фону - низ зліва
вирівнювання тексту по центру

Псевдоелемент before для іконок:
розмір шрифта - 80px

Головний текст під іконками:
зовнішні відступи зверху та знизу - 20px
розмір шрифта - 24px

Допоміжний текст під іконками:
колір шрифта - #474747

Блок, в якому знаходиться іконка та текст:
ширина - 30%

span в тексті під іконками:
колір - #2d5415
вага шрифта - 600

Блок з текстом на зеленому фоні:
ширина - 40%
розмір шрифта - 20px
розміщення справа (float: right)
колір фону - #b2db96
внутрішні відступи - 30px
вирівнювання тексту по лівій стороні

***** Третя секція *****
blog:
колір фону - #b2db96
висота - 70vh

Підзаголовок:
розмір шрифта - 30px
внутрішній відступ зверху - 30px
внутрішній відступ знизу - 30px

Блоки із статтями:
колір фону - білий
внутрішні відступи - 30px
ширина - 25%
висота - 330px

Зображення всередині блоків із статтями:
ширина - 100%
висота - 150px

Заголовок блоків зі статтями:
зовнішні відступи зверху та знизу - 10px

Опис блоків зі статтями:
зовнішній відступ знизу - 20px

***** Четверта секція *****
Посилання в футері:
ширина - 100%
колір шрифта - #2d5415
вирівнювання тексту по центру

Текст всередині посилань:
зовнішній відступ зверху - 10px
розмір шрифта - 20px

При наведенні на посилання:
колір шрифта - #719659

Іконки соцмереж:
вирівнювання тексту по центру
вирівнювання блоку по центру за допомогою margin
розмір шрифта - 60px

Сторінка статті.
Структура сторінки статті:

<nav class=“nav__post”> *(тут додано клас до навігації, адже треба задати стилістичні зміни)*
</nav>

<section class=“post__main”>

головне зображення статті.post__main__img

заголовок статті.post__title
текст статті.post__text

параграф з іншим текстом.post__text
ще один заголовок статті.post__title
зображення статті.post__img

параграф з іншим текстом.post__text
параграф з іншим текстом.post__text
відео.post__video

</section>

nav__post:
внутрішній відступ знизу - 250px

nav__post a:
колір - чорний

post__main:
зовнішній відступ зверху - 40px

Головне зображення статті:
ширина - 45%
дисплей - лінійний блок

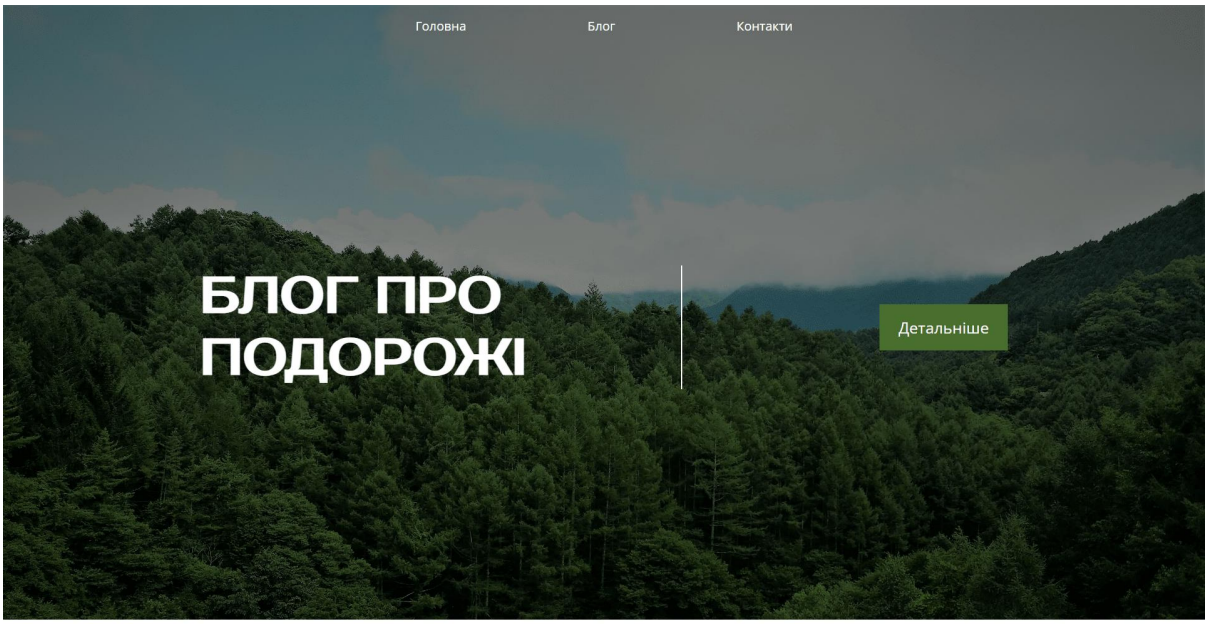
wrapper:
дисплей - лінійний блок
вертикальне вирівнювання - верх
ширину - 50%
зовнішній відступ зліва - 30px

Заголовок статті:
шрифт - 'Prosto One', cursive
зовнішній відступ знизу - 10px

Зображення статті:
дисплей - блок
вирівнювання по центру за допомогою margin
внутрішні відступи зверху та знизу - 15px

Відео:
дисплей - блок
вирівнювання по центру за допомогою margin
зовнішні відступи
зверху - 30px
знизу - 50px

Текст та папку assets для вебсторінки можна отримати **у classroom.**
Фото головної сторінки:



Нові маршрути

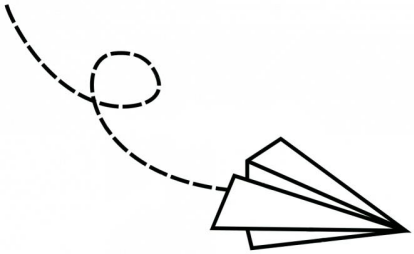
Іноді важко знайти інформацію про країну чи місто. Цей блог допоможе дізнатись про найцікавіші маршрути подорожей!

Від сходу до заходу

На цьому сайті зібрані розповіді про найрізноманітніші точки світу, - від холодних гір до морських пляжів.

По всьому світі :)

Досвідчені туристи щодня вивчають цей світ і розповідають про всі незвичайні речі саме тут!



Подорожі допомагають нам відпочити від побуту, забути про роботу, струсити з себе стрес і проблеми, а також переоцінити своє життя. Відвідуючи інші країни або міста, відпочиваючи на курортах, плескаючись в теплому морі, ми по-новому оцінюємо світ, а також краще пізнаємо себе. Лише в подорожах виходить повністю розслабитися, відкритися і впорядкувати свої думки.

Що почитати?

Скелі Довбуша

Основний скельний масив має площу понад двохсот метрів на один кілометр. Але тут багато і інших окремих скель.

Перейти до статті

Хортиця

Невеликий острів на могутньому Дніпрі з ключовою роллю в українській історії. Хортиця є одним із Семи чудес України.

Перейти до статті

Підгорецький замок

Підгорецький замок належить не тільки до найбільш визначних архітектурних пам'яток Львівщини, а й усієї України.

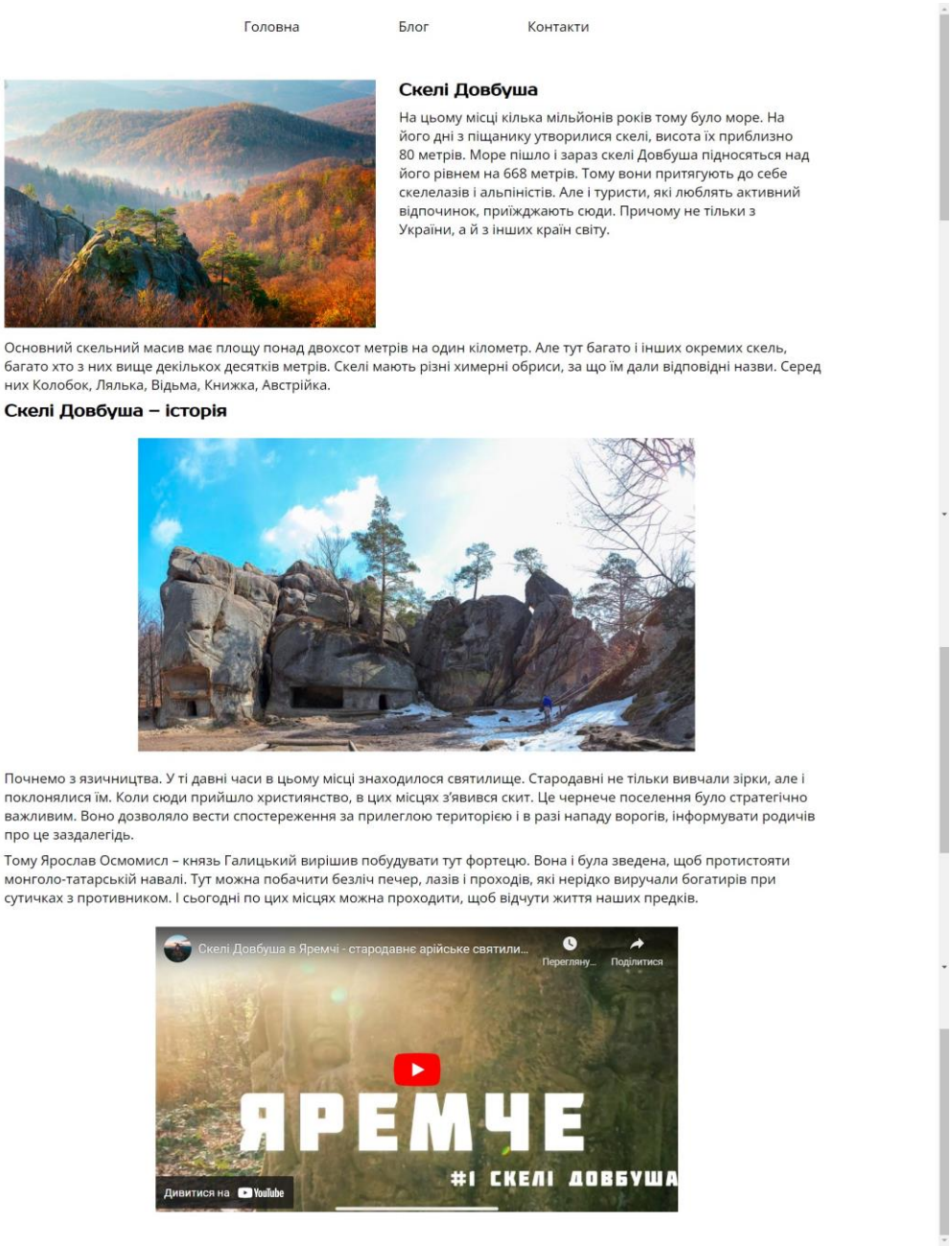
Перейти до статті

Інстаграм

Фейсбук

Ютуб

Фото сторінка статті:



Звіт до практичного заняття №7

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №8

Тема. Робота з таблицями за допомогою HTML

Мета: ознайомлення з основними методами створення таблиць та її форматування.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Теоретичні відомості

table- створює таблицю. Будь-яка таблиця складається з рядків і комірок, які задаються за допомогою тегів <tr>, <th> і <td>

<tr> - створює рядок таблиці

<th> - призначений для створення однієї комірки таблиці, яка позначається як заголовна (текст всередині буде жирним і більшим)

<td> - призначений для створення однієї комірки таблиці (звичайної)

<tbody> - призначений для зберігання одного чи кількох рядків таблиці. Це дозволяє створювати структурні блоки, до яких можна застосовувати єдине оформлення через стилі, а також керувати їх видом через скрипти.

Допускається застосовувати кілька тегів <tbody> всередині контейнера <table> . Доступні і інші види угруповань рядків — <tfoot> і <thead> , жоден з них не повинен перекриватися з елементом <tbody> .

Синтаксис:

```
<table>
<thead> .... </thead>
<tfoot> ... </tfoot>
<tbody>
<tr>
   <td> ... </td>
</tr>
</tbody>
</table>
```

<caption> - призначений для створення заголовка до таблиці і може розміщуватися тільки всередині контейнера <table> , причому відразу після тега. Такий заголовок є текст, що за замовчуванням відображається перед таблицею і описує її зміст.

cellspacing - встановлює відстань між зовнішніми кордонами комірок.

Псевдоклас :nth-child - використовується для додавання стилю до елементів на основі їх порядку в html. Наприклад якщо є підряд 5 рядків <td> а нам потрібно звернутися до 3, то щоб не створювати додаткових класів можна просто написати td :nth-child(3) - і прописати властивості застосуються тільки до нього.

Синтаксис

елемент: nth-child (odd | even | <число> | <вираз>) {...}

Значення

odd - всі непарні номери елементів.

even - всі парні номера елементів.

число - порядковий номер дочірнього елемента відносно свого батька. Нумерація починається з 1, це буде перший елемент у списку.

вираз - здається в вигляді an + b, де a і b цілі числа, a n - лічильник, який автоматично приймає значення 0, 1, 2 ...

Якщо a дорівнює нулю, то воно не пишеться і запис скорочується до b. Якщо b дорівнює нулю, то воно також не вказується і вираз записується в формі an . a і b можуть бути негативними числами, в цьому випадку знак плюс змінюється на мінус, наприклад: 5n-1.

Нище наведені деякі можливі вирази і ключові слова, а також зазначено, які номери елементів будуть задіяні.

Значення	Номери елементів	Опис
1	1	Перший елемент, є синонімом псевдокласу : first-child .
5	5	П'ятий елемент.
2n	2, 4, 6, 8, 10	Все парні елементи, аналог значення even .
2n + 1	1, 3, 5, 7, 9	Все непарні елементи, аналог значення odd .
3n + 2	2, 5, 8, 11, 14	—
-n + 3	3, 2, 1	—
5n-2	3, 8, 13, 18, 23	—
even	2, 4, 6, 8, 10	Все парні елементи.
odd	1, 3, 5, 7, 9	Все непарні елементи.

Завдання для самостійної роботи

Завдання 1.

Створити таблицю, як показано на скріншоті. Задати таблиці клас “time”.

За допомогою атрибуту colspan створити довшу комірку напроти в дня “Понеділок”.

В CSS задати такі параметри:

time:

колір фону - #eae5e0

максимальна ширина - 500px

ширина - 100%

вирівнювання по центру за допомогою margin

розміщення тексту по центру

Для тегу рядків (tr) використати селектор ***nth-child*** - *за допомогою нього можна автоматично задавати різні стилі для певних рядків таблиці.*

В нашому випадку ми будемо змінювати колір фону для кожного другого рядку таблиці. Для цього в дужках нам потрібно вказати параметр “**2n**”, тобто кожен другий.

```
tr:nth-child(2n) {
   background-color: #d2c0ae;
```

}

th (комірки-заголовки) та td (звичайні комірки):

ширина - 10em

внутрішні відступи - 15px

кордон - #6a5757, 1px, штриховий

Макет сайту для роботи можна отримати ТУТ НАТИСНИ

	Робочий час	Обідня перерва
Понеділок	Вихідний	
Вівторок	9:00 - 18:00	13:00 - 13:30
Середа	9:00 - 18:00	13:00 - 13:30
Четвер	9:00 - 17:00	13:00 - 13:30
П'ятниця	9:00 - 18:00	13:00 - 13:30
Субота	9:00 - 20:00	13:00 - 13:30
Неділя	9:00 - 20:00	13:00 - 13:30

Завдання 2.

Підключити шрифт "**Open Sans**" за допомогою онлайн-сервісу Google Fonts.

Створити таблицю з розкладом ваших уроків, як показано на скріншоті.

Кольори обрати самостійно за власним вподобанням.

За потреби можете додати рядків або стовпців.

Для коректного вирівнювання таблиці по центру, необхідно створити блок "container", а в ньому - таблицю з класом "inner".

В таблиці додати атрибут cellspacing зі значенням "0".

Рядку (tr) із переліком днів задати клас "days". Також усім коміркам в стовпці з нумерацією додати клас "number", всім коміркам в стовпці годин - клас "time".

В CSS задати такі параметри:

Забрати усі автоматичні відступи за допомогою коду:

```
*,
*:before,
*:after {
margin: 0;
padding: 0;
}
```

Тіло документа:

задати власний колір фону (або залишити білим)

шрифт - 'Open Sans', sans-serif

container:
максимальна ширина - 1200px
ширина - 100%
вирівнювання по центру за допомогою margin

inner:
дисплей - флекс
вирівнювання елементів - центр
розподілення контенту по сторінці (justify-content) - центр
висота - 100vh

Для рядків застосувати селектор *nth-child(2n+3)* - таким чином ми пропускаємо перший рядок, де в нас вказані дні тижня, та починаємо фарбувати в потрібний колір кожен другий рядок НЕ враховуючи рядок днів:
колір фону - білий

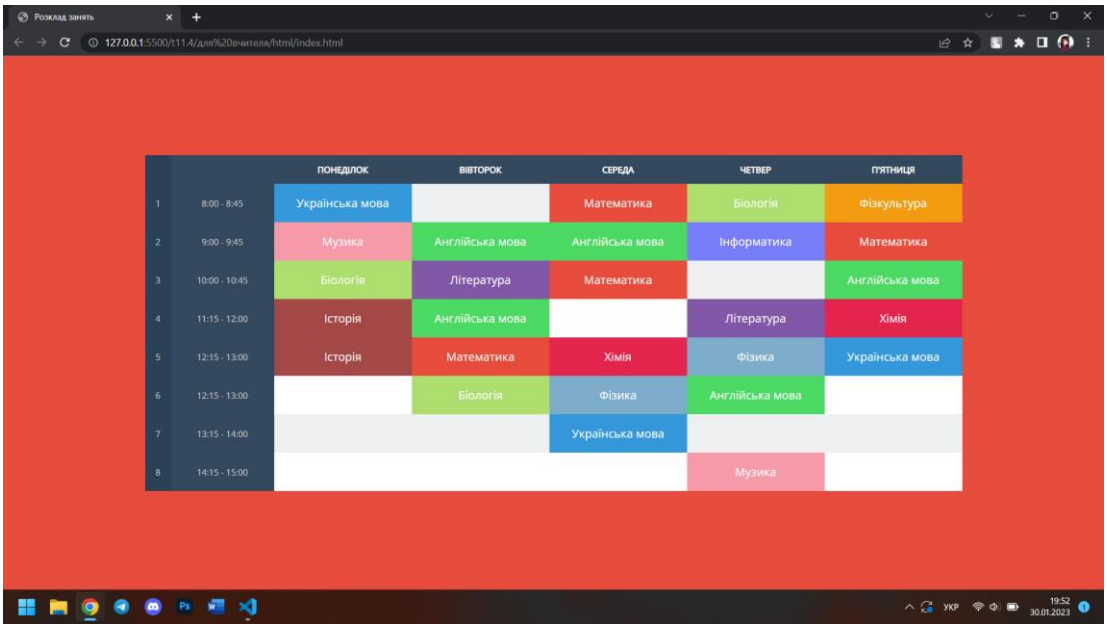
Для рядків застосувати селектор *nth-child(2n)* - в цьому варіанті ми фарбуємо в потрібний колір кожен другий рядок, враховуючи рядок із днями:
колір фону - #eff0f1

Тег комірок-заголовків та комірок:
внутрішні відступи - 1em
ширина - 10em
розміщення тексту по центру
колір тексту - білий

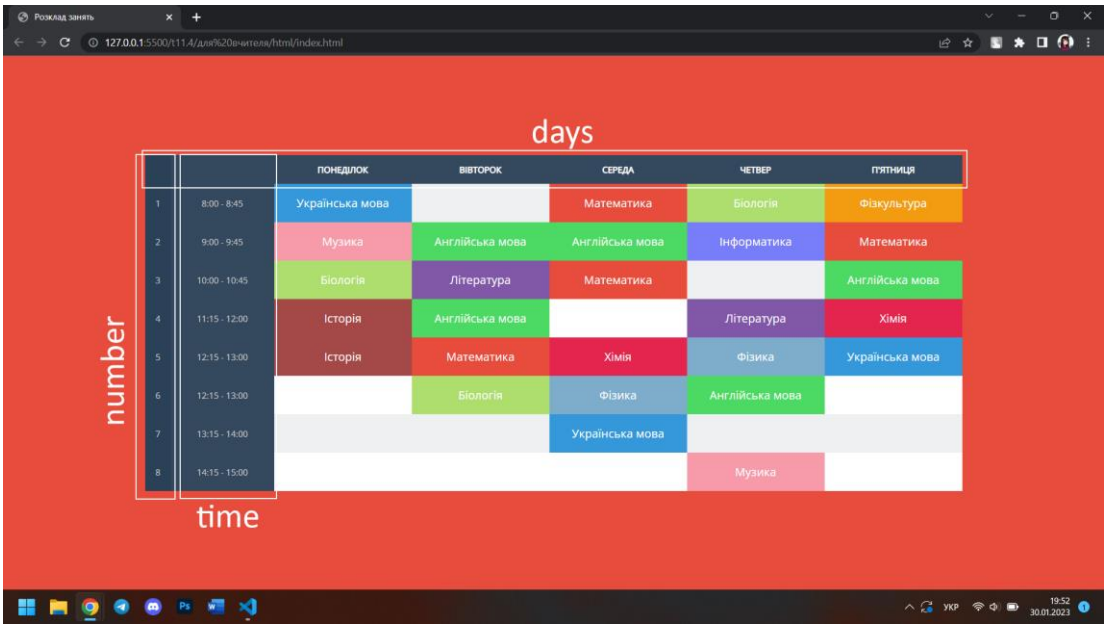
days, time, number:
колір фону - #34495e
колір тексту - #cbcbcb
розмір шрифта - 12px
текстова трансформація - верхній регістр

number:
колір фону - #2b4157
внутрішні відступи - 0em !important
ширина - 3em

Для кожного із ваших предметів вигадати назву класу та в CSS задати колір фону.



Назви класів у рядках та стовпцях.



Звіт до практичного заняття №8

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №9

Тема. Поглиблине вивчення Flex-ів.

Мета: продовжувати вивчати технологію Flex-ів.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Теоретичні відомості

Основна (найпростіша) структура використання Flex-ів :

У файлі .html

```html
<body>
    <section class="container">
        <div class="inner">
            <!-- 1 елемент, що буде флексом -->
            <!-- 2 елемент, що буде флексом -->
            <!-- n елемент, що буде флексом -->
        </div>
    </section>
</body>
!УВАГА! Флексами можуть бути такі елементи,
як просто блоки (теги div), покликання (теги a),
зображення (теги img) та інші
```

У файлі .css

```css
/* Базові налаштування секції: */
.container {
    max-width: 1200px; /* налаштування максмальної ширини (якщо потрібно) */
    width: 100%; /* розтягування вмісту на всю ширину */
    margin: 0 auto; /* вирівнювання по центрі (якщо потрібно) */
}
/* Головні налаштування, щоб запрацювали флекси */
.inner {
    display: flex; /* перетворення елементів на флекси */
    flex-direction: row; /* щоб флекси стояли у рядок,
    якщо потрібно у стовпчик то значення буде: column */
    align-items: center; /*  вирівнювання по висоті */
    justify-content: space-between; /* вирівнювання по ширині  */
}
.елемент {
    /* стилізація елементів */
}
```

Властивості стилів:

flex-direction - напрямок розташування флексів

Можливі значення:

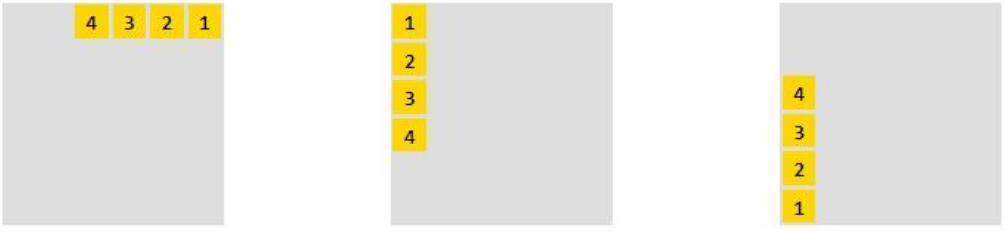
row - розташування у рядок (зліва на право)

row-reverse - розташування у рядок дзеркально (справа на ліво)

column - розташування у стовпчик (вертикально) (зверху вниз)

column-reverse - розташування у стовпчик дзеркально (вертикально) (знизу вверх)

Ви бачите три ідентичних блоки, але для них задані різні напрямки осі: *row-reverse*, *column* й *column-reverse* відповідно.



align-items - вирівнювання флексів по вертикалі

Можливі значення:

stretch - елементи розтягуються та займають весь доступний простір контейнера.

center - елементи розташовані по центру контейнера.

flex-start - елементи розташовані на початку контейнера.

flex-end - елементи розташовані в кінці контейнера.

baseline - елементи розташовані на базовій лінії контейнера.

Ви бачите два ідентичні контейнери, але для них задані різні значення для вирівнювання: *flex-start*, *center* й *flex-end* відповідно.

justify-content - вирівнювання флексів по горизонталі

Можливі значення:

flex-start - елементи розташовані на початку контейнера.

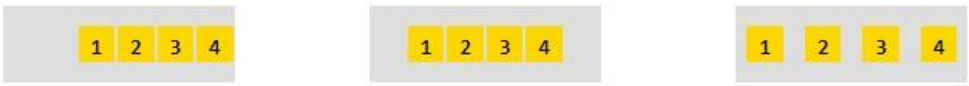
flex-end - елементи розташовані в кінці контейнера.

center - елементи розташовані в центрі контейнера.

space-between - флекси рівномірно розподіляються по всьому рядку. Перший і останній флекс притискаються до відповідних країв контейнера.

space-around - флекси рівномірно розподіляються по всьому рядку. Порожній простір перед першим і після останнього елементів дорівнює половині простору між двома сусідніми елементами.

Ви бачите три ідентичних блоки, але для них задані різні значення для justify-content: *flex-end*, *center* й *space-around* відповідно.



Завдання для самостійної роботи

Завдання 1.

Структура HTML:

```
<header class="header">
    <nav>

    </nav>
</header>

<section class="intro">
    <div class="container">
        <div class="intro__inner">
```

```
        </div>
      </div>
    </section>

    <section class="about__first">
      <div class="container">
        <div class="about__inner">

        </div>
      </div>
    </section>

    <section class="about__second">
      <div class="container">
        <div class="about__inner">

        </div>
      </div>
    </section>
```

CSS:

Необхідно локально підключити три шрифт - *GothamRounded-Bold, GothamRounded-Medium, GothamRounded-Book.*

Для всіх елементів забираємо автоматичні відступи.

container:

максимальна ширина - 1000px
ширина - 100%
вирівнювання по центру за допомогою margin

Елемент посилання:

декорація тексту - відсутня
колір тексту - білий

Кнопка у нас має один і той самий дизайн, тому прописуємо її стиль лише один раз, а в HTML кожній кнопці задаємо один і той самий клас.

btn__green:

колір фону - #16a085
внутрішні відступи зверху та знизу - 20px
внутрішні відступи справа та зліва - 45px
шрифт - GothamRounded-Bold
колір тексту - білий
трансформація тексту - верхній регістр
кордони - відсутні

заокруглення кордонів - 5px
перехід - all 400ms

При наведенні на btn__green:
колір фону - #59e4c8
колір тексту - чорний

Елемент навігації:
дисплей - флекс
напрямок флекс елементів - рядок
вирівнювання елементів всередині рядків - центр
розподіл контенту по блоку - центр

nav__link *(клас кожного посилання в навігації):*
шрифт - 16px, GothamRounded-Bold
трансформація тексту - верхній регістр
зовнішні відступи*[1] - 0 20px
перехід - color 400ms

*[1] *відступи працюють за принципом годинникової стрілки. Можна не вказувати окремим рядком для якої сторони вводиться значення відступу, натомість написати це все одним рядком прописавши просто "margin".*

Коли ми вказуємо два числа - це означає зверху та знизу (перше число) і справа та зліва (друге число).

Якщо ми вказуємо чорити числа - це означає для всіх чотирьох сторін в такому порядку: верх, справа, знизу, зліва.

При наведенні на nav__link:
колір тексту - #16a085

Логотип:
ширина - 3em
зовнішні відступи - 0 20px

header:
ширина - 100%
внутрішній відступ зверху - 40px
позиція - абсолютна
top: 0;
left: 0;
right: 0;

z-index: 1000;

intro:
ширина - 100%
висота - 100vh
фон - зображення background.png, внизу, без повторень
розмір фону - покриття
дисплей - флекс
напрямок флекс елементів - стовпець
розподіл контенту по блоку - центр
вирівнювання тексту по центру

intro__inner:
максимальна ширина - 520px
ширина - 100%
вирівнювання по центру за допомогою margin
відступи між буквами в тексті (letter-spacing) - 0.2em

Заголовок в intro:
розмір шрифта - 36px
трансформація тексту - верхній регістр
зовнішній відступ знизу - 10px

Підзаголовок в intro:
шрифт - 16px, 'GothamRounded-Book'
зовнішній відступ знизу - 30px

Блок з іконками-посиланнями на соцмережі:
ширина - 100%
позиція - абсолютна
bottom: 0;
left: 0;
z-index: 1;
внутрішній відступ знизу - 20px

links__inner *(всередині попереднього блоку):*
максимальна ширина - 150px
ширина - 100%
вирівнювання по центру за допомогою margin
дисплей - флекс
напрямок флекс елементів - рядок

розподіл контенту по блоку - рівномірно

Іконки:
розмір шрифта - 24px
перехід - color 400ms

При наведенні на іконки:
колір шрифта - #16a085

about__first:
фон - зображення about_background1.png, центр, без повторень
розмір фону - покриття

about__second:
фон - зображення about_background2.png, центр, без повторень
розмір фону - покриття

about__inner:
дисплей - флекс
розподіл контенту по блоку - флекс-початок

Для блоку about__second необхідно окремо задати вирівнювання по іншому кінцю сторінки, щоб всі елементи були справа:

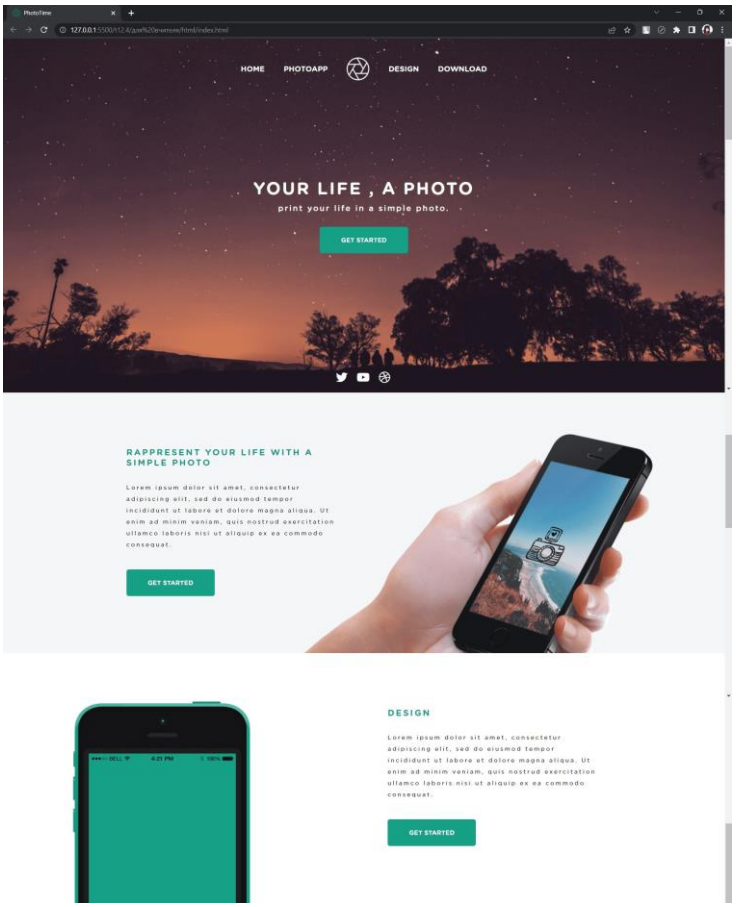
```
.about__second>.container>.about__inner {
   justify-content: flex-end;
}
```

about__item - блок із заголовком, текстом та кнопкою:
ширина - 100%
відступи між буквами в тексті (letter-spacing) - 0.2em
внутрішні відступи - 120px 0
максимальна ширина - 450px
ширина - 100%

Заголовок в блоці about__item:
колір - #16a085
шрифт - 18px, 'GothamRounded-Bold'
трансформація тексту - верхній регістр
зовнішній відступ знизу - 30px

Текст в блоці about__item:
шрифт - 12px, 'GothamRounded-Book';

зовнішній відступ знизу - 40px
висота лінії (line-height) - 24px
Матеріали для вебсайту можна завантажити у classroom.



Завдання 2. *Продовження створення сайту із завдання 1.*
Структура HTML:

```
<section class="community">
    <div class="container">
        <div class="community__inner">

        </div>
    </div>
</section>

<section class="download">
    <div class="container">
        <div class="download__inner">

        </div>
    </div>
</section>

<section class="footer">
```

```
    <div class="container">
      <div class="footer__inner">

      </div>
    </div>
  </section>
```

CSS:

community:
ширина - 100%
колір фону - #f5f6f7
внутрішні відступи - 65px 0

community__inner:
ширина - 100%
дисплей - флекс
напрямок флекс елементів - рядок
вирівнювання елементів всередині рядків - центр
розподіл контенту по блоку - простір між блоками

community__text - блок із логотипом, заголовком та коротким текстом:
максимальна ширина - 200px
ширина - 100%
дисплей - флекс
напрямок флекс елементів - стовпчик
вирівнювання тексту по центру
вирівнювання елементів всередині рядків - центр
відступи між буквами в тексті (letter-spacing) - 0.2em
колір шрифта - #373737

Заголовок в блоці community__text:
трансформація тексту - верхній регістр
шрифт - 16px, 'GothamRounded-Bold'
зовнішні відступи - 20px 0 20px 0

Підзаголовок в блоці community__text:
шрифт - 10px, 'GothamRounded-Medium'

Логотипу в цьому блоці задаємо такий самий клас, як і логотипу на верху сторінки.

community__photo - блок із двома фотографіями:
ширина - 100%
дисплей - флекс
напрямок флекс елементів - рядок
розподіл контенту по блоку - флекс-кінець

Фото в блоці community__photo:
ширина - 40%
зовнішні відступи - 0 10px

download__inner:
максимальна ширина - 850px
ширина - 100%
вирівнювання тексту по центру
вирівнювання по центру за допомогою margin
колір шрифта - #373737
відступи між буквами в тексті (letter-spacing) - 0.2em
трансформація тексту - верхній регістр
внутрішні відступи - 100px 0

Заголовок (DOWNLOAD IT):
шрифт - 20px, 'GothamRounded-Bold'
зовнішній відступ знизу - 15px

Підзаголовок:
шрифт - 12px, 'GothamRounded-Medium'
зовнішній відступ знизу - 50px

Блок із двома кнопками:
максимальна ширина - 600px
ширина - 100%
вирівнювання по центру за допомогою margin
дисплей - флекс
напрямок флекс елементів - рядок
розподіл контенту по блоку - рівномірно

footer:
внутрішні відступи - 50px 0
колір фону - #f5f6f7

footer__inner:

дисплей - флекс
напрямок флекс елементів - рядок
розподіл контенту по блоку - рівномірно
вирівнювання елементів всередині рядків - центр
шрифт - 14px, 'GothamRounded-Bold'
трансформація тексту - верхній регістр

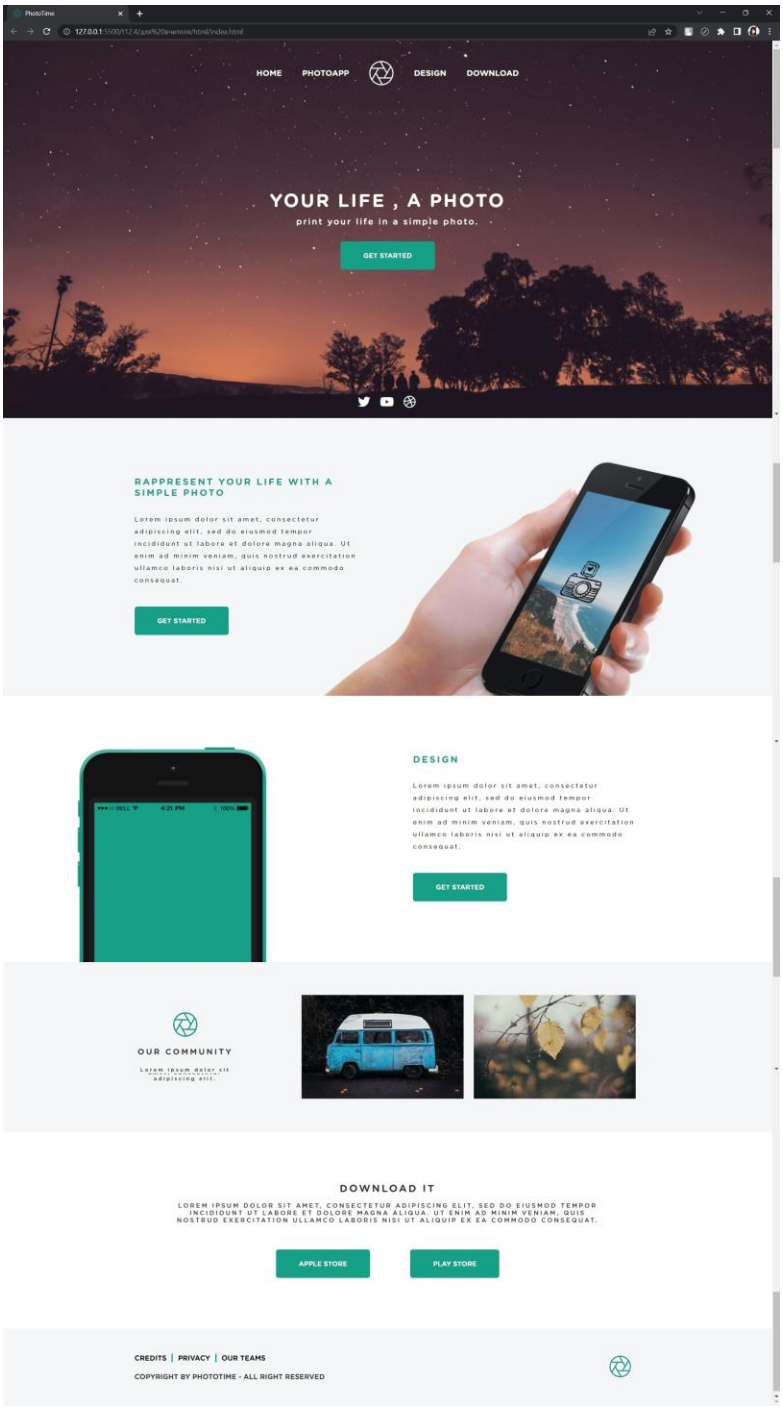
Всередині footer__inner є ще два блоки - footer__text (де знаходяться посилання та текст) та footer__logo з логотипом.

footer__text:
максимальна ширина - 400px
ширина - 100%
дисплей - флекс
напрямок флекс елементів - рядок
перенесення флекс елементів - wrap

footer__btn (елементи посилання з текстом CREDITS, PRIVACY, OUR TEAMS):
колір шрифта - чорний
внутрішні відступи - 0 10px
кордони справа - #16a085 3px, суцільні
зовнішній відступ знизу - 20px

Перший елемент footer__btn:
внутрішній відступ зліва - 0

Останній елемент footer_:_btn
кордни - відсутні



Звіт до практичного заняття №9

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №10

Тема. Стилізація меню + фрейми основи

Мета: ознайомитися з поняттям фрейм, як важливим інструментом у веброзробці, навчитися стилізувати меню.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Теоретичні відомості

Про позиціонування елементів. Z-індекс

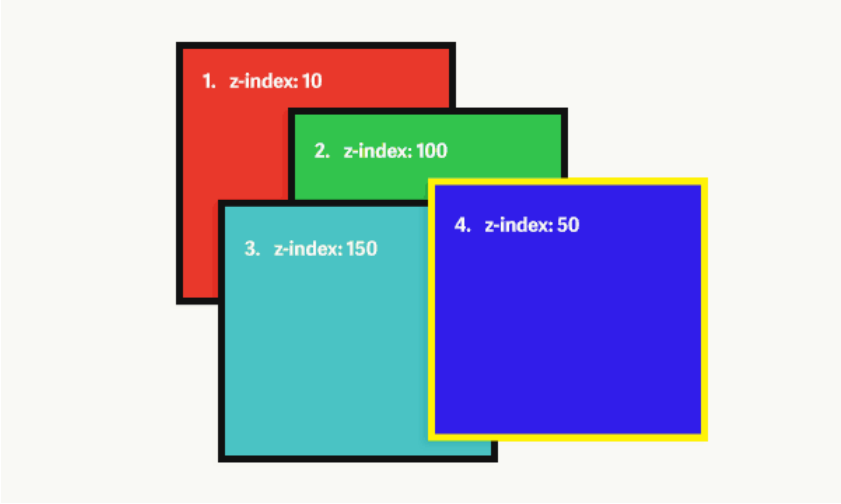
Значення z-index, top, right і left в CSS використовуються для позиціонування елементів на веб-сторінці. Ось докладне пояснення кожного з них:

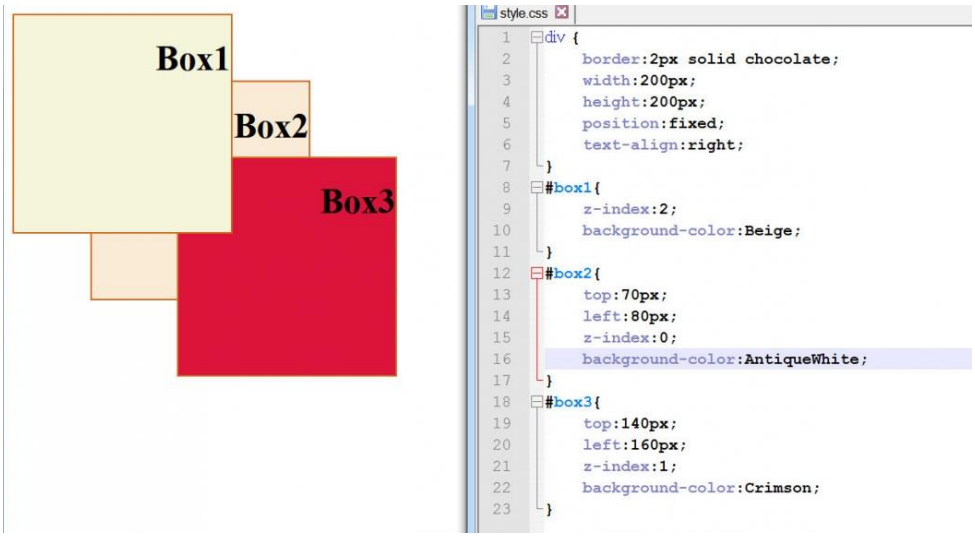
1. **z-index:** Визначає, як елемент розташовується на з-осі (вертикальна вісь) відносно інших елементів. Якщо ви маєте два елементи, які перекриваються, ви можете використовувати z-index, щоб вказати, який елемент повинен знаходитися вище і який нижче. Елемент з більшим значенням z-index буде відображатися поверх елементів з меншим значенням z-index. Значення z-index можуть бути позитивними, від'ємними або нульовими числами.
2. **top:** Визначає відстань від верхнього краю контейнера, в якому розміщений елемент. Наприклад, якщо ви задаєте top: 50px; для елемента, то його верхній край буде зміщено від верхнього краю контейнера на 50 пікселів.
3. **right:** Визначає відстань від правого краю контейнера, в якому розміщений елемент. Наприклад, якщо ви задаєте right: 50px; для елемента, то його правий край буде зміщено від правого краю контейнера на 50 пікселів.
4. **left:** Визначає відстань від лівого краю контейнера, в якому розміщений елемент. Наприклад, якщо ви задаєте left: 50px; для елемента, то його лівий край буде зміщено від лівого краю контейнера на 50 пікселів.

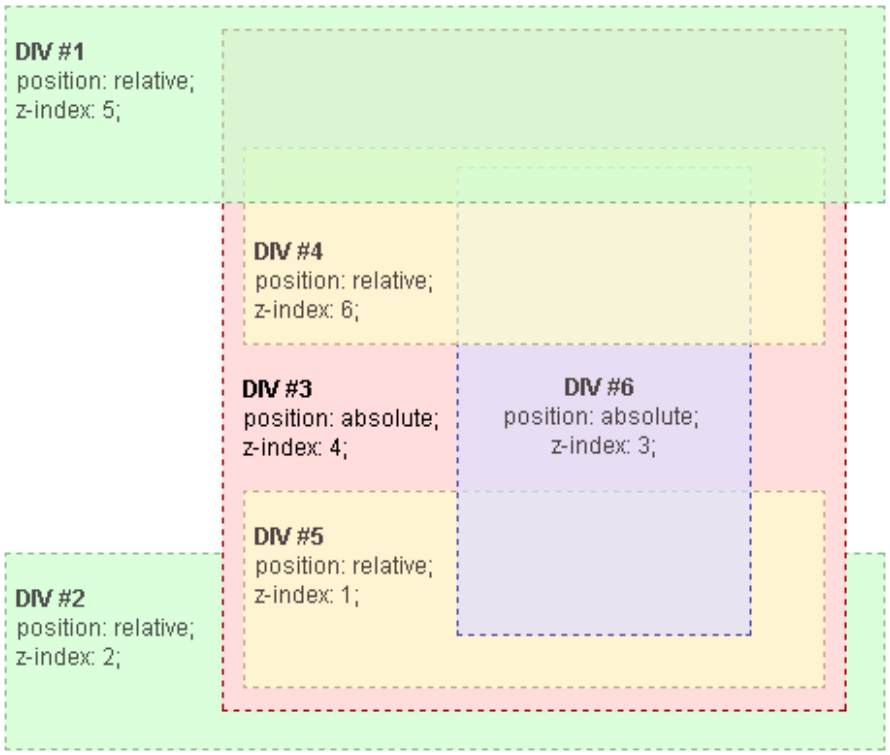
Зверніть увагу, що top, right і left працюють лише для елементів, для яких була встановлена **position** на значення, відмінне від static. Наприклад, якщо ви задаєте position: relative; для елемента, то top, right і left будуть відноситься до його вихідної позиції в документі.

Отже, z-index, top, right і left є важливими властивостями для позиціонування елементів на веб-сторінці. Завдяки ним ви можете визначити, як елементи будуть розташовуватися відносно інших елементів та який елемент буде відображатися поверх іншого. Пам'ятайте, що ці властивості працюють тільки для елементів, для яких була встановлена position на значення, відмінне від static.

Приклади застосування :







Фрейми та **iframe** є важливими інструментами для розробки веб-сторінок, які дозволяють вбудовувати вміст інших веб-сторінок всередині поточної сторінки. Давайте детальніше розберемося, як це працює та чому це є зручним і корисним.

1. Фрейми (Frames): Фрейми – це старіший підхід *(його ми не використовуємо, але для ознайомлення вам варто про це знати)* до вбудовування змісту інших вебсторінок. Вони дозволяють поділити вікно браузера на кілька незалежних областей, кожна з яких може завантажувати окрему вебсторінку. У HTML це реалізується за допомогою тегу **<frameset>**, який визначає розміри та розташування фреймів, та тегу <frame>, який вказує вміст для кожного фрейму.

Проте використання фреймів має кілька ***недоліків:***

- Поганий для SEO: Пошукові системи не завжди добре індексують зміст у фреймах, що може негативно позначитися на позиціях в пошукових результатах.

- Складність підтримки: Деякі старі браузери та пристрої можуть погано підтримувати фрейми, що призводить до проблем з відображенням вмісту.

2. **iframe**: iframe (Inline Frame) - це сучасний підхід до вбудовування вмісту інших вебсторінок в поточну сторінку. Він реалізується за допомогою тегу <iframe>, який дозволяє створювати вбудовану область на сторінці, де можна завантажити окрему вебсторінку. Один і той самий вміст може бути вбудований на декілька сторінок з допомогою iframe.

Переваги *використання iframe:*

- Простота використання: iframe легко впроваджувати - вам потрібно лише вказати URL сторінки, яку ви хочете вбудувати, та налаштувати розміри iframe.
- Незалежність контенту: Вміст, вбудований через iframe, не впливає на решту сторінки. Це дозволяє розділяти веб-сторінку на частини з різним контентом, при цьому кожна частина може завантажувати власний вміст.
- Підтримка SEO: Пошукові системи краще розуміють індексують контент, що вбудований через iframe, ніж у випадку з фреймами.
- Динамічність: Завантажений контент в iframe можна змінювати динамічно, без перезавантаження всієї сторінки.

Зважайте на те, що використання iframe також має свої недоліки:

- Безпека: Якщо iframe використовується для завантаження контенту з ненадійного джерела, це може представляти загрозу для безпеки, так як він може мати доступ до даних і взаємодіяти зі сторінкою, на якій він вбудований.
- Розділений контент: Використання занадто багатьох iframe на сторінці може зробити її менш читабельною та складною в обслуговуванні.

У нашому прикладі, що є у 3 завданні, ми використовуємо iframe для вбудовування сторінок "page1.html", "page2.html" і т.д. всередині контейнера з id="content". Це дозволяє нам динамічно змінювати вміст контейнера без перезавантаження всієї сторінки, що є зручним і швидким способом створення багатосторінкового сайту.

На сьогоднішній день, у зв'язку з обмеженнями безпеки і бажанням покращити SEO, iframe зазвичай використовується з розумом та обережністю. Важливо зберігати баланс між зручністю та безпекою, добре зважаючи, де і як використовувати iframe на вашому сайті.

Завдання для самостійної роботи

Завдання 1. Створити меню за допомогою ненумерованого списку, всередині кожного елементу списку прописати тег посилання.

CSS:

Забрати усі стандартні відступи

Тіло документа:

дисплей - флекс

вирівнювання елементів всередині рядків - центр

розподіл контенту по блоку - центр

висота - 100vh

колір фону - #5aabe4

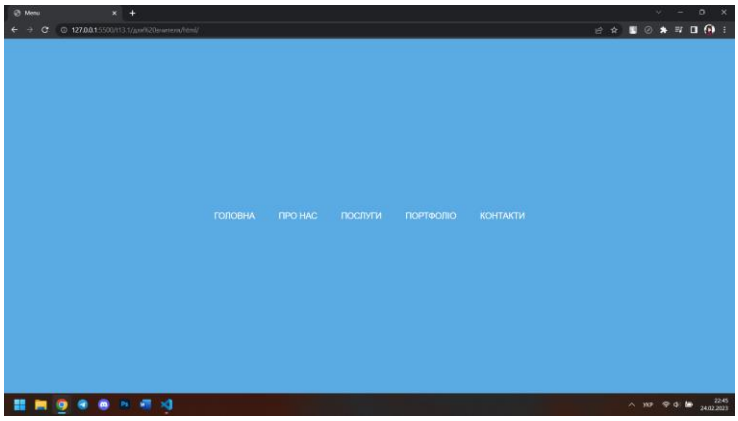
Ненумерований список:
дисплей - флекс

Елемент списку:
маркування списку - відсутнє
зовнішні відступи - 0 20px

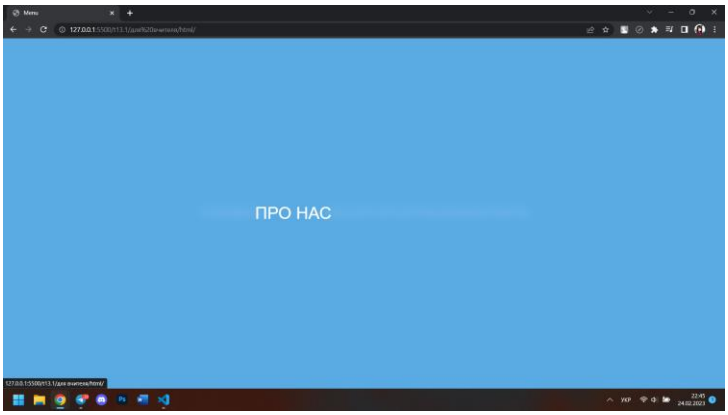
Тег посилання:
дисплей - блоку
позиція - відносна
декорація тексту - відсутня
внутрішні відступи - 5px
шрифт - sans-serif, 18px, білий
трансформація тексту - верхній регістр
перехід - 0,5 секунди

Стилі для тегу посилання при наведенні на ненумерований список (ul:hover a):
трансформація - масштаб(1.5)
непрозорість - 0.2
фільтр - розмиття(5px) (filter: blur(5px))

Стилі для елементів списку при наведенні на посилання (li a:hover):
трансформація - масштаб(2)
непрозорість - 1
фільтр - розмиття(0)



При наведенні на пункт меню:



Завдання 2.

HTML: Створити 6 html файлів - index та 6 сторінок. Підключити шрифт Inter та FontAwesome.

Головна сторінка:

Створити меню за допомогою ненумерованого списку із класом *menu__items*. Також створити секцію *main*, всередині її *container*, всередині контейнеру - блок з *id="content"*

В кожному тегу посилання необхідно додати атрибут *data-item* - це забезпечить правильну роботу ефектів при наведенні на кнопку. В лапки даного атрибуду необхідно вписати той самий текст, що і в тегу посилання. Також необхідно додати атрибут *"target"*, де ми вказуємо *id* блоку, в якому буде відображатись фрейм.

Головна

В блоці контенту додати фрейм:

<iframe src="page1.html" name="content" frameborder="0" scrolling="no" onload="resizeIframe(this)"></iframe>

Також в секцію head додати скрипт, що буде автоматично розраховувати висоту фрейму залежно від вмісту.

```
<script type="text/javascript">
  function resizeIframe(obj) {
    obj.style.height = 0;
    obj.style.height = obj.contentWindow.document.body.scrollHeight + 'px';
  }
</script>
```

На кожній із сторінок (окрім index) весь контент ми розміщуємо в блоці **"main__inner".**

Сторінка викладачів:

Створити блок cards, в якому буде розміщуватись 4 блоки cards__item.

CSS:

Забрати всі стандартні відступи.

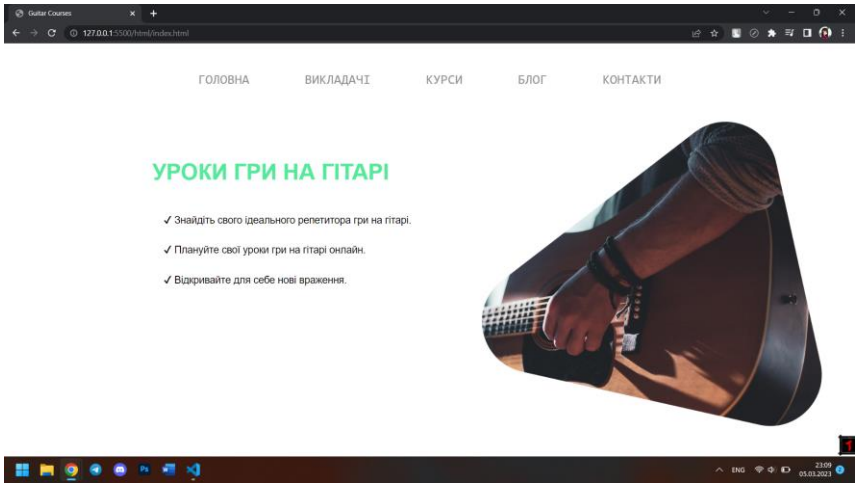
Тіло документа:
шрифт - 18px, 'Inter', sans-serif

container:
максимальна ширина - 1000px
ширина, висота - 100%
вирівнювання по центру за допомогою margin

Тег параграфу:
висота текстової лінії - 25px

Тег посилання:
текстова декорація - відсутня
колір тексту - чорний

(Навігація)



menu__items:
маркування списку - відсутнє
дисплей - флекс
напрямок флекс контенту - рядок
вирівнювання елементів всередині рядків - центр
розподіл контенту по блоку - центр
шрифт - Hack, monospace

Тег елементу списку всередині блоку menu__items:
зовнішні відступи - 50px

Тег посилання всередині тегів елементів списку всередині блоку menu__items (menu__items li a):

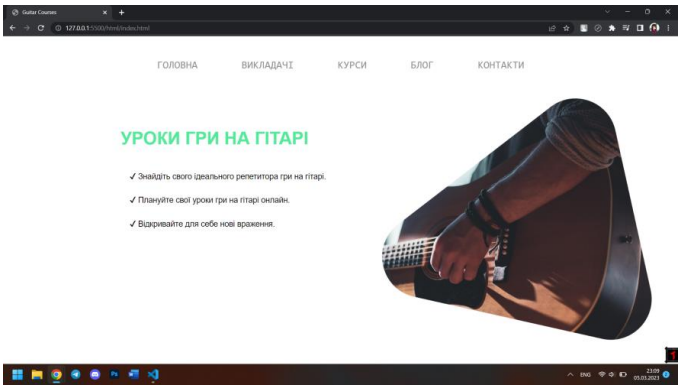
шрифт - 24px, #8f8f8f, вага - 400
перехід - 0.5 секунди
позиція - відносна
трансформація тексту - верхній регістр

Псевдоелемент before для тегу посилання(запис за таким самим принципом, як в попередньому рядку):
контент - attr(data-item)
перехід - 0.5 секунди
колір шрифта - #55e99b
позиція - абсолютна
top - 0
bottom - 0
left - 0
right - 0
ширина - 0
надлишок - прихований

Псевдоелемент before при наведенні на тег посилання:
ширина - 100%
перехід - 0.5 секунди

Стилі тегу фреймів:
ширина - 100%
кордони - відсутні

(Секція main)



main:
фон - зображення background.png, розміщений справа, без повторень
розмір фону - вміщений
висота - 82vh

main__inner:

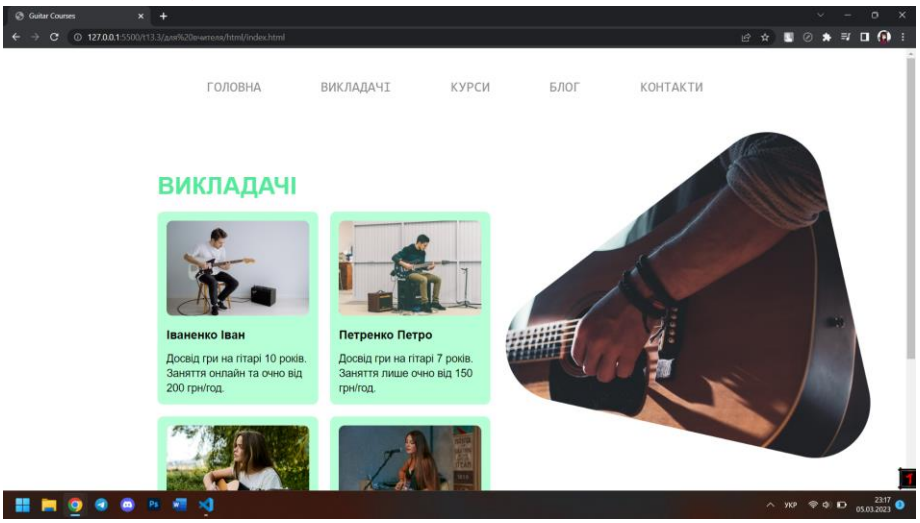
ширина - 60%
дисплей - флекс
напрямок флекс контенту - стовпець
розподіл контенту по блоку - флекс-початок
внутрішній відступ зверху та знизу - 80px

Теги елементів списку всередині нумерованого списку:
зовнішні відступи - 30px 20px
маркування списку - відсутнє

Псевдоелемент before для тегів елементів списку всередині нумерованого списку(ol li:before):
контент - "✓ "

Тег заголовку першого рівня ("Уроки гри на гітарі" і т.п.):
трансформація тексту - верхній регістр
шрифт - 40px, #55e99b
зовнішній відступ знизу - 20px

(Сторінка викладачів)



cards:
дисплей - флекс
напрямок флекс контенту - рядок
перенесення флекс елементів - так

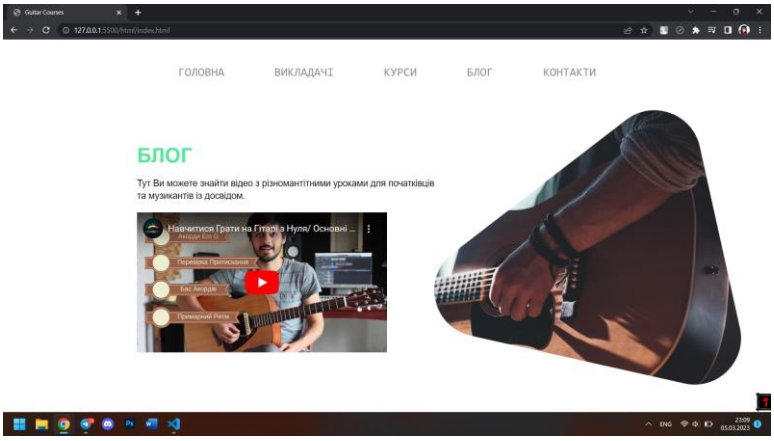
cards__item:
колір фону - #b7ffd7
ширина - 40%
зовнішні відступи - 0 20px 20px 0
радіус кордонів - 10px

внутрішні відступи - 15px

зображення всередині блоку cards__item:
ширина - 100%
радіус кордонів - 10px

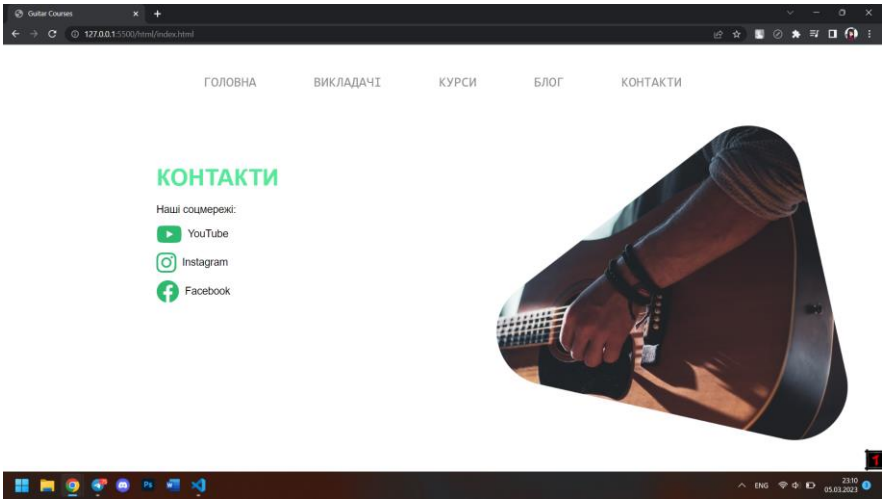
Ім'я викладачів:
шрифт - 20px, вага - 600
зовнішній відступ знизу та зверху - 15px

(Сторінка блогу)



Відео:
висота - 280px
ширина - 500px
зовнішній відступ зверху - 20px

(Сторінка контактів)



Ненумерований список контактів:
зовнішній відступ зверху - 10px

Елементи списку контаків:
зовнішній відступ знизу - 10px

Посилання списку контактів:
дисплей - флекс
вирівнювання елементів всередині рядків - центр

Іконки контактів:
шрифт - 40px, #2fb96f
зовнішній відступ справа - 10px
Фото для вебсайту можете завантажити у classroom.

Звіт до практичного заняття №10

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №11

Тема. Вебформи. Їх властивості та методи.

Мета: ознайомитися з поняттям форми її властивостями та методами, навчитися створювати основні елементи форми.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Теоретичні відомості

Тег **<form>** встановлює форму на вебсторінці. Форма призначена для обміну даними між користувачем і сервером. Область застосування форм не обмежена відправкою даних на сервер, за допомогою клієнтських скриптів можна отримати доступ до будь-якого елементу форми, змінювати його і застосовувати на власний розсуд.

Документ може містити будь-яку кількість форм, але одночасно на сервер може бути відправлена тільки одна форма. З цієї причини дані форм повинні бути незалежні один від одного.

Для відправки форми на сервер використовується кнопка ***Submit***, того ж можна домогтися, якщо натиснути клавішу Enter в межах форми. Якщо кнопка Submit відсутній в формі, клавіша Enter імітує її використання.

Коли форма відправляється на сервер, управління даними передається програмі, заданої **атрибутом action** тега <form>. Попередньо браузер готує інформацію у вигляді пари **«ім'я=значення»**, де ім'я визначається **атрибутом name тега <input>**, а **значення введено користувачем** або встановлено **в поле форми** за умовчанням.

Тег **<input>** є одним з різнобічних елементів форми і дозволяє створювати різні елементи інтерфейсу і забезпечити взаємодію з користувачем. Головним чином <input> призначений для створення текстових полів, різних кнопок, перемикачів і прапорців.

Тег **<label>** встановлює зв'язок між певною міткою, в якості якої зазвичай виступає текст, і елементом форми (<input> , <select> , <textarea>). Такий зв'язок необхідний, щоб змінювати значення елементів форми при натисканні курсором миші на текст. Крім того, за допомогою <label> можна встановлювати гарячі клавіші на клавіатурі і переходити на активний елемент подібно посиланням.

Існує два способи зв'язування об'єкта і мітки. Перший полягає у використанні **ідентифікатора id всередині елемента форми** і надання його імені в якості атрибута **for** тега <label> . При другому способі елемент форми поміщається всередину контейнера <label> .

Тег **<select>** дозволяє створити елемент інтерфейсу у вигляді списку, а також список з одним або множинним вибором, як показано далі. Кінцевий вигляд залежить від використання атрибута size тега <select> , який встановлює висоту списку. Ширина списку визначається самим широким текстом, зазначеним в тезі <option>, а також може змінюватися за допомогою стилів.

Кожен пункт створюється за допомогою тега **<option>**, який повинен бути вкладений в контейнер <select>.

Тег **<option>** визначає окремі пункти списку, що створюється за допомогою контейнера <select>. Ширина списку визначається самим широким текстом, зазначеним в тезі <option>, а також може змінюватися за допомогою стилів.

Атрибут **action** до тега <form> - вказує html сторінку яка завантажується після відправки форми. Цей атрибут не є обов’язковий, але часто використовується.

Атрибути тега <input>:

type - повідомляє браузеру, до якого типу належить елемент форми.

name - ім'я поля, призначене для того, щоб обробник форми міг його ідентифікувати і після отримати з його допомогою значення даного рядка у JS.

placeholder - Виводить текст підказку (*він зникне після введення інформації у форму*).

Min та **max** - встановлює мінімальну та максимальну кількість символів у полі.

required - робить це поле обов’язковим для заповнення

pattern - вказує шаблон для введення інформації у рядок, наприклад для введення українського номера - *pattern="[+]d{3}s[(]d{2}[)]sd{3}[-]d{2}[-]d{2}"*. Якщо присутній атрибут pattern, то форма не буде відправлятися, поки не буде правильно введено відповідну інформацію.

Атрибут **size** до тега **<select>** - встановлює висоту списку, що розкривається. По стандарту дорівнює 1, якщо висота то список буде більше то список буде уже у вигляді множинного вибору.

Атрибути до тега <option>

selected - Заздалегідь встановлює певний пункт списку виділеним.

value - Значення пункту списку, яке буде відправлено на сервер або прочитано за допомогою скриптів.

Подія onclick виникає при натисканні лівою кнопкою миші на елементі, до якого доданий атрибут onclick. Синтаксис onclick="скрипт"

Нові дії JavaScript:

document.getElementById("my-form") - знаходить елемент HTML з ідентифікатором "my-form" на сторінці і повертає посилання на цей елемент.

form.addEventListener - додає обробник події "НазваПодії" до елемента форми. Коли форма надсилається, виконується вказана функція.

event.preventDefault() - зупиняє стандартну дію події "НазваПодії" форми, що відправляє дані на сервер.

form.reset() - очищує всі поля форми, щоб можна було ввести нові дані.

Конструкція switch у JavaScript використовується для реалізації умовної логіки з декількома можливими шляхами виконання коду, в залежності від значення однієї змінної. Основні правила та принципи використання конструкції switch включають наступне:

1. switch використовується для перевірки однієї змінної на декілька можливих значень. Ця змінна називається "контрольованою змінною" або "виразом".

2. В switch використовуються блоки case, які представляють різні варіанти значень для контрольованої змінної.

3. Коли виконується switch, значення контрольованої змінної порівнюється з кожним варіантом case по черзі.

4. Якщо значення контрольованої змінної збігається з варіантом case, виконується код, який пов'язаний з цим варіантом. Виконання продовжується до кінця switc, якщо не зустрічається оператор break.

5. Оператор break використовується для припинення виконання switch, після того як знайдено відповідний варіант. Він забезпечує виход з switch-блоку, щоб код не виконувався для наступних варіантів.

6. Можливо використовувати блок default, який визначає код, який виконується, якщо жоден з варіантів case не збігається зі значенням контрольованої змінної. Блок default є необов'язковим.

Щодо можливості заміни конструкції switch на конструкцію if-else, то в багатьох випадках це можливо. Однак, використання switch може зробити код більш читабельним та зручним, особливо якщо є багато варіантів значень для контрольованої змінної. Конструкцію switch часто використовують, коли треба здійснити вибір серед декількох альтернатив, а не просто перевірити одну умову. За допомогою конструкції if-else можна виконати ті ж самі завдання, але код може стати більш об'ємним та менш зрозумілим, особливо при багатьох умовах.

У загальному, обирання між switch` та if-else залежить від конкретних вимог та особливостей задачі, а також від читабельності та підтримки майбутніх змін у коді.

Завдання для самостійної роботи

Завдання 1

HTML: Підключити шрифт ***Montserrat***. Створити форму, всередині її контейнер. "Реєстрація" - заголовок першого рівня. Текст після нього - в тег параграфу.

Для кожного поля вводу необхідно присвоювати name та id.

Також перед кожним полем вводу треба додати label з атрибутом "for = (id відповідного поля)". Атрибут name буде в подальшому використовуватись в JS.

Усі поля обов'язкові.

Поле вводу імені - тип "текст", мінімальна довжина - 5

Поле вводу пошти - тип "email"

Поле вводу паролю - тип "пароль", паттерн - усі англійські літери, верхнього та нижнього регістру, цифри; довжина - від 4 до 8 ("[a-zA-Z0-9]{4,8}")

Між лейблом та випадним списком необхідно додати тег "br".

Випадний список.

Перше значення, яке одразу показується при відкритті форми ("Код"):
вибраний, прихований

Поле вводу номеру - тип "tel", мінімальна та максимальна довжина значення - 9, паттерн - цифри від 0 до 9; довжина - 9 ("[0-9]{9}").

Для того, щоб стилізувати звичайний **checkbox**, нам необхідно помістити його в label:

```
<label class="checkbox">
   Я погоджуюсь з <a href="#">Умови та конфіденційність</a>.
   <input type="checkbox">
   <span class="checkbox__style"></span>
</label>
```

Також всередині ми додали **span** для того, щоб стилізувати галочку, яка буде з'являтись при натисканні.

Кнопка - тип "submit"

CSS:

Тіло документа:

шрифт - 'Montserrat', sans-serif
колір фону - #f7f7f7
висота - 100vh
зовнішні відступи - 0
дисплей - флекс

розподіл контенту по блоку - центр
вирівнювання елементів всередині рядків - центр

Для усіх елементів сторінки, а також для псевдоелементів before та after задати параметр:
box-sizing: border-box;

Тег посилання:
колір шрифта - dodgerblue

Контейнер:
внутрішні відступи - 20px 30px
максимальна ширина - 500px
ширина - 100%
кордони - 1px, суцільні, #ccc
радіус кордонів - 10px
фон - білий

тінь:
-webkit-box-shadow: 2px 1px 21px -9px rgba(0, 0, 0, 0.38);
-moz-box-shadow: 2px 1px 21px -9px rgba(0, 0, 0, 0.38);
-box-shadow: 2px 1px 21px -9px rgba(0, 0, 0, 0.38);
*(або створити власний варіант тіні за допомогою **онлайн-генераторів**).*

Для усіх полів вводу:
шрифт - 'Montserrat', sans-serif
внутрішні відступи - 15px
зовнішні відступи - 5px 0 22px 0
кордони - відсутні
фон - #f7f7f7
рамка навколо поля вводу - відсутня (outline: none)

Поля вводу типу текст, пароль, пошта:
ширина - 100%

Поле вводу типу пароль, що **пройшло перевірку** на відповідність до паттерну (:valid):
колір фону - #c1f3c1

Поле вводу типу пароль, що **НЕ пройшло перевірку** на відповідність до паттерну (:valid):

колір фону - #eea9a9

Випадний список:
ширина - 20%

Поле вводу типу телефон:
ширина - 79%

Селектор**: focus** для полів вводу типу **текст, телефон, пошта**, а також для **випадного списку**:
колір фону - #ddd

checkbox:
ширина - 100%
позиція - відносна
внутрішній відступ зліва - 30px
курсор - вказівник
перехід - для усього 0.2 секунди
висота лінії тексту - 28px

input всередині класу checkbox (цим кодом ми приховуємо стандартний чекбокс):
позиція - абсолютна
непрозорість - 0
курсор - вказівник
висота - 0
ширина - 0

checkbox__style:
позиція - абсолютна
top - 0
left - 0
висота - 20px
ширина - 20px
колір фону - білий
кордони - 1px, суцільні, #ccc
радіус кордонів - 4px
перехід - для усього 0.2 секунди

Коли натиснуто на input, наш стилізований інпут змінює значення (.checkbox input:checked~.checkbox__style):

колір фону - #07f
колір кордонів - #07f

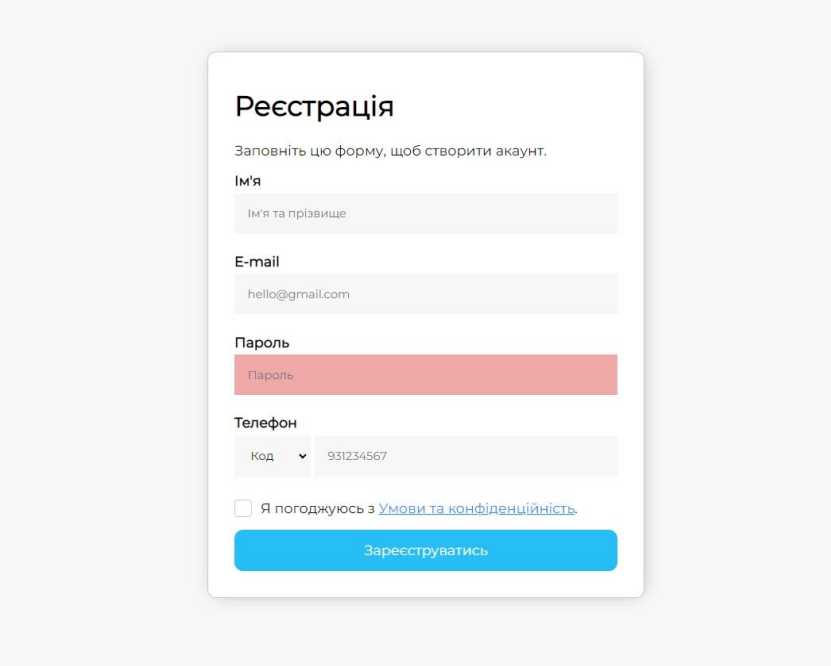
(Стилізована галочка):
Псевдоелемент after для нашого стилізованого чекбоксу (checkbox__style):
контент - ""
позиція - абсолютна
дисплей - відсутній
перехід - для усього 0.2 секунди
left - 6px
top - 1px
ширина - 6px
висота - 12px
кордон - суцільний, білий
ширина кордонів: 0 2px 2px 0
трансформація - обертання на 45 градусів
Коли натиснуто на input, з'являється стилізована галочка (.checkbox input:checked~.checkbox__style:after):
дисплей - блок

Кнопка:
колір фону - #0eb7f4
шрифт - 'Montserrat', sans-serif, 16px, білий
внутрішні відступи - 14px 20px
зовнішні відступи - 10px 0
кордони - відсутні
курсор - вказівник
ширина - 100%
непрозорість - 0.9
радіус кордонів - 10px

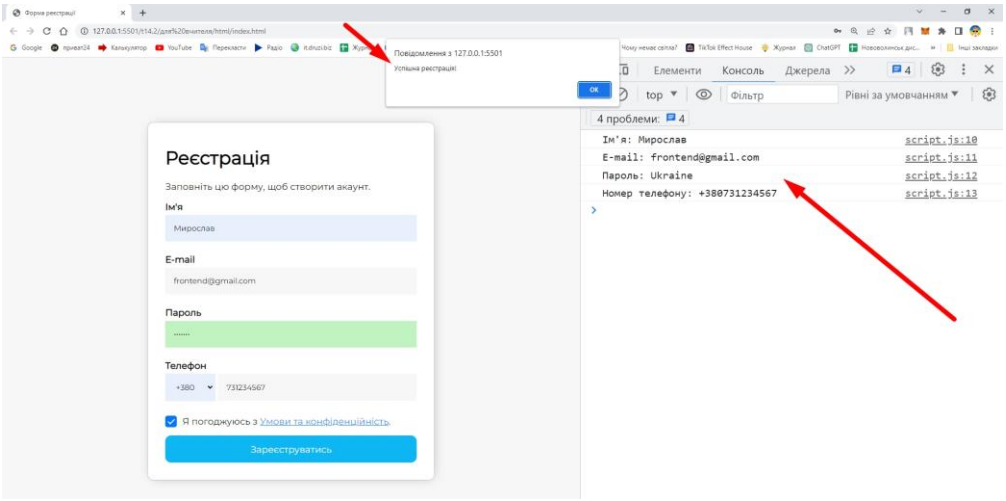
При наведенні на кнопку:
непрозорість - 1

JAVASCRIPT:
1. Вибираємо перший елемент <form> на сторінці.
2. Додаємо слухач подій до форми, який відслідковує подію "submit" (подання форми).
3. Забороняємо формі відправлятися на сервер за замовчуванням.
4. Отримуємо значення, введене користувачем у поле з ідентифікатором 'name'.
5. Отримуємо значення, введене користувачем у поле з ідентифікатором 'email'.

6. Отримуємо значення, введене користувачем у поле з ідентифікатором 'psw' (пароль).
7. Отримуємо значення, введене користувачем у поле з ідентифікатором 'phoneCode' (код країни).
8. Отримуємо значення, введене користувачем у поле з ідентифікатором 'phone' (номер телефону).
9. Виводимо значення змінних 'name', 'email', 'psw', 'phoneCode' та 'phone' у консоль браузера із підписами кожного рядка українською (номер телефону виведіть в один рядок об'єднавши код країни і сам номер).
10. Виводимо спливаюче повідомлення про успішну реєстрацію.
11. Скидаємо значення полів форми до їхніх початкових значень.



Після відправлення форми отримуємо такий результат:



Звіт до практичного заняття №11

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №12

Тема. Семантична верстка HTML5.

Мета: ознайомитися з поняттям семантичної верстки як підхідом до створення вебсторінок, в якому використовуються HTML теги згідно їх призначення та значення.

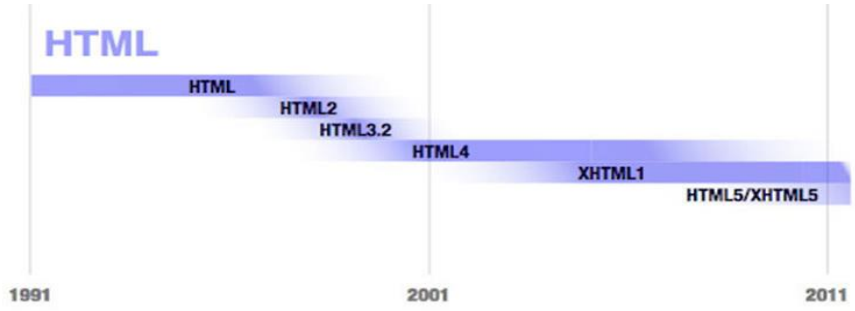
Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Теоретичні відомості

Семантична верстка є підхідом до створення вебсторінок, в якому використовуються HTML теги згідно їх призначення та значення.

Головною метою семантичної верстки є покращення розуміння контенту сторінки не лише для людей, але й для пошукових систем та інших автоматизованих засобів, які аналізують сторінки.



Чому семантична верстка є важливою для розуміння контенту вебсторінки:

1. Покращення доступності: Використання семантичних тегів допомагає людям з обмеженими можливостями, таким як незрячі або люди з вадами зору, використовувати спеціальні технології, щоб зрозуміти контент сторінки. Наприклад, використання правильних тегів дозволяє користувачам з використанням читачів екрану легко навігувати сторінкою та отримувати контекст інформації.

2. Покращення SEO: Пошукові системи, такі як Google, використовують семантичну структуру сторінки для розуміння та індексування контенту. Використання семантичних тегів допомагає зрозуміти, які частини сторінки є заголовками, підрозділами, навігаційними елементами тощо, що позитивно впливає на позиції сторінки у пошукових результатів.

3. Зрозумілість коду: Семантична верстка робить HTML-код більш читабельним та зрозумілим. Коли розробники інших веб-сайтів або ви самі переглядаєте код, семантична структура сторінки допомагає швидше розібратися у її логіці та змісті.

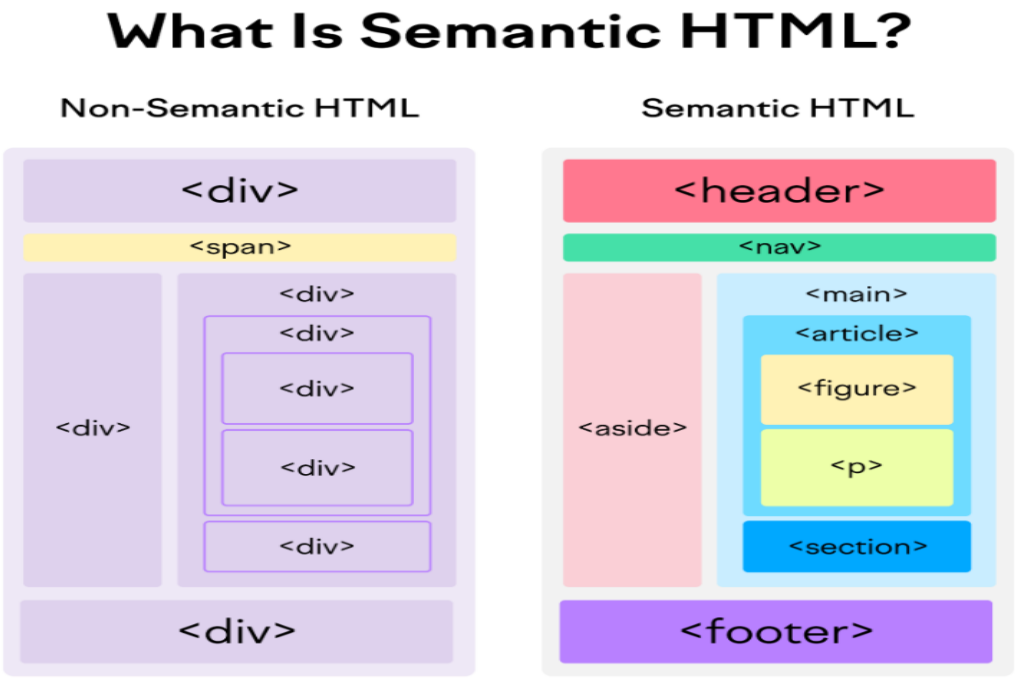
Відмінності між семантичною та несемантичною (структурною) версткою:

1. Семантична верстка: застосовується, коли використовуються HTML теги згідно їх призначення та значення. Теги використовуються для надання смислу та структури контенту, що полегшує його розуміння для людей та пошукових систем. Наприклад, використання <header> для заголовка сторінки, <nav> для навігаційного меню, <article> для самостійного контенту тощо.

2. Несемантична (структурна) верстка: застосовується, коли використовуються загальні блокові елементи, такі як або , без використання спеціальних тегів, що надають смисл структурі. Такий підхід може робити код менш зрозумілим для розробників та інших автоматизованих засобів, що аналізують сторінки.

Основні семантичні теги:

- **<header>**: **цей тег використовується для визначення заголовка або верхньої частини сторінки**. Він може містити логотип, назву сайту або інші елементи, які ідентифікують заголовок сторінки.
- **<nav>**: **цей тег використовується для визначення блоку з навігаційними елементами**, такими як посилання на різні сторінки сайту або розділи сайту. Цей блок надає користувачам зручний спосіб переміщення по веб-сторінці або сайту в цілому.
- **<main>**: **цей тег визначає основний контент сторінки**, який зазвичай відображається безпосередньо на екрані. Він може містити текст, зображення, відео та інші елементи, які є ключовими для сторінки.
- **<article>**: ц**ей тег використовується для визначення незалежного блоку контенту**, такого як окремий блоговий пост, новина або стаття. Його вміст повинен мати сенс сам по собі і може бути розповсюджуваним окремо від іншого контенту сторінки.
- **<section>**: ц**ей тег визначає секцію або блок контенту на сторінці**. Він допомагає організувати і групувати схожі елементи, які мають спільний контекст або тему.
- **<aside>**: **цей тег використовується для визначення додаткового контенту, який зазвичай знаходиться поза основним контентом сторінки**. Він може містити додаткову інформацію, супутні посилання, рекламу або віджети.
- **<footer>**: ц**ей тег визначає підвал сторінки або секції.** Він може містити авторські права, контактну інформацію, посилання на політику конфіденційності або інші елементи, які зазвичай знаходяться в нижній частині сторінки.



Ці семантичні теги допомагають створювати структурований та зрозумілий контент на сторінках, полегшуючи їх розуміння іншим розробникам та закріплюючи їх доступність для користувачів і пошукових систем.

Наприклад, використання тегу <header> для визначення заголовка сторінки дозволяє легко знайти цю частину і зрозуміти, що саме вона представляє. Так само, використання тегу <nav> для навігаційного меню допомагає інтерпретувати цю область як набір посилань для навігації по сайту.

Ознайомлення з рештою семантичних тегів, таких як <article>, <section> і <aside>, дозволить зрозуміти, як можна використовуватися для структурування та організації різних частин контенту на сторінці. Наприклад, елемент <article> може використовуватися для самостійних блоків новин, блогових постів або інших видів контенту, які можуть функціонувати незалежно від іншого контенту на сторінці.

Важливо підкреслити, що вони не лише надають стилістичного оформлення, але й мають семантичне значення, яке визначає структуру і смисл контенту.

Завдання для самостійної роботи

Завдання 1.

HTML:

Усі блоки мають бути обгорнуті в **div *'wrapper'***

header - містить заголовок першого рівня

Div *'container'* містить:

aside - заголовок другого рівня, параграф, навігація

2 однакових блока section містять:

2 article - заголовок другого рівня, параграф, кнопка

Після блоку **container** знаходить **footer** з параграфом.

CSS:

Основні стилі блоків:

body:
висота лінії - 1.6
зовнішні та внутрішні відступи - 0
висота - 100%

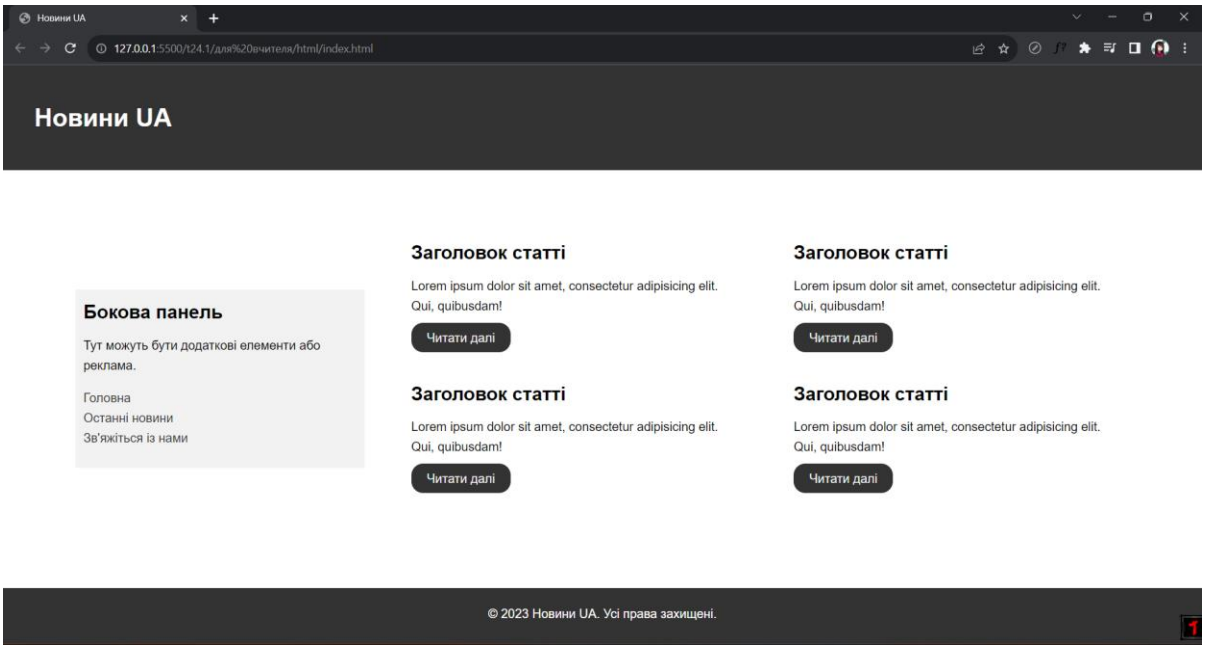
html:
висота - 100%

wrapper:
мінімальна висота - 100%
дисплей - флекс
напрямок флекс елементів - стовпець

container:
дисплей - флекс
напрямок флекс елементів - рядок
розподіл флекс-контенту - простір навколо
вирівнювання елементів всередині рядків - центр
вирівнювання по центру за допомогою margin
ширина - 100%
максимальна ширина - 1400px
флекс - 1 1 авто

aside:
ширина - 25%

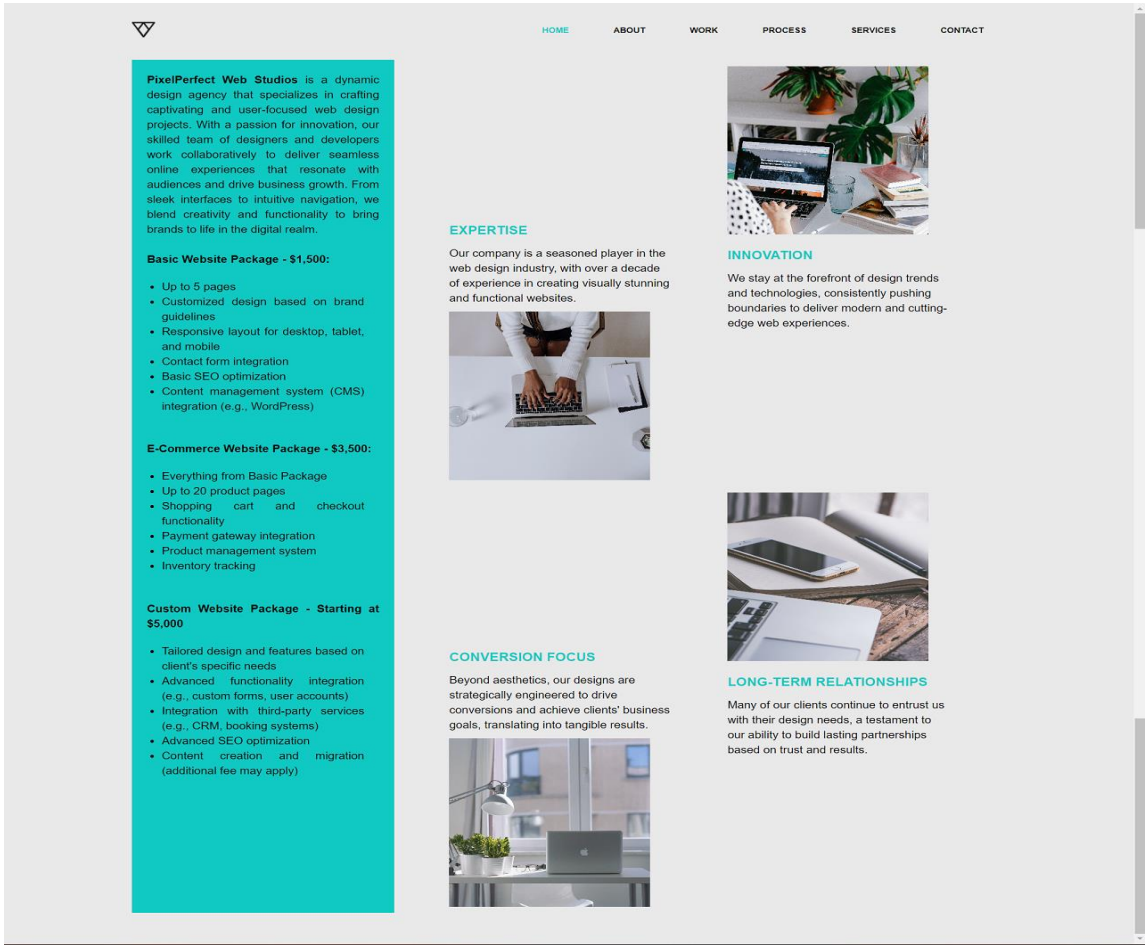
section:
ширина - 30%

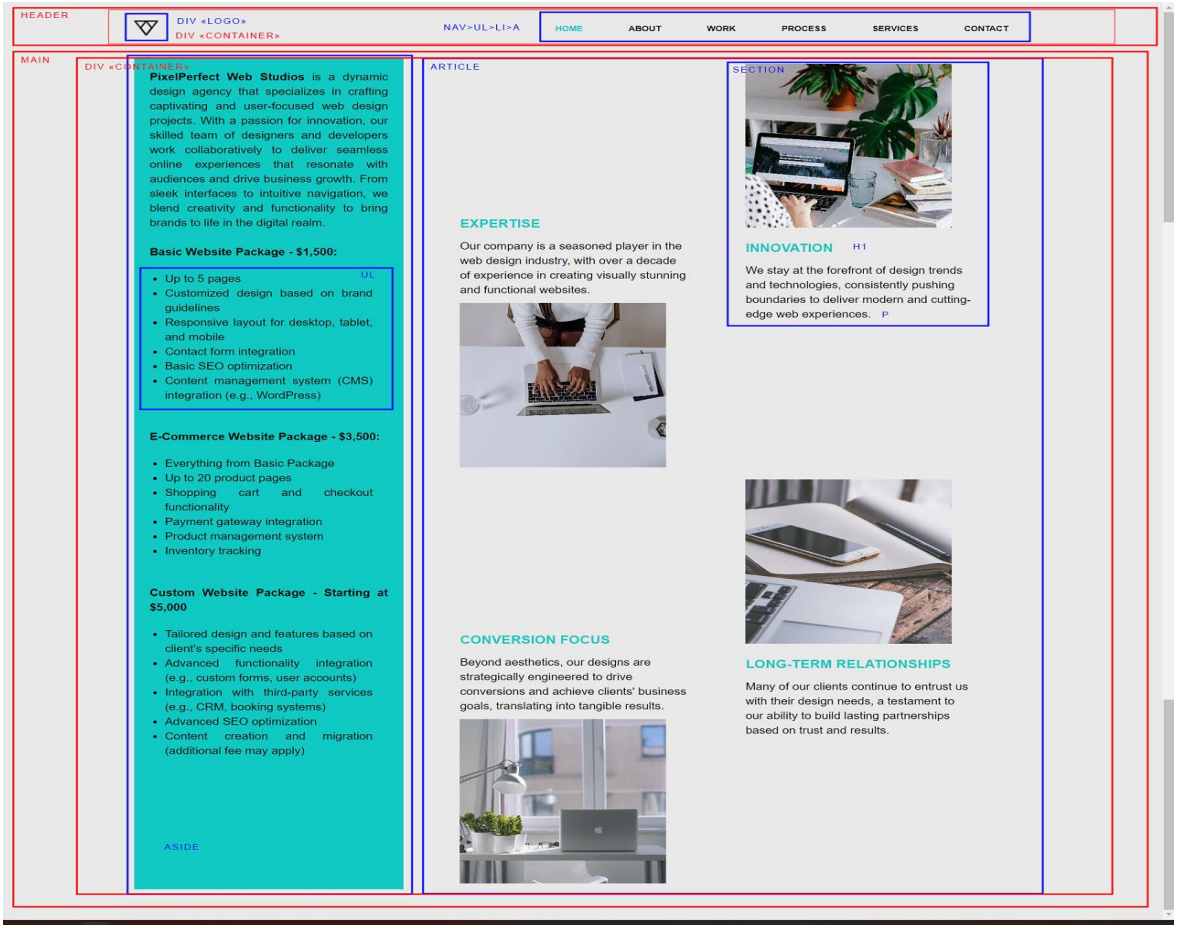


Завдання 2. За даним макетом на фото потрібно відтворити вебсторінку.

Усі стилі у файлі style.css уже прописані, тому важливо використати правильні назви тегів та класів.

Початковий макет та текст вебсторінки можите завантажити у **classroom**.





Звіт до практичного заняття №12

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №13

Тема. CSS-анімації. Анімація елементів плагінами Animate.css і Wow.js

Мета: навчитися створювати анімації на вебсторінці різними засобами.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Теоретичні відомості

CSS-анімації - це потужний інструмент, який дозволяє створювати рухливі та привабливі вебелементи. Вони додають життя до вебсторінок, залучають увагу користувачів та поліпшують їх взаємодію зі сторінкою.

Для створення CSS-анімацій використовується модуль **@keyframes**, який визначає кадри анімації. Це вказує, як елемент буде змінюватися протягом певного проміжку часу. Можна налаштовувати різні властивості, такі як положення, розмір, прозорість, кольори тощо, на різних етапах анімації.

Основні кроки для створення CSS-анімації:

1. **Визначення анімаційного ключового кадру (*@keyframes*):**

Використовуйте ключове слово @keyframes, за яким слідує ім'я анімації та оголошення кадрів анімації з властивостями CSS.

Наприклад:

```
@keyframes animationName {
0% {
   /* Початкові стилі елемента */
}
50% {
   /* Середні стилі елемента */
}
100% {
   /* Кінцеві стилі елемента */
}
}
```

2. **Призначення анімації до елемента**:

Виберіть елемент, до якого бажаєте застосувати анімацію, і використовуйте властивість `animation-name`, щоб вказати ім'я анімації.

Наприклад:

```
.element {
animation-name: animationName;
}
```

3. Налаштування тривалості та інших параметрів анімації:

Використовуйте додаткові властивості, щоб визначити тривалість, тип перехідного ефекту, затримку тощо. Ось кілька прикладів:

```
.element {
animation-duration: 2s; /* Тривалість анімації */
animation-timing-function: ease-in-out; /* Тип перехідного ефекту */
animation-delay: 1s; /* Затримка перед початком анімації */
animation-iteration-count: infinite; /* Кількість повторень (безкінечність) */
animation-direction: alternate; /* Напрям зміни анімації */
}
```

Додатково, CSS пропонує різні функції, які можна використовувати в анімаціях, такі як transform, opacity, color, scale, rotate тощо. Вони дозволяють змінювати властивості елементів під час анімації.

Наприклад:

```
@keyframes animationName {
0% {
transform: scale(1);
}
50% {
transform: scale(1.5);
```

```
}
100% {
transform: scale(1);
}
}
```

Більшість властивостей CSS-анімацій можна об'єднати в одну властивість animation, яка містить різні значення, розділені пробілами. Це зручно, оскільки зменшує кількість рядків коду та полегшує керування анімаціями.

Властивість animation має наступний синтаксис:

animation: name duration timing-function delay iteration-count direction fill-mode;

Ось **опис** кожної частини властивості animation:

- name: ім'я анімації, визначене з використанням `@keyframes`.
- duration: тривалість анімації, вказана в секундах (s) або мілісекундах (ms).
- timing-function: тип перехідного ефекту між кадрами анімації.
- delay: затримка перед початком анімації, вказана в секундах (s) або мілісекундах (ms).
- iteration-count: кількість повторень анімації або значення infinite для безкінечного повторення.
- direction: напрям зміни анімації.
- fill-mode: стан елемента до початку анімації та після її завершення.

Наприклад, ось як виглядає використання властивості `animation` з об'єднаними значеннями:

```
.element {
animation: animationName 2s ease-in-out 1s infinite alternate;
}
```

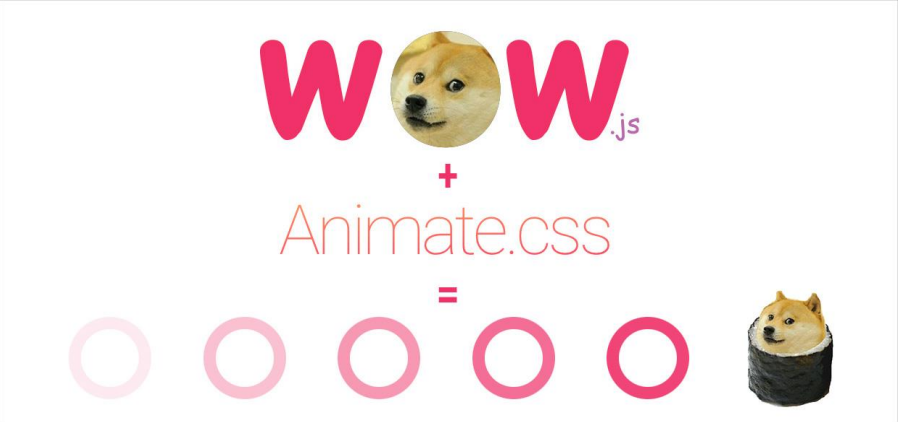
У цьому прикладі, елемент з класом .element отримує анімацію з ім'ям animationName, тривалістю 2 секунди, типом перехідного ефекту ease-in-out, затримкою 1 секунда, безкінечним повторенням, та зміною напряму кожного циклу.

Використання властивості animation дозволяє стиснути код та зробити його більш читабельним, особливо якщо вам потрібно застосувати багато різних властивостей для анімації. Пам'ятайте, що властивість animation також може бути комбінована з окремими властивостями, якщо вам потрібно налаштувати окремі параметри більш детально.

Не соромтесь експериментувати з різними значеннями властивостей анімації, щоб створити унікальні та захоплюючі ефекти на вебсторінках.

Animate.css і **wow.js** - це два популярних плагіни для створення анімації на вебсайті з використанням CSS. **Animate.css** надає бібліотеку з набором

готових анімаційних ефектів, тоді як **wow.js** дозволяє активувати ці анімації при прокрутці сторінки.



Щоб підключити ці плагіни до свого вебсайту, спочатку потрібно завантажити їх з офіційних репозиторіїв або використовувати CDN-посилання. Ось детальна інструкція з підключення кожного з цих плагінів:

1. Підключення **Animate.css**:

- Завантажте Animate.css
- Скопіюйте файл animate.min.css у ваш проєкт.
- Додайте посилання на CSS-файл у вашому HTML-документі:*<link rel="stylesheet" href="шлях_до/animate.css">*

1. Підключення **wow.js**:

- Завантажте wow.js
- Скопіюйте файли wow.min.js у ваш проєкт.
- Додайте посилання на JavaScript-файл у вашому HTML-документі:

<script src="шлях_до/wow.min.js"></script>, а після додайте ще скріпт для ініціалізації wow.js у коді:

<script> new WOW().init(); </script>

Це дозволить активувати анімацію wow.js під час прокручування сторінки. Ви можете використовувати різні класи анімації та експериментувати зі швидкістю та затримкою анімації, налаштовуючи параметри в коді.

Тепер, коли плагіни підключені до вашого вебсайту, ви можете почати використовувати їх для створення анімацій.

Ось кілька прикладів використання плагінів:

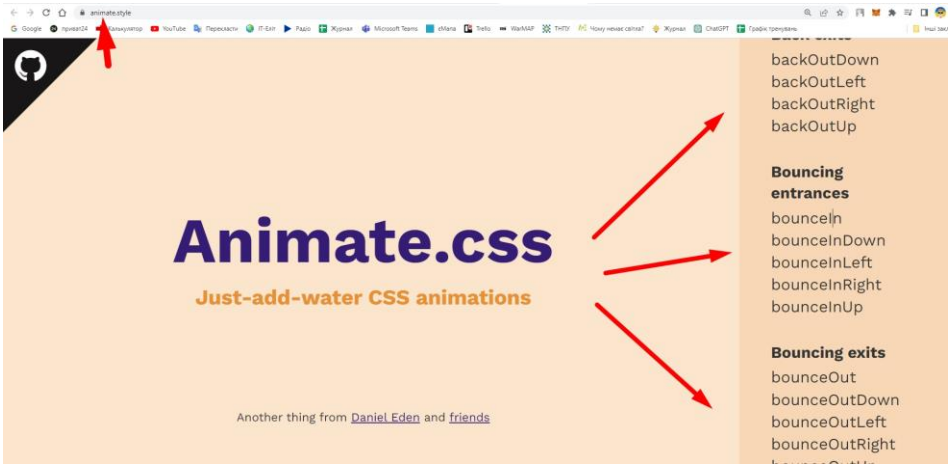
Додавання класів для анімації:

- Додайте **клас "wow"** до елементу, до якого ви хочете застосувати анімацію.
- Додайте **один з класів анімації** до цього елементу. Наприклад, "fadeIn" або "bounce".
- Приклад HTML-коду:

Привіт, світ!

Назви анімацій та як вони працюють можна переглянути на **ГОЛОВНІЙ СТОРІНЦІ Animate.css.** Обираєте анімацію яка вам сподобалася

--> клікаєте --> її назва скопіюється --> і вставляєте її як ще один клас через пробіл після **wow.**



Тепер розглянемо, як можна змінювати параметри анімацій з плагінами Animate.css і wow.js.

1. Animate.css:

Animate.css містить багато класів для різних анімацій. Зазвичай, елементам додаються класи, щоб застосувати анімацію. Але для додаткового контролю над анімаціями, можна змінювати параметри за допомогою додаткових класів або стилів.

Наприклад, додамо анімацію bounceIn для елемента div з затримкою 1 секунда та тривалістю 2 секунди:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="animate.min.css">
    <style>
      .custom-animation {
      animation-delay: 1s;
      animation-duration: 2s;
      }
    </style>
  </head>
<body>
  <div class="animated bounceIn custom-animation">Привіт, світ!</div>
</body>
</html>
```

В цьому прикладі, ми додали клас "bounceIn" для елементу для анімації. Крім того, ми створили стиль з назвою "*custom-animation*" і задали атрибути animation-delay і animation-duration, щоб змінити параметри анімації для цього елемента.

Animate.css також надає можливість використовувати зворотні анімації. Для цього можна додати додатковий клас "animated" і назву анімації з префіксом "out", наприклад, "fadeOut".

2. wow.js:

Wow.js додає анімації при прокрутці сторінки та дозволяє змінювати параметри за допомогою атрибутів `data-wow-*`.

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="animate.min.css">
  </head>
  <body>
    <div class="wow fadeIn" data-wow-delay="1s" data-wow-duration="2s">Привіт, світ!</div>
        <script src="wow.min.js"></script>
    <script>
      new WOW().init();
    </script>
  </body>
</html>
```

У цьому прикладі, ми додали клас "wow" і "fadeIn" для анімації елемента . Крім того, ми використовуємо атрибути data-wow-delay і data-wow-duration, щоб змінити затримку і тривалість анімації для цього елемента.

За допомогою wow.js ви також можете змінювати інші параметри, такі як відстань до елементу, коли анімація буде запущена, використовуючи атрибут data-wow-offset.

Завдання для самостійної роботи

Завдання 1

HTML:

Створити блоки з класами: header, main, footer.

В header помістити заголовок першого рівня, навігацію з ненумерованим списком та порожній блок "slider"

В main помістити три секції, в кожній з яких знаходиться блоки (id відповідно до фігури, клас 'figures') з таким вмістом:

1. порожній блок з класами shape та відповідною фігурою (square, circle, triangle)
2. блок з класом shape-name
3. блок з класом formula
4. блок з класом formula (має бути два)

В footer розмістити блок з класом footer-text.

CSS:

Забрати у всіх елементів внутрішні та зовнішні відступи.

html:

прокручування - повільне

body:
шрифт - Arial, sans-serif

header:
фон - лінійний градієнт вліво з кольорів #ffff00 та #ff0000
колір тексту - білий
внутрішні відступи - 20px

Ненумерований список в навігації:
тип списку - відсутній
внутрішні відступи - 0
зовнішні відступи - 10px 0
дисплей - флекс

Елементи списку в ненумерованому списку в навігації (nav ul li):
зовнішній відступ справа - 20px

nav-link (елементи посилання):
колір тексту - білий
текстова декорація - відсутня
перехід - колір, 0.3 секунди, ease

При наведенні на nav-link:
колір тексту - жовтий

slider:
ширина - 20px
висота - 10px
колір фону - червоний
позиція - відносна
анімація - example 4 секунди, безкінечна

Ключові кадри example:
0%:
  колір фону - #ff0000
  left - 0px
  top - 0px
50%:
  колір фону - #ffff00

left - 300px
top - 0px
100%:
колір фону - #ff0000
left - 0px
top - 0px

.main, .figures, .shape:
дисплей - флекс
напрямок флекс-контенту - стовпець
вирівнювання елементів всередині рядків - центр

main:
розподіл контенту по флекс-блоку - простір навколо
зовнішній відступ зверху - 50px

Секції:
зовнішній відступ знизу - 100px

shape:
зовнішній відступ знизу - 30px

shape-name:
ромір шрифта - 24px
шрифт - жирний
зовнішній відступ знизу - 10px

formula:
розмір шрифта - 16px
зовнішній відступ знизу - 5px

Квадрат:
ширина - 100px
висота - 100px
колір фону - червоний
анімація - rotate-color-square 4 секунди, безкінечна, лінійна

Ключові кадри rotate-color-square:
0%:
трансформація - поворот на 0 градусів
колір фону - червоний

25%:
трансформація - поворот на 90 градусів
колір фону - жовтий
50%:
трансформація - поворот на 180 градусів
колір фону - синій
75%:
трансформація - поворот на 270 градусів
колір фону - зелений
100%:
трансформація - поворот на 360 градусів
колір фону - червоний

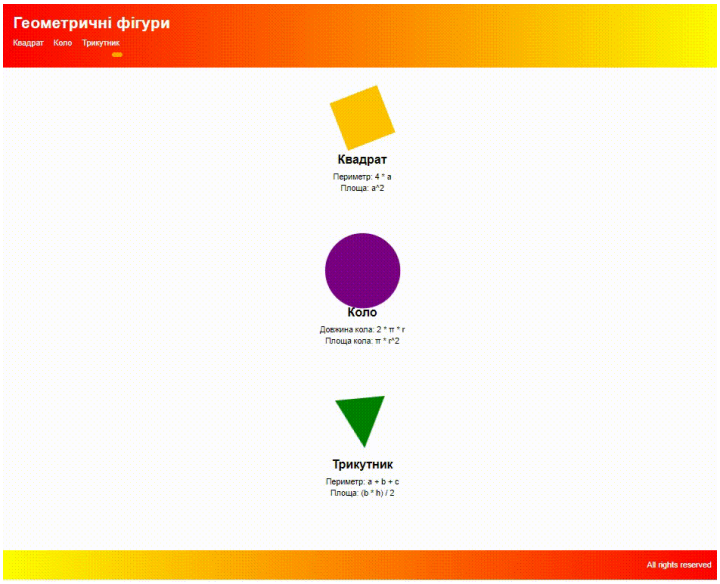
Коло:
ширина - 100px
висота - 100px
колір фону - синій
радіус кордонів - 50%
анімація - scale-color-circle 2 секунди, безкінечна, alternate

Ключові кадри scale-color-circle:
0%:
трансформація - масштаб(1)
колір фону - синій
50%:
трансформація - масштаб(1.5)
колір фону - фіолетовий
100%:
трансформація - масштаб(1)
колір фону - синій

Трикутник:
ширина - 0
висота - 0
кордони зліва - 50px, суцільні, прозорі
кордони справа - 50px, суцільні, прозорі
кордони знизу - 100px, суцільні, зелені
анімація - flip-color-triangle 2 секунди, безкінечна
зовнішній відступ знизу - 20px

Ключові кадри flip-color-triangle:
0%:
трансформація - поворот на 0 градусів
50%:
трансформація - поворот на 180 градусів
100%:
трансформація - поворот на 360 градусів

footer-text:
горизонтальне вирівнювання тексту - право
внутрішні відступи - 20px
фон - лінійний градієнт вліво з кольорів #ffff00 та #ff0000
колір тексту - білий



Завдання 2

Підключити FontAwesome. Підключити Animate.css і **wow.js** *(див. теорію до теми)*. Змінити прізвище ім'я та фото на ваше власне, також дописати на сторінці усі пусті тексти *(де зараз вставлений текст-риба Lorem)* на ваш власний, відповідно до теми сторінки *(висловити свою думку)*. Встановіть покликання на ваші особисті соцмережі до відповідних кнопок, за потреби змініть іконки.

Встановіть наступні налаштування **анімацій**, їх **затримки** та **тривалості** для наступних блоків:

header 's-header sticky':
1. div 'header-logo' > анімація bounceInLeft
2. nav 'header-nav-wrap' > ul 'header-main-nav' > в кожному li > **анімація slideInDown**
3. nav 'header-socials' > **анімація bounceInRight**

section id='intro', class='s-intro full-width':
1. div 'row intro-content' >
   div 'column intro-text' > **анімація fadeInLeft ; затрика анімації 1s**
   div 'intro-pic' > **анімація slideInRight" ; затрика анімації 1.5s**

section id='about', class='s-about';
усі елементи обгорнуті в блок div 'about-me':
1. div 'row' > div 'about-me__text' > p 'lead' > до перших двох додати **анімацію fadeInLeft** до інших двох **fadeInRight**
2. div 'row about-me__buttons' >
   div 'half-width' - a 'btn btn--stroke full-width' > **анімація fadeInUp ; затрика анімації 500ms**
   div 'half-width' - a 'btn btn--primary full-width' > **анімація fadeInUp ; затрика анімації 800ms**

Усі наступні елементи обгорнуті в блок div 'about-experience', досі знаходяться всередині 'about-me':
1. div 'row heading-block' > div 'column' > **анімація bounceIn**
2. div 'row about-experience__timeline' > 2 div 'column'
   в блоках 'column' > div 'timeline'>
   div 'timeline__icon-wrap' *(таких є 2, один тут а другий нижче шукайте)* > **анімація bounceIn**
   Три перших однакових блоки div 'timeline__block':
      div 'timeline__bullet'
      div 'timeline__header' > **анімація bounceInLeft ; затримка анімації 400ms**
      div 'timeline__desc' > **анімація flipInX ; затримка анімації 1000ms**
   Три других однакових блоки div 'timeline__block':
      div 'timeline__bullet'
      div 'timeline__header' > анімація bounceInRight ; **затримка анімації 400ms**
      div 'timeline__desc' > **анімація flipInX ; затримка анімації 1000ms**

section id='services' class='s-services ss-dark':
1. div 'shadow-overlay'
2. div 'row heading-block heading-block--center' > div 'column' > h2 'section-heading section-heading--centerbottom' > **анімація tada**, а нижче до p 'section-desc' > **анімація pulse ; затримка анімації 0.5s**
3. div 'row services-list block-large-1-3 block-medium-1-2' > в ньому розмістити 6 однакових блоків:
дальше йде три блоки div 'column item-service'
> до першого **анімація slideInLeft ; затримка анімації 1s**
> до другого **анімація slideInDown ; затримка анімації 1.5s**
> до третього **анімація slideInRight ; затримка анімації 2s**

> до четвертого **анімація slideInLeft ; затримка анімації 2.5s**
> до п'ятого **анімація slideInDown ; затримка анімації 3s**
> до шостого **анімація slideInRight ; затримка анімації 3.5s**

section id='works' class='s-works':
1. div 'row heading-block heading-block--center' > div 'column' > h2 'section-heading section-heading--centerbottom' > **анімація flip; тривалість анімації 2.5s;**
нижче p 'section-desc'> **анімація flash; затримка анімації 2s; час анімації 5s;**
2. div 'masonry-wrap' > div 'masonry' > div 'grid-sizer', 6 однакових блоків:
div 'masonry__brick' >**анімація** fadeIn **; затримка анімації 1s;**

До наступтих блоків 'masonry__brick' >**анімації flipInY та flipInX тв затримки анімації 2s та 1s до усіх по черзі блоків міняються**

section id='contact' class='s-contact ss-dark':
1. div 'row heading-block' > div 'column large-full' > **самі оберіть анімацію та її тривалість чи затримку до цього елемента.**
2. div 'row contact-main' > div 'column large-full' > **самі оберіть анімацію та її тривалість чи затримку до цього елемента.**
p 'section-desc' > **самі оберіть анімацію та її тривалість чи затримку до цього елемента.**
3. div 'row contact-infos' >
div 'column large-5 medium-full contact-phone' > **самі оберіть анімацію та її тривалість чи затримку до цього елемента.**
div 'column large-7 medium-full contact-social' >**самі оберіть анімацію та її тривалість чи затримку до цього елемента.**

footer:
1. div 'row' >
div 'column large-full ss-copyright' > **самі оберіть анімацію та її тривалість чи затримку до цього елемента.**

Макет та відео для вебсторінки можете завантажити у classroom.

Звіт до практичного заняття №13

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №14

Тема. Підсумкова робота 3.

Мета: створити власний сайт з використанням мови опису HTML5 та CSS, технологія Flex, а також добавити анімації.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3, Visual Studio Code.

Завдання для самостійної роботи

1. Визначіться з темою та концепцією сайту, який Ви бажаєте створити.

2. Заповніть файл технічного завдання і затвердіть його у викладача.

3. Зразок заповнення технічого завдання ви можете переглянути у classroom.

4. Лише після затвердження можете переходити до розробки сайту.

ТЕХНІЧНЕ ЗАВДАННЯ
на розробку сайту «НАЗВА САЙТУ»

Розробник : «Ім'я Прізвище»

1. ОСНОВНА ІНФОРМАЦІЯ:

Описати на яку тему буде Ваш сайт, яка ціль його створення і яку функцію буде виконувати розроблений Вами сайт.

2. ЕТАПИ І ТЕРМІНИ СТВОРЕННЯ САЙТУ:

Опрацювання технічного завдання. Розробка концепції сайту.

прописати зовнішній вигляд сайту – або у вигляді таблиці, або намалювати в графічному редакторі, зберегти і вставити зображення

Інформаційне проектування:

прописати вміст сайту – більш детально описати, що саме буде розміщено на сторінках сайту

- Оформлення технічного завдання (*повністю заповнити цей файл*).
- Затвердження у викладача (*викладач повинен затвердити дане ТЗ, Вашу ідею сайту, тільки після цього можна приступати до його створення*).

Загальний термін робіт по створенню сайту – 4 заняття (з доопрацюванням сайту вдома).

- Розробка ескізу базового дизайну *(написати порожній макет сайту і розставити блоки згідно продуманого дизайну)*.
- Інформаційне проектування *(продумати вміст сайту, підібрати тексти та зображення)*.
- Створення працюючого шаблону сайту: повне його наповнення – сторінки, тексти, посилання, зображення тощо.
- Публікація сайту на сервері: *вставити адресу сайту*
- Підготовка презентації сайту – доповідь та демонстрація (розраховувати на презентацію у 5-10 хвилин).
- Презентація сайту перед групою (*наступний тиждень після затвердження цього технічного завдання*).

3. СТРУКТУРА І ОПИС САЙТУ:

1. Сторінки сайту:

Прописати назви всіх сторінок Вашого сайту

2. Меню сайту:

Прописати всі пункти меню Вашого сайту та його поведінку (в якому вікні буде відкриватися пункт, відображення при наведенні мишею, відображення поточного пункту, відображення пунктів, які вже були переглянуті)

3. Сторінка «_____»:

Прописати відображення сторінки, її вміст, посилання (крім меню), зображення тощо.

4. Сторінка «_____»:

Прописати відображення сторінки, її вміст, посилання (крім меню), зображення тощо.

5. Сторінка «_____»:

Прописати відображення сторінки, її вміст, посилання (крім меню), зображення тощо.

6. Сторінка «_____»:

Прописати подібним чином усі сторінки.

*весь червоний текст потрібно замінити на Ваші відповіді.

УВАГА!

Ваш сайт в будь-якому випадку повинен містити:

- не менше 5 сторінок
- на кожній сторінці повинне бути, як мінімум, одне зображення або відео чи аудіофайл
- мінімум 1 зображення має бути розміщеним на фоні блоку чи сторінки
- мінімум 1 список маркований (можна змінити значки) та 1 список нумерований
- мінімум 1 таблиця
- мінімум 1 форма на сайті (наприклад, реєстрації, зворотного зв'язку, замовлення тощо), і дані введені користувачем мають виводитися у консоль!
- кнопки (наприклад, пункти меню – це кнопки, або кнопки на сторінках для переходу на форму (внутрішні покликання), та декілька зовнішніх покликань тощо),
- стандарт розмітки – html5, тобто використовувати семантичну верстку
- мінімум 1 css-анімація
- декілька параметрів :hover для будь-яких елементів (наприклад, кнопки, покликання, зображення тощо).
- мінімум 1 розділ має бути стилізований flex блоками.
- використати іконки FontAwesome
- застосувати мінімум до 2-3 елементів анімацію Animate.css і wow.js

Звіт до практичного заняття №14

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №15

Тема. Основні положення мови сценаріїв Java Script

Мета: навчити працювати із змінними введення виведення даних, сформувати поняття про основні оператори мови Java Script.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Теоретичні відомості

Створення змінних

Створити змінну в програмі можна декількома способами:

• За допомогою оператора присвоєння значень у форматі:

ім'я_змінної = значення

Оператор присвоєння позначається символом рівності **“=“.**

Приклад:

myName = «Іван»

• За допомогою ключового слова **var** (від variable — змінна) у форматі:

var ім'я_змінної

У цьому випадку створена змінна не має ніякого значення, але може його одержати надалі за допомогою оператора присвоєння.

Приклад:

var myName

myName =«Іван»

• За допомогою ключового слова **var** і оператора присвоєння у форматі:

var ім'я_змінної = значення

Приклад

var myName = «Іван»

Оператор	Приклад	Еквівалентний вираз
+=	X+=Y	X=X + Y
-=	X-=Y	X=X-Y
*=	X*=Y	X=X*Y
/=	X/=Y	X=X/Y
%=	X%=Y	X=X%Y

арифметичні оператори	Назва	Приклад
+	Додавання	X + Y
-	Віднімання	X-Y
*	Множення	X*Y
/	Ділення	X/Y
%	Ділення по модулю	X%Y

++	Збільшення на 1	X++
--	Зменшення на 1	Y--

Оператор	**Назви**	**Приклад**
==	Дорівнює	X==Y
!=	Не дорівнює	X!=Y
>	Більше, ніж	X>У
>=	Більше або дорівнює (ме менше)	X>=Y
<	Менше, ніж	X<Y
<=	Менше або дорівнює (не більше)	X<=Y

Оператор	**Назва**	**Приклад**
!	Заперечення (НЕ)	!X
&&	І	X&&Y
||	АБО	X||Y

X	**Y**	**X&&Y**	**X||Y**
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Завдання для самостійної роботи

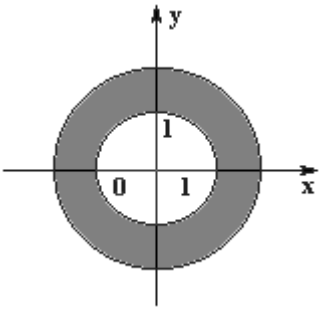
1. Напишіть вираз істинний тоді і тільки тоді, коли:

- значення цілої змінної M ділиться на ціло на значення цілої змінної K;
- значення дійсних змінних A, B та C утворюють неспадаючу послідовність;
- значення змінної X є найбільшим з трьох попарно різних значень X, Y, Z;
- жодна з логічних змінних A, B, C не істинна;

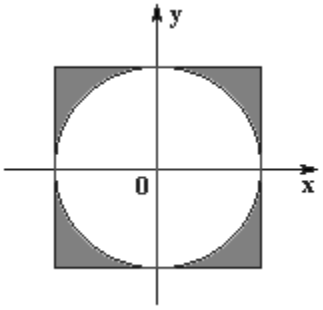
2.Накресліть та заштрихуйте область, таку, що заданий вираз., в якому значення x та y трактуються як координати точки на площині, приймає значення true:

- (X*Y>0);
- Y+X<5 && X*X+Y*Y>i;
- Y<X*X+2 || Y>6.

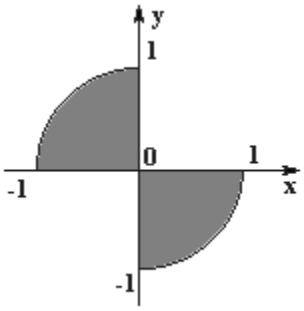
3. Напишіть формулу, істинну тоді і тільки тоді, коли точка на площині з координатами X и Y попадає в заштриховану область (Мал. 6. - Мал. 9.).



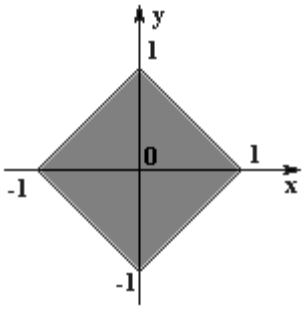
Мал. 6. Точка попадає в область, утворену виключенням двох кругів



Мал. 7. Точка попадає в область, утворену виключенням квадрату та кругу



Мал. 8. Точка попадає в область, утворену двома секторами



Мал. 9. Точка попадає в область, утворену ромбом

4.Оформити звіт.

Звіт до практичного заняття №15

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Охарактеризуйте мову програмування JavaScript.

- Які Ви знаєте команди для введення даних?
- Які Ви знаєте команди для виведення даних?
- Як записуються імена зміних?
- Охарактеризуйте типи даних.
- Для чого призначені оператори у програмі?
- Як записутються коментарі у програмі?
- Охарактеризуйте арифметичні оператори та оператори порівняння.
- Охарактеризуйте логічні оператори.

Практичне заняття №16

Тема. Галуженні та циклічні програми мовою сценаріїв Java Script

Мета: сформувати поняття про галуженні та циклічні програми мови Java Script, скласти програму за допомогою сценаріїв Java Script.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Теоретичні відомості

Оператори умовного переходу

Обчислювальний процес можна направити тим чи іншим шляхом в залежності від того, чи виконується деяка умова чи ні. Для цієї мети служать оператори умовного переходу ***if і switch.***

Оператор if

Оператор умовного переходу if дозволяє реалізувати структуру умовного виразу якщо..., то ..., інакше...

Синтаксис оператора if переходу наступний:

```
if (умова)
        { код, котрий виконується в разі виконання умови }
  else
        { код, котрий виконується, якщо умова не виконана }
```

У фігурних дужках розташовується блок коду — кілька виразів. Якщо в блоці використовується не більш одного виразу, то фігурні дужки можна не писати. Частина цієї конструкції, обумовлена ключовим словом else (інакше), необов'язкова. У цьому випадку залишається тільки частина, визначена ключовим словом if (якщо):

```
if (умова)
        { код, що працює, якщо умова виконана }
```

Конструкція оператора умовного переходу допускає вкладення інших операторів умовного переходу. Умова зазвичай являє собою вираз логічного типу.

Оператор switch

Для організації перевірки невеликої кількості умов цілком достатньо використовувати розглянутий вище оператор if. Однак у випадку декількох умов більш зручний і наочним виявляється оператор switch (перемикач). Він

особливо зручний, якщо потрібно перевірити кілька умов, що не є взаємовиключними.

Синтаксис оператора ***switch*** *виглядає в такий спосіб:*

```
switch (вираз) {
   case варіант1:
      код
      [break]
   case варіант2:
      код
      [break]
...
   [default:
код]
}
```

Параметр вираз оператора switch може приймати рядкові, числові і логічні значення. Зрозуміло, що у випадку логічного виразу можливі тільки два варіанти. Ключові слова (оператори) break і default не є обов'язковими, про що свідчать прямокутні дужки. Тут вони є елементами опису синтаксису, і при написанні операторів їхній указувати не потрібно. Якщо оператор break зазначений, то перевірка інших умов не здійснюється. Якщо зазначено оператор default, то наступний за ним код виконується, якщо значення виразу не відповідає жодному з варіантів. Якщо усі варіанти можливих значень передбачені в операторі switch, то оператор default можна не використовувати.

Оператори циклу

Оператор циклу забезпечує багаторазове виконання блоку програмного коду доти, поки не виконається деяка умова, У JavaScript передбачені три, оператора циклу: for, while і do-while.

Оператор for

Оператор for (для) також називають оператором з лічильником циклів, хоча в ньому зовсім не обов'язково використовувати лічильник.

Синтаксис цього оператора наступний:

```
for ( [початковий вираз] ; [умова] ; [вираз оновлення] )
     {
        код
      }
```

Тут квадратні дужки лише вказують на те, що вказані в них параметри не є обов'язковими.

Усе, що знаходиться в круглих дужках праворуч від ключового слова for, називається заголовком оператора циклу, а вміст фігурних дужок — його тілом.

Оператор циклу працює в такий спосіб. Спочатку виконується початковий вираз. Потім перевіряється умова. Якщо вона істинна, то виконується код, записаний у фігурних дужках. Після цього виконується вираз оновлення (третій параметр оператора for). У випадку неістинності умови оператор циклу припиняє роботу (при цьому код не виконується).

Типова структура оператора циклу з використанням ***break*** *має такий вигляд*:

```
for ( [початковий вираз] ; [умова1] ; [вираз оновлення] }
    {
      код
          if (умова2) {
              код
              break
                    }
      код
    }
```

Для керування обчисленнями в операторі циклу можна також використовувати оператор **continue** (продовження). Так само, як і break, цей оператор застосовується в тілі оператора циклу разом з оператором умовного переходу. Однак, на відміну від break, оператор continue припиняє виконання наступного за ним коду, виконує вираз оновлення і повертає обчислювальний процес у початок оператора циклу, де відбувається перевірка умови, зазначеної в заголовку.

Типова структура оператора циклу з використанням ***continue*** *має такий вигляд:*

```
for ( [початковий вираз] ; [умова1] ; [вираз оновлення] )
  {
        код
        if (умова2) {
          код
          continue
          }
        код
   }
```

Оператор while

Оператор циклу while (доти, поки) має структуру більш просту, чим оператор for, і працює трохи інакше.

Синтаксис цього оператора наступний:

```
while ( умова )
    {
      код
     }
```

При виконанні цього оператора спочатку здійснюється перевірка умови, зазначеної в заголовку, тобто в круглих дужках праворуч від ключового слова while. Якщо вона виконується, то виконується код у тілі оператора циклу. У противному випадку код не виконується. При виконанні коду (завершенні першої ітерації) обчислювальний процес повертається до заголовка, де знову перевіряється умова і т.д.

Оператор do-while

Оператор do-while (роби доти, поки) являє собою конструкцію з двох операторів, використовуваних спільно.

Синтаксис цієї конструкції наступний:

```
do {
        код
  }
  while (умова)
```

На відміну від оператора while в операторі do-while код виконується хоча б один раз, незалежно від умови. Умова перевіряється після виконання коду, якщо вона істинна, то знову виконується код у тілі оператора do. У противному випадку робота оператора do-while завершується. В операторі while умова перевірялась, в першу чергу, до виконання коду в тілі. Якщо при першому звертанні до оператора while умова не виконується, то код не буде виконаний ніколи.

Завдання для самостійної роботи

1. Розв'язати задачі:

Задача 1

Нехай $x_1=x_2=x_3=x_4=1$; $x_i=x_{i-1}+x_{i-3}$, i=5,6,… Знайти двадцятий елемент послідовності. Масив не використовувати.

Задача 2

Напишіть програму, яка визначає чи можна побудувати трикутник с заданими довжинами сторін.

Задача 3.

Точка на площині задається своїми координатами. Визначте, якій з чвертей прямокутної системи координат належить задана точка.

Звіт до практичного заняття №16

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Охарактеризуйте оператори умовного переходу.
- Наведіть приклади задач.
- Охарактеризуйте оператори циклу.
- Наведіть приклади задач.

Практичне заняття №17

Тема. Знайомство з технологією AJAX.

Мета: сформувати поняття AJAX – інноваційний підхід до розробки сайту, визначити основні переваги та недоліки.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Теоретичні відомості

Використання AJAX (Asynchronous JavaScript and XML) у роботі сайту уже стало «фішкою», яка свідчить про його інноваційність і відповідність сучасним тенденціям. AJAX є величезним проривом у розвитку Інтернет-технологій, але водночас це етап їх розвитку, який давно назрівав, а тому був неминучим. AJAX – це підхід до веб-розробки, який дає можливість веб-сторінці довантажувати необхідну інформацію без перезавантаження самої сторінки.

AJAX почали використовувати у 2005 році. AJAX не є окремою технологією, а концепцією використання декількох технологій, що існували і раніше. До цих технологій належать javascript, HTML, CSS, PHP, DOM, dHTML. AJAX є одним з елементів концепції DHTML, використовує DHTML для динамічної зміни контенту, а XMLHttpRequest – для динамічних звернень до сервера (без перезавантаження сторінки).

AJAX (Asynchronous JavaScript And XML - асинхронний JavaScript і XML) представляє собою технологію, що дозволяє при необхідності у фоновому режимі (не перериваючи роботи користувача і непомітно для нього) виконувати запити до серверу і отримувати додаткові дані для відновлення окремих частин Web-сторінки, тим самим виключаючи необхідність повторної завантаження сторінки. Наприклад, виконувати на стороні сервера перевірку правильності заповнення даних користувачем по мірі їх введення.

Без використання технології AJAX для вирішення цієї задачи є такі можливості:

- виконувати перевірку на стороні сервера, але в цьому випадку необхідно формировать нову Web-сторінок, що збільшує завантаження мережі і збільшує час очікування клієнта;
- виконувати перевірку на стороні клієнта, але при цьому часто необхідно пхранить великий обсяг інформації на комп'ютері клієнта.

Для застосування AJAX необхідні наступні компоненти:

- ✓ ***JavaScript*** (основний компонент);
- ✓ ***об'єкт XMLHttpRequest***;
- ✓ ***серверні технології*** (наприклад, PHP).

Спочатку технологію AJAX розробила фірма Microsoft як об'єкт керування ActiveX після браузера Internet Explorer. Потім фірма Mozilla створила об'єкт XMLHttpRequest з (майже) ідентичними API, який в даний час підтримується всіма сучасними браузерами.

Завдання для самостійної роботи

Завдання 1. Створити текстові файлы 1.txt та 2.txt Файл 1.txt містить назви фруктів в форматі JSON Файл 2.txt містить назви овочів в форматі JSON.

Завдання 2. Створити файл index.html, який містить JavaScript та два посилання: «фрукти» і «овочі». Натискання на посилання «фрукти» мусить завантажувати на поточну сторінку вміст файла 1.txt, і натискання на посилання «овочі» мусить завантажувати на поточну сторінку вміст файла 2.txt.

Вебсторінка має такий вигляд:

фрукти овочі

Після натискання на ссылку __фрукти__ в зелному прямокутнику має з'явитись перелік фруктів із файла 1.txt, а після натискання на посилання __овочі__ в зеленому прямокутнику має з'явитись перелік овочів із файла 2.txt.

*****УВАГА** змінюватись мусить лише вміст зеленого прямокутника. Всі інші частини сторінки залишаються незмінними.***

Звіт до практичного заняття №17

1. Роздрукувати титульну сторінку, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- З якою метою використовується технологія AJAX?
- Назвіть переваги технології AJAX.
- Назвіть недоліки технології AJAX.
- Охарактеризуйте особливості технології AJAX.

Практичне заняття № 18

Тема. Встановлення локального сервера на вибір.

Мета: вивчення основних принципів встановлення локальних серверів.

Обладнання: ПЕОМ, доступ до інтернет.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Теоретичні відомості

Локальний вебсервер

Сучасний сайт являє собою не просто набір HTML-документів, а й включає в себе безліч технологій, у тому числі серверних, таких як: SSI (Server-Side Includes, включення на стороні сервера), PHP (PHP: Hypertext Preprocessor, PHP: препроцесор гіпертексту), бази даних та багато іншого. Для ознайомлення та вивчення цих технологій помилкою буде використовувати відвідуваний робочий сайт в Інтернеті, тому має сенс встановити необхідний комплект програм на локальний комп'ютер і тестувати всі на ньому.

Найбільш популярною зв'язкою таких програм є вебсервер Apache, мова програмування PHP, система управління базами даних MySQL, оболонка для адміністрування баз даних PhpMyAdmin, шаблонизатор Smarty.

Надалі будуть розглянуті програми для домашнього комп'ютера з операційною системою Windows. Коротко опишемо, що представляють собою технології, які будуть використовуватися для створення сайту.

Вебсервер

Вебсервером називається програма, яка аналізує запити, що приходять і формує готові документи, які відправляються користувачеві. В якості

вебсервера часто виступає Apache, як найбільш сталий і популярний в Інтернеті. За різними оцінками його частка становить майже 50% від загального числа використовуваних веб-серверів.

http://www.apache.org

PHP

Популярна мова програмування, що використовується при розробці сайту.

http://www.php.net

MySQL

Система управління базами даних.

http://www.mysql.com

PhpMyAdmin

Вебінтерфейс для створення і управління базами даних MySQL. Дозволяє переглядати таблиці, змінювати їх зміст, модифікувати структуру, робити вибірку даних, сортувати інформацію. Всі дії здійснюються прямо в браузері, в спеціально розробленому під нього дружньому інтерфейсі.

http://www.phpmyadmin.net

Smarty

Потужна система шаблонів для PHP. Використовує свою власну мову, який поєднує HTML і спеціальні теги Smarty. Шаблони потрібні для поділу програмного коду та подання документа або по-іншому, для відділення логіки від змісту.

http://www.smarty.net

Всі зазначені програми та технології є відкритими, можуть бути безкоштовно завантажені і вільно використовуватися.

Можна встановити необхідні програми, скачавши їх з сайту виробника і налаштувавши під свої потреби. Однак, це часом вимагає знайомство з програмою і володіння необхідною кваліфікацією, якої у початківців немає. Оскільки мова йде не про повноцінний робочий сервер, а про комп'ютер для тестування і налагодження, то має сенс використовувати готові комплекти. Такий комплект містить в собі всі необхідні для веброзробника програми, легко налаштовується і управляється.

При розробці сайтів, більшість веброзробників, встановлюють і попередньо налаштовують свої проєкти саме на локальному комп’ютері, для цього необхідно завантажити і встановити локальний сервер.

OpenServer - це потужний та багатофункціональний інструмент для локальної розробки вебсайтів на операційній системі Windows. Він дозволяє створювати повноцінні серверні середовища на локальному комп'ютері, що значно полегшує процес розробки, тестування та налагодження вебпроєктів.

Як перенести сайт з локального сервера на хостинг.

Природно, наш локальний сервер потрібен тільки для того, щоб створити сайт. Але працювати він повинен на реальному хостингу, тому після того, як сайт готовий, потрібно його перенести на хостинг, щоб він був доступний не тільки нам, але і всім нашим майбутнім відвідувачам. Зробити це легко - за допомогою того ж файлового менеджера Total Commander, або просто закачати заархівований сайт через панель управління на хостингу. З цього боку все досить легко. Трохи складніше буде з базою даних. Тут потрібно буде спочатку створити резервну копію бази даних сайту за допомогою кнопки «Експорт» в phpMyAdmin локального сервера. У phpMyAdmin хостингу потрібно буде створити нову базу даних за тим же принципом, що і на локальному сервері в OpenServer, і за допомогою кнопки «Імпорт» залити нову базу даних на хостинг. У файлі конфігурації config.php, який тепер буде знаходитися в кореневій директорії нашого сайту також потрібно буде внести зміни - вказати ім'я бази даних, ім'я користувача бази даних, пароль до БД і ім'я сервера бази даних. Ось, власне, і все. Після того, як файли сайту залиті і залита нова база даних, можна починати користуватися нашим сайтом в інтернеті.

Завдання для самостійної роботи

1. Встановити на комп’ютер локальний сервер OpenServer 6.0 за посиланням https://ospanel.io/

2. Процес установки з скріншотами навести в електронному звіті.

3. Протестувати запуск сторінки на вебсервері, для цього в папці home створіть mysite, а в ньому за допомогою текстового редактора файл index.html, записавши туди будь-яке повідомлення.

4. Оформити звіт.

Звіт до практичного заняття №18

1. Роздрукувати титульну сторінку практичної роботи, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Для чого потрібний локальний сервер?
- Яким чином перенести сайт з локального сервера на хостинг.
- Що таке джентльменський набір?
- Що входить в пакет локального серверу?
- Яким локальним сервером користуєтеся Ви? Чому?
- Як видалити локальний сервер?

Практичне заняття №19

Тема. Знайомство з мовою написання сценаріїв PHP.

Мета: ознайомитися з базовими алгоритмічними структурами мови PHP.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Теоретичні відомості

Сценарії PHP зберігаються у файлі з розширенням php. Самі ж фрагменти коду на мові PHP відокремлюються конструкціями <? ?>. Іншими складовими php-файлу можуть бути фрагменти HTML-коду. Стрічки що починаються символами "//" вважаються коментарями. Багато стрічкові коментарії формуються за допомогою конструкції /* */. У кінці кожного оператора повинен бути символ ";". Імена змінних завжди розпочинаються із символу "$", самі ж змінні не потребуються попереднього опису. У PHP підтримуються шість основних типів даних.

Цілі числа. Крім чисел у 10-ковій системі числення у PHP реалізовано підтримку 8-кових та 16-кових чисел. Числа у 8-ковій системі числення записуються із символом "0" попереду (04528), у 16-ковій – "0x" або "0X" (0x9A5F). Над змінними цілочисельного типу виконуються арифметичні операції додавання ($a+$b), віднімання ($a-$b), множення ($a*$b), ділення ($a/$b), отримання остачі від ділення ($a%$b), інкременту ($a++), декременту($a--), порівняння: рівність ($a= =$b), нерівність ($a!=$b), більше ($a>$b), менше ($a<$b).

Дійсні числа записуються у форматі з плаваючою крапкою (16.53) та з фіксованою крапкою 16.736e5.

Рядки у мові PHP – це послідовність символів, записаних в лапках "" або '', причому при опрацюванні рядків у межах подвійних лапок ("") замість змінних підставляються їх значення. Над змінними рядкового типу виконуються операції визначення символу у вказаній позиції ($s[4] –поверне 5 символ стрічки) та об'єднання (конкатенації ($s1.$s2)).

Логічні величини набувають лише два значення істина (true) або хибність (false). Над змінними логічного типу виконуються операції заперечення (!$b), логічного множення (($a && $b) та додавання ($a || $b).

Для виводу стрічки у поле браузера використовується оператором **echo**, після якого вказати змінну або стрічку.

Операція присвоєння здійснюється за допомогою символу дорівнює (=).

Завдання для самостійної роботи

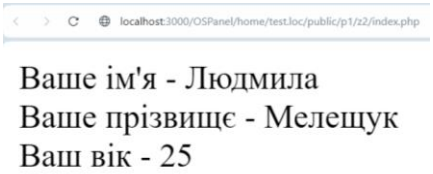
1. Створити програму, що виконує суму, різницю, добуток, частку значень двох змінних $a і $b.

Результат вивести на екран у вигляді:

Сума чисел 7 і 5 = 12

2. Створити програму, що виводить на екран ваше ім'я, прізвище і ваш вік. Ім'я, прізвище і вік повинні зберігатись у змінних - $name, $lastname, $age

Результат вивести на екран у вигляді:



3. Створити програму, що виводить на екран вашу вагу та зріст. Вага та зріст повинні зберігатись у змінних - $w та $h.

Результат вивести на екран у вигляді:

Моя вага -	52
Мій зріст -	165

4. Порахувати площу кола і об'єм кулі з радіусом r = 12 см (формули для обчислень знайти в Інтернеті). Площу кола і об'єм кулі вивести закругливши два знаки після коми.

Відображення файлу повинно бути таким:

Радіус = 12см
Площа круга = 452.39см^2
Об'єм кулі = 7238.23см^3

5. Знайти гіпотенузу прямокутного трикутника по заданих величинах його катетів.

Результат вивести на екран у вигляді:

катет	A	4 см
катет	B	3 см
гіпотенуза	C	5 см

Звіт до практичного заняття №19

1. Роздрукувати титульну сторінку практичної роботи, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Чи можна виконати php-скрипт у файлі з розширенням html?
- Чи можна використовувати дескриптори мови HTML у php-файлі?

- На Вашу думку транслятор мови PHP є інтерпретатором чи компілятором?
- З'ясуйте де виконується php-код, а де код html?
- Назвіть базові алгоритмічні структури мови php?
- Перелічіть базові типи даних мови php? Чи є обов'язковим попередній опис змінних?

Практичне заняття №20

Тема. Робота з командами розгалуження «if, else»

Мета: ознайомитися з базовими алгоритмічними структурами мови PHP.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Теоретичні відомості та технологія роботи

Операція присвоєння здійснюється за допомогою символу дорівнює (=). Іншими базовими конструкціями мови є оператори умови (if, switch) та організації циклів while, for.

Конструкція if

Вказані дії виконуються тільки тоді, коли умова є істинною.

```
if (умова)
{
    Дія;
}
```

Конструкція if...else

Якщо умова істинна, виконується дія із блоку if, в протилежному випадку — з блоку else.

```
if(умова)
{
  Дія;
}
Else
{
  Дія;
}
```

Конструкція elseif

Якщо умова блоку if істина, виконаються дії блоку if. В іншому випадку, якщо умова блоку elseif істина, виконаються дії блоку elseif. У всіх інших випадках виконається дії з блоку else.

```
if(умова)
{
  Дія;
}
elseif(умова)
```

{
Дія;
}
Else
{
Дія;
}

Конструкція switch

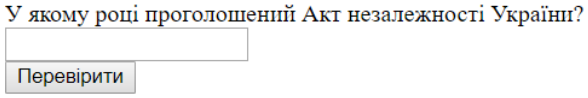
Якщо значення змінної відповідає значенню одного з блоків case, виконаються дії з цього блоку. В іншому випадку - з блоку default.

switch(Змінна)
{
case Значення 1:
Дія 1;
[break;]
case Значеннння 2:
Дія 2;
[break;]
[default: Дія;]
}

Завдання для самостійної роботи

1. Створіть запитання до теста з перевіркою чи вірно відповів користувач.

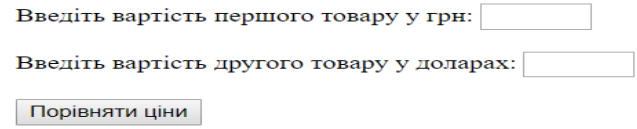
Відображення 1 файлу повинно бути таким:



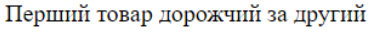
Результат вивести на екран у вигляді:

2. За перший товар ви заплатили гривнями, за другий товар заплатили доларами. Визначити який товар коштує дорожче, якщо один долар коштує 42.3 грн.

Відображення 1 файлу повинно бути таким:



Результат вивести на екран у вигляді:



3. Створити програму «Ваша вага». Вхідні дані: ваша вага, ваш ріст.
Дія програми:

1. Якщо ваша вага є нормою, тоді на екран виводиться повідомлення: «Ваша вага нормальна.»

2. Якщо ваша вага більше норми, тоді на екран виводиться повідомлення: «Ви важите більше норми, вам необхідно схуднути на … кг.»

3. Якщо ваша вага менше норми, тоді на екран виводиться повідомлення: «Ви важите менше норми, вам необхідно поправитись на … кг.»

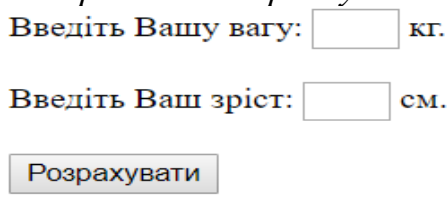
Нормальна вага обчислюється за формулою:

Норма = ріст у сантиметрах – 100

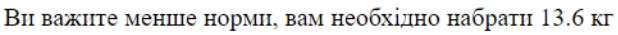
Нормою є теж вага більша і менша на 10% від норми.

Наприклад, якщо ви маєте ріст 160 см. і важите 51 кг., тоді нормою є 60кг (160-100). 10% від норми є 6кг (1/10 від 60 кг), тому норма є від 54 кг. до 66 кг, якщо ваша вага 51 кг, тоді ви важите менше норми на 3 кг. Отже відповідь комп'ютера: «Ви важите менше норми, вам необхідно набрати 3 кг. ваги».

Відображення 1 файлу повинно бути таким:



Результат вивести на екран у вигляді:



4. Створити програму «Калькулятор». По центру екрану створити таблицю з двома полями для введення чисел і чотирьома кнопками «+» , «-» , «*» , «/».

Відображення 1 файлу (index.php) повинно бути таким:

Відображення 2 файлу (p.php) при введені у поля чисел 1, 2 і натисненні кнопки [+] буде таким:

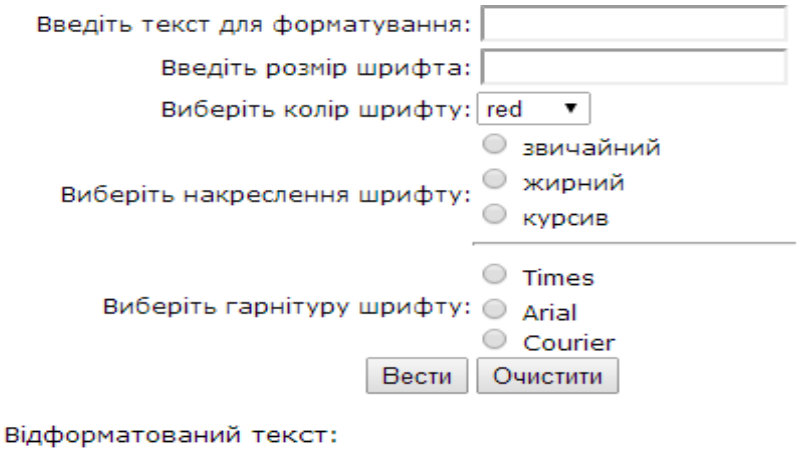
1+2=3

Відображення 2 файлу (p.php) при введені у поля чисел 3, 2 і натисненні кнопки [*] буде таким:

3*2=6

5. Розробити сценарій для форматування введеного тексту (передбачити визначення таких властивостей тексту: кольору, гарнітури, розміру, кеглю).

Відображення 1 файлу (index.php) повинно бути таким:



6. Створити програму «Тест» (12 питань) на знання мови HTML і CSS.

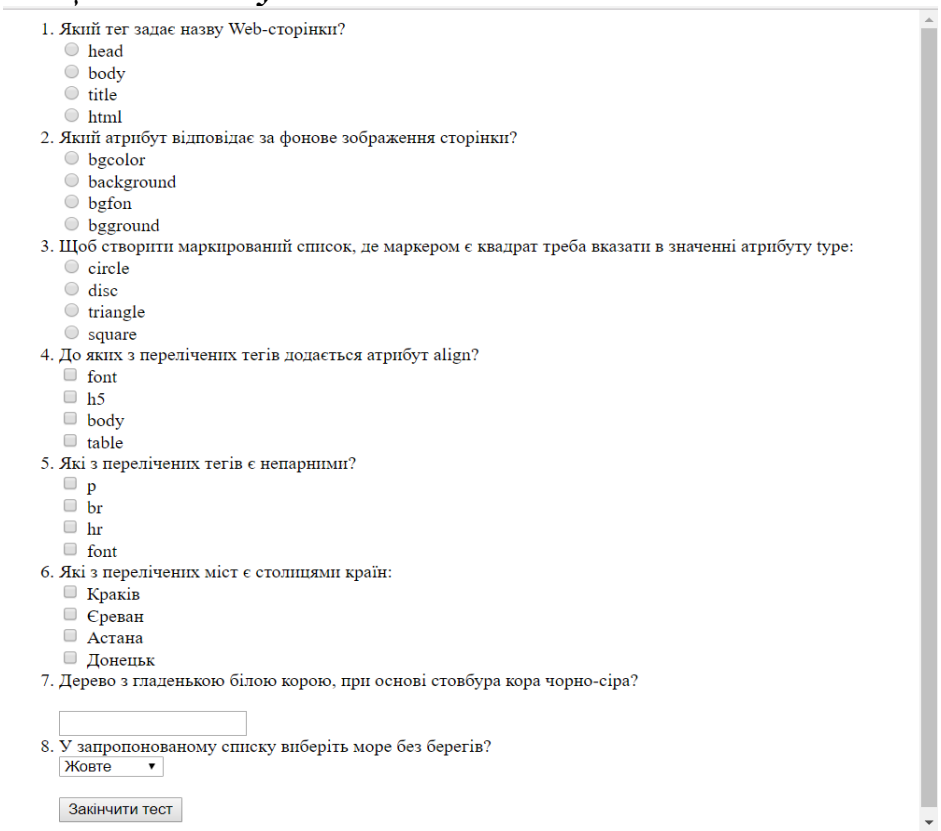
1) вибір однієї вірної відповіді з декількох запропонованих - 4 питання;

2) вибір декількох вірних відповідей з запропонованих - 4 питання;

3) введення вірної відповіді у текстове поле - 4 питання;

4) вибір вірної відповіді з «випадаючого» списку - 4 питання;

Тест естетично оформити у вигляді сайту з використанням каскадної таблиці стилів: style.css.



Результат:

Ваша оцінка = 3

Звіт до практичного заняття №20

1. Роздрукувати титульну сторінку практичної роботи, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Як записуються константи в PHP?
- Які Ви знаєте оператори присвоєння в PHP?
- Які Ви знаєте оператори порівняння в PHP?
- Які Ви знаєте оператори арифметичні в PHP?
- Які Ви знаєте стрічкові оператори в PHP?
- Дайте характеристику типам даних в PHP.
- Що таке альтернативний синтаксис?
- Як записується структура оператор if?

Практичне заняття №21

Тема. Цикли «for, while, do…while»

Мета: ознайомитися з базовими алгоритмічними структурами мови PHP.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Теоретичні відомості

Цикли призначені для багаторазового виконання набору інструкцій. У **циклі for** вказується початкове і кінцеве значення лічильника, а также крок, з яким лічильник буде змінюватися. Змінюватися лічильник може як в позитивну, так і негативну сторону. Дії виконаються стільки разів, скільки ітерацій пройде від початкового значення лічильника до досягнення кінцевого, з вказаним кроком.

```
for(початок;кінець;крок)
{
  Дія;
}
```

Цикл while. Дії будуть виконуватися до тих пір, поки умова істинна. **Цикл while** є циклом з передумовою.

```
while (умова)
{
 Дія;
}
```

Цикл do ... while. Цикл **do ... while** є циклом з післяумовою. Це означає, що спочатку буде виконуватися дія, а потім перевірятися умова. Таким чином дія завжди виконається мінімум один раз.

```
do{

  Дія;

} while (умова);
```

Оператори управління циклами break, continue

Break переривает роботу циклу. Інтерпретатор перейде до виконання інструкцій, наступних за циклом. **Continue** переривает виконання поточної ітерації циклу. Цикл продовжить виконуватися з наступної ітерації:

```
$index = 1;
```

```
while ($index < 10)
{
  echo «$index «;
  $index++;
  if ($index == 5)
    break;
}
$index = 0;
while ($index < 10)
{
  $index++;
  if ($index == 5)
    continue;
  echo «$index»;
}
```

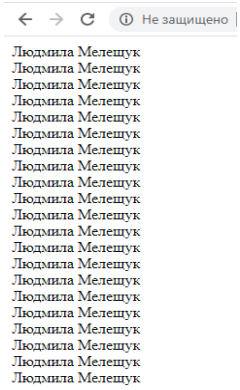
Оператор циклу foreach

Цикл foreach Дуже зручний при роботі з масивами. Зазначені дії виконуються для кожного елемента масиву $array, при цьому $key - номер елемента масиву $array, $ value - значення цього елемента.

```
foreach ($array as [ $key => ] $value)
{
 Дія;
}
```

Завдання для самостійної роботи

1. Вивести на екран своє ім'я і прізвище 30 разів (використати цикл «for»).



2. Створіть програму знаходження суми чисел від 1 до 100. Використати цикл «for».

Сума чисел від 1 до 100 = 5050

3. Створіть програму знаходження добутку всіх чисел кратних 3 і менших 20. Використати цикл «while».

Добуток чисел кратних 3 і менших 20 = 524880

4. Створіть програму знаходження суми чисел (використати цикл «while»):

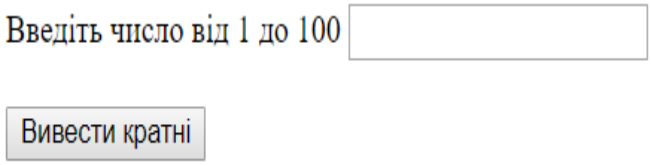
Сума чисел: 1, 1/2, 1/3, 1/4 ... 1/20 = 3.5977396571437

5. Створіть програму виводу на екран всіх чисел кратних введеному і менших = 100 (цикл «while»). Програма також повинна порахувати кількість таких (кратних) чисел.

Якщо, наприклад користувач введе число 15, тоді комп'ютер повинен вивести на екран числа: 15, 30, 45, 60, 75, 90 , кількісь = 6.

Якщо, наприклад користувач введе число 25, тоді комп'ютер повинен додавати числа: 25, 50, 75, 100 , кількісь = 4.

Відображення 1 файлу (index.php) може бути таким:



Відображення 2 файлу (p.php) може бути таким:

Числа кратні 25: 25 50 75 100 . Їх кількість - 4

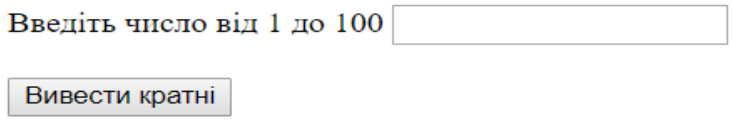
6. Створіть програму знаходження суми всіх чисел кратних введетному і менших = 100 (цикл «while»).

Якщо, наприклад користувач введе число 30, тоді комп'ютер повинен додавати числа: 30+60+90 , результат = 180;

Якщо, наприклад користувач введе число 20, тоді комп'ютер повинен додавати числа: 20+40+60+80+100 , результат = 300;

Якщо, наприклад користувач введе число 40, тоді комп'ютер повинен додавати числа: 40+80 , результат = 120.

Відображення 1 файлу (index.php) може бути таким:



Відображення 2 файлу (p.php) може бути таким:

Числа кратні 20: 20 40 60 80 100 . Їх кількість - 5

Звіт до практичного заняття №21

1. Роздрукувати титульну сторінку практичної роботи, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Охарактеризуйте оператор циклу з пердумовою while?
- Як записується структура оператор switch?
- Опишіть з якою метою використовуються цикли?
- Охарактеризуйте синтаксис циклів?
- Що таке цикли?

Практичне заняття №22

Тема. Розробка PHP сценаріїв з використанням масивів.

Мета: ознайомитися з типом даних в PHP масивами і стандартними функціями роботи з массивами.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Теоретичні відомості

Тип array (масив)

Масив у PHP являє собою впорядковану карту – тип, що перетворює значення в ключі. Цей тип оптимізовано у декількох напрямках, тому ви можете використовувати його як масив, список (вектор), хеш-таблицю (що є реалізацією карти), стек, черга і т.д. Оскільки ви можете мати як значення інший масив PHP, можна також легко емулювати дерева.

Задати масив можна за допомогою конструкції array() або безпосередньо задаючи значення його елементам.

Визначення за допомогою array()

array ([key] => value,[key1] => value1, ...)

Мовна конструкція array() приймає як параметри пари **ключ** => значення, розділені комами. Символ => встановлює відповідність між значенням і його ключем. Ключ може бути як цілим числом, так і рядком, а значення може бути будь-якого ніснуючого в PHP типу. Числовий ключ масиву часто називають індексом. Індексування масиву в PHP починається з нуля. Значення елемента масиву можна одержати, вказавши після імені масиву в квадратних дужках ключ шуканого елемента. Якщо ключ масиву являє собою стандартний запис цілого числа, то він розглядається як число, в противному випадку – як рядок. Тому запис $a[«1»] рівносильний запису $a[1], так само як і $a[«-1»] рівносильне $a[-1].

Якщо для елемента ключ не заданий, то в якості ключа береться максимальний числовий ключ, збільшений на одиницю. Якщо вказати ключ, якому вже було присвоєно якесь значення, то воно буде перезаписано.

Починаючи з PHP 4.3.0, якщо максимальний ключ – від'ємне число, то наступним ключем масиву буде нуль (0).

Якщо використовувати як ключ TRUE або FALSE, то його значення переводиться відповідно в одиницю і нуль типу integer. Якщо використовувати NULL, то замість ключа одержимо порожню стрічку. Можна використовувати і сам порожню стрічку як ключ, при цьому її треба брати в лапки. Так що це не те ж саме, що використання порожніх квадратних дужок. Не можна використовувати як ключ масиви й об'єкти.

Визначення за допомогою синтаксису квадратних дужок

Створити масив можна, просто записуючи в нього значення. Як ми вже говорили, значення елемента масиву можна одержати за допомогою квадратних дужок, всередині яких потрібно вказати його ключ наприклад, $book[«php»]. Якщо вказати новий ключ і нове значення наприклад, $book[«new_key»]=«new_value», то в масив додасться новий елемент. Якщо ми не вкажемо ключ, а тільки привласнимо значення $book[]=«new_value», то новий елемент масиву буде мати числовий ключ, на одиницю більший максимального існуючого. Якщо масив, у який ми додаємо значення, ще не існує, то він буде створений.

Для того щоб змінити конкретний елемент масиву, потрібно просто присвоїти йому з його ключем нове значення. Змінити ключ елемента не можна, можна тільки знищити елемент (пару ключ/значення) і додати нову. Щоб видалити елемент масиву, потрібно використовувати функцію unset().

Помітимо, що, коли використовуються порожні квадратні дужки, максимальний числовий ключ шукається серед ключів, що існують у масиві з моменту останнього переіндексування. Переіндексувати масив можна за допомогою функції array_values().

Завдання для самостійної роботи

1. Заповнити масив $m[0…9] довільними числами використавши функцію array(). Виведіть на екран цей масив у таблиці:

0 ---> 56
1 ---> 2
2 ---> 48
3 ---> 77
4 ---> 9
5 ---> 11
6 ---> 15
7 ---> 68
8 ---> 35
9 ---> 65

2. Використавши команду «=« і функцію rand() заповнити масив $m[1…10] довільними числами. Виведіть на екран цей масив у таблиці:

1	45
2	27
3	44
4	51
5	81
6	56
7	4
8	20
9	70
10	9

3. Створіть 3 масиви:

$name[0..9] - імена студентів,

$ukr[0..9] - оцінки за рік з української мови,

$eng[0..9] - оцінки за рік з англійської мови.

Програма повинна створити 4 масив $sb[0..9] - середній бал з вивчення мов кожного студента, знайти середній бал з вивчення мов всіх студентів.

№	Ім'я	укр.мова	англ.мова	С/Б
1	Марія:	7	8	7.5
2	Петро:	10	10	10
3	Христина:	9	10	9.5
4	Аня:	9	8	8.5
5	Ярослав:	8	10	9
6	Роман:	11	12	11.5
7	Віта:	4	7	5.5
8	Катя:	11	10	10.5
9	Микола:	10	10	10
10	Олександр:	8	9	8.5

Середній бал учнів = 9.05

4. Створіть два масиви:

$a[0..9] довільне число від 10 до 20 (використайте rand()),

$b[0..9] довільне число від 30 до 40 (використайте rand()),

Створіть, через обчислення, масив $c[0..9], у якому:

$c[0] = корінь квадратний(квадрат $a[0] + квадрат $b[0])

$c[1] = корінь квадратний(квадрат $a[1] + квадрат $b[1])

$c[2] = корінь квадратний(квадрат $a[3] + квадрат $b[3])

...

$c[9] = корінь квадратний(квадрат $a[9] + квадрат $b[9])

Виведіть на екран таблицю з значеннями a, b, c.

Для рішення цієї задачі у програмі використайте тільки 1 цикл.

a	b	c
17	35	7.21
15	33	6.93
11	32	6.56
14	34	6.93
16	31	6.86
19	35	7.35
14	31	6.71
20	36	7.48
17	34	7.14
12	37	7

5. Створити двовимірний масив «Inform» з оцінками студентів з інформатики:

№	Ім'я учня	1	2	3	4	5	6	7
1	Василь	12	н	8	н	н	10	10
2	Марія	8		н		4	н	7
3	Ірина	н	11		н	7		6
4	Петро		8	8	9		н	7
5	Саша	н	н	н		9	н	7
6	Коля	9		8	10	10	11	12
7	Гріша	9					7	н
8	Оля	н	н			6		
9	Галя	4	5		н	н	н	5
10	Тарас	10	10	8	н	11	12	9

Колонки » 1 «, « 2 «, « 3 «, ... , « 7 « - номер теми.

«н» - пропуск заняття.

« « (пустота в комірці масиву) - учень був на занятті, але оцінку не отримав.

Вивести масив на екран у вигляді таблиці.

6. Написати програму, яка перевіряє, чи є введена з клавіатури квадратна матриця «магічним квадратом». Магічним квадратом називається матриця, у якої сума чисел в кожному горизонтальному ряду, в кожному вертикальному і по кожній з діагоналей одна і та ж (див. нижче).

2 9 4
7 5 3
6 1 8

Звіт до практичного заняття №22

1. Роздрукувати титульну сторінку практичної роботи, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Що таке масив?
- Яким чином задається синтаксис масиву?
- Що таке ключ?
- Як відбувається індексування у масиві?

Практичне заняття №23

Тема. PHP функцій, визначені користувачем.

Мета: навчится задавати власні PHP функції та використовувати їх в програмних кодах.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Завдання для самостійної роботи

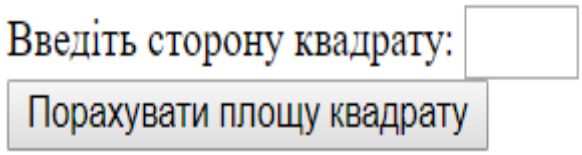
1. Зробити таку послідовність дій:

1) Створити файл для власної бібліотеки функцій - **myfunctions.php**.

2) У файлі **myfunctions.php** створити функцію **sqr($x)**. Функція **sqr($x)** повинна повернути площу квадрату з стороною **$x**.

3) Створити 1 файл: **index.php**.

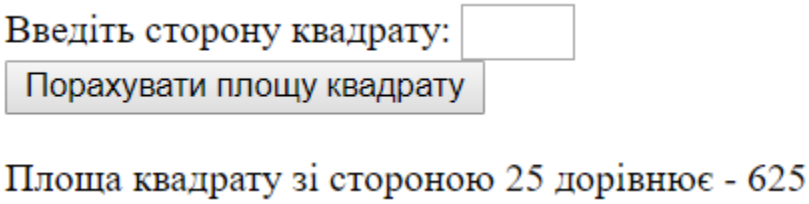
4) У файлі **index.php** створити форму для ввода сторони квадрату:



5) З допомогою команди **include** під'єднайте файл бібліотеки - **myfunctions.php**, так ми зможемо функції прописані у файлі **myfunctions.php**.

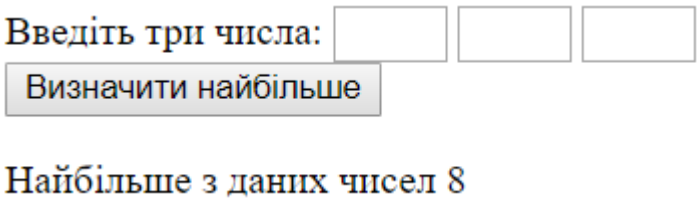
6) Порахуйте площу квадрату використавши функцію **area_sqr($x)**, значення площі виведіть на екран.

ЗАУВАЖЕННЯ! В подальшій роботі всі власні функції ми будемо записувати у файл бібліотеки.



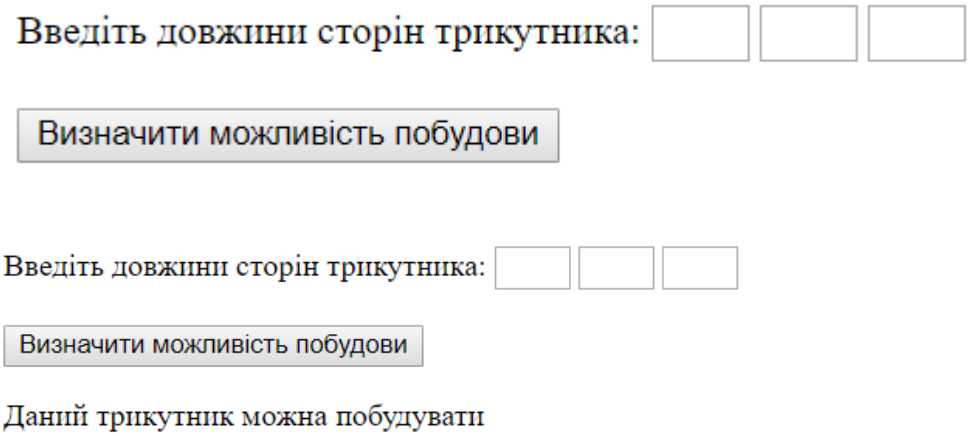
Площа квадрату зі стороною 25 дорівнює - 625

2. Створити програму, що визначає яке з трьох введених чисел є найбільшим. Програма використовує функцію max_num_of_three($x, $y, $z), що повертає значення найбільшого з трьох введених. У випадку, коли всі числа рівні, функція повертає значення буть-якого числа. У випадку, коли два числа є рівні і більші третього, тоді вивести значення одного з цих двох.



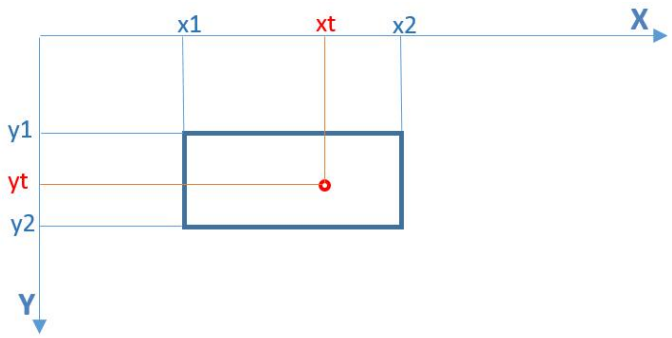
3. Створити функцію this_triangle($x, $y, $z), що визначає можливість побудови трикутника з сторін у яких довжини рівні $x, $y, $z. Функція повинна повертати логічну константну true, якщо побудова трикутника можлива, у іншому випадку функція повинна повертати false. У функції this_triangle($x, $y, $z), для визначення найбільшої сторони, використайте функцію max_num_of_three($x, $y, $z).

Створіть програму, яка визначає можливість побудови трикутника з введених трьох сторін.



4. Створіть функцію, яка визначає чи точка з координатами xt, yt знаходиться всередині прямокутника з координатами x1, y1, x2, y2, назва функції:

inside_rectangle($xt, $yt, $x1, $y1, $x2, $y2)



Введіть координати точки: x = y =

Введіть координати прямокутника: x_1 = x_2 = y_1 = y_2 =

Визначити розташування точки відносно прямокутника

Введіть координати точки: x = y =

Введіть координати прямокутника: x_1 = x_2 = y_1 = y_2 =

Визначити розташування точки відносно прямокутника

Точка з координатами 8, 4 знаходиться всередині прямокутника з координатами 0, 10, 0, 15

5. Створіть функцію is_in_array($x, $v), яка у одновимірному масиві $x шукає значення $v. Функція повертає true у випадку, якщо значення $v є у масиві. Використайте цю функцію у програмі.

Введіть прізвище учня:

Продовжити

Введіть прізвище учня:

Продовжити

В класі є учень з прізвищем Жук

Звіт до практичного заняття №23

1. Роздрукувати титульну сторінку практичної роботи, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Для чого потрібні функції?
- Як записується загальний вигляд функції?
- Що таке аргумент функції?
- Які Ви знаєте способи передачі функції за допомогою аргументів?

Практичне заняття №24

Тема. Розробка PHP сценаріїв з використанням рядкових змінних

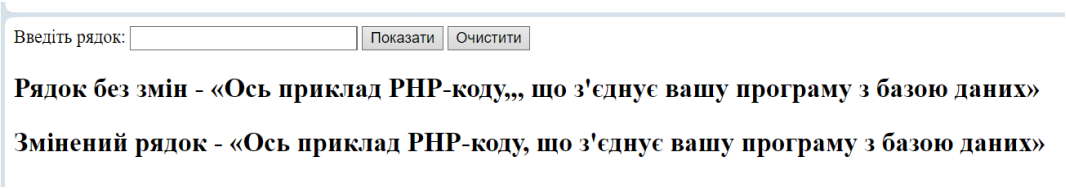
Мета: ознайомитися з можливостями PHP при роботі зі стрічками та навчится використовувати їх в своїх сценаріях.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Завдання для самостійної роботи

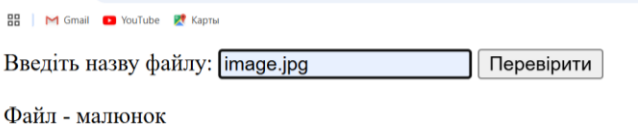
1. Створити програму, яка з рядка $r видаляє лишні коми. *Наприклад:* у рядку «Ось приклад PHP-коду,,, що з'єднує вашу програму з базою даних» є дві лишні коми.



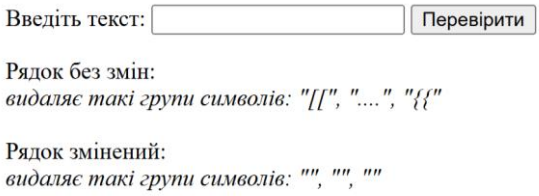
2. Створити програму, яка кожен символ з рядка $r виводить в окремому рядку і замінює прогалину на «_». *Наприклад: якщо користувач ввів рядок.*

3. Створити програму, яка з рядка $r видаляє три зайві символи, замінюючі їх на 1.

4. Створити функцію this_picture($r), яка визначає чи переданий їй рядок $r є назвою файлу у якому міститься графічне зображення. Функція повинна відслідковувати тільки такі типи файлів: *.jpg, *.png, *.gif.



5. Створити фунцію pr($r), що з рядка видаляє такі групи символів: «[[», «….», «{{».



6. Користувач вводить рядок, програма перетворює цей рядок у інший, який складається лише з символів які зустрічаються у першому рядку більше ніж 1 раз.

7. Перевірити чи введене користувачем слово є палідром (то що читається з ліва на право і з права на ліво однаково).

Наприклад: Анна, дід, Пилип..

Звіт до практичного заняття №24

1. Роздрукувати титульну сторінку практичної роботи, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Що таке стрічка?
- Назвіть основні способи її представлення?
- Як відбувається пошук елемента в стрічці?
- Назвіть функції вирізання під стрічки.
- Як відбувається поділ і з’єднання стрічки?

Практичне заняття №25

Тема. Розробка PHP сценаріїв з використанням файлів.

Мета: ознайомитися із засобами роботи з файлами мови PHP.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Теоретичні відомості та технологія роботи

Для роботи із файловою системою Web-сервера в мові реалізовано наступні процедури та функції:

file_exists(string назва_файлу) – повертає значення істини, якщо файл із заданим ім’ям існує.

is_file (string назва_файлу) – перевіряє існування вказаного файлу, а також можливість виконання з ним операцій читання-запису.

filesize(string назва_файлу) – повертає розмір файлу в байтах.

fopen (string назва_файлу, string режим доступу) – відкриває вказаний файл у вказаному режимі, "r" – читання, "w" –запису, "a" – додання даних, а також повертає цілочисельне значення (ідентифікатор відкритого файлу). Зазначений файл може бути розміщений на комп'ютері, на якому виконується Web-сервер, на іншому HHTP чи FTP-сервері.

fclose (int ідентифікатор відкритого файлу) – закриває файл із вказаним ідентифікатором.

fread(int ідентифікатор, int кількість_байт) – зчитує із файлу із заданим ідентифікатором вказану кількість байт.

fgetc(int ідентифікатор) – зчитує із файлу один символ.

fgets(int ідентифікатор, int довжина) – зчитує із файлу стрічку вказаної довжини.

file(назва_файлу) –завантажує файл в індексований масив, кожен елемент якого відповідає одній стрічці файлу.

int fwrite(int ідентифікатор, string стрічка) – здійснює запис значення рядкової змінної у файл.

int readfile (string назва_файлу) – зчитує вміст файлу і виводить його у вікно браузера.

Завдання для самостійної роботи

1. Розробити сценарій, що перевірятиме чи в даному файлі міститься введена стрічка.

2. Розробити сценарій для виконання авторизації користувача. Дані про користувачів зберігати у файлі виду:

назва_користувача_1:пароль_користувача_1
назва_користувача_2:пароль_користувача_2
.....
назва_користувача_N:пароль_користувача_N

3. Розробити сценарій, який буде заносити в файл з форми потрібні дані і відразу ж їх виводити в необхідному форматі.

Вказівка*: повинно бути два файли: перший повинен містити саму форму і виводити вміст, другий повенен додавати дані в файл і переадресовуватись відразу на перший (* ***Header(«Location: файл на який переадресовується»);*** *).*

4. Розробити сценарій з допомогою якого можна проводити опитування і перегляд його результатів.

Звіт до практичного заняття №25

1. Роздрукувати титульну сторінку практичної роботи, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Опишіть етапи роботи із файлами.
- Що таке дескриптор файлу?
- Опишіть результат виклику функцій fopen(“f.dat”,”r”) та fopen(“f.dat”,”w”) у випадку коли файл f.dat не існує.
- Якими, на Вашу думку, можуть бути наслідки не закриття файлу
  - ✓ на локальному комп’ютері?
  - ✓ на HTTP чи FTP сервері?
- Опишіть відмінності функцій fgets() та file(). Який недолік має використання функції fgets()?

Практичне заняття №26

Тема. Сценарії PHP для виконання основних операцій з базами даних MySGL.

Мета: набути навиків роботи із сервером СУБД MySQL через сценарії PHP. Удосконалити вміння створювати SQL-запити.

Обладнання: ПК з ОС Windows.

Програмне забезпечення: браузери різних виробників, текстовий редактор Notepad++, Sublime Text 3.

Теоретичні відомості

MySQL – це програма-сервер, що працює на комп'ютері. Клієнтські програми (наприклад, сценарії) відправляють їй спеціальні запити за допомогою мережевих засобів, сервер їх опрацьовує за спеціальними запитом клієнта результат або його частина передає клієнту.

Перш ніж працювати з базою даних, необхідно встановити з нею мережеве з’єднання, а також провести авторизацію користувача. Для цього використовується функція mysql_connect(). **intmysql_connect([сервер] [користувач] [пароль]).** З'єднання з MySQL-сервером буде автоматично закрите після закінчення роботи сценарію, або ж при виклику функції mysql_close().

Для формування запитів до бази даних застосовується функція mysql_query, яка повертає ідентифікатор результуючого набору даних. Результат відразу не пересилається клієнту? Так от, щоб працювати з ним надалі, і використовується цей ідентифікатор. Існує дуже багато функцій, які приймають його як параметр і повертають ті або інші дані.

Синтаксис фнкції такий:

int mysql_query(string $query [,int $link_identifier])

Параметр $query є запитом-стрічкою, який повинен бути описаний за

правилами мови SQL. Необов'язковий параметр $link_identifier використовується для відновлення втраченого зв'язку із сервером MySQL.

Розглянемо детальніше основні конструкції для формування SQL-запитів:

1. insert into table_name (назва_таблиці1 назва_таблиці2 ...) values('значення1', 'значення2'...) –запит додає до таблиці назва_таблиці запис, поля якого значення1, набувають значень val1

2. delete from назва_таблиці where вираз – Знищує із таблиці записи, для яких є істинним вираз. Параметр вираз є логічним виразом. Наприклад:(id<10) and (age=25)

3. select * from where вираз [order by назва_поля [desc]] – проводить пошук записів, які відповідають заданому виразу. Якщо отриманих записів декілька, то при заданій конструкції **order by** вони будуть відсортовані за тим полем **назва_поля** (якщо задано слово **desc**, то сортування буде проведено у зворотному порядку). Символ * вказує на те що слід отримати дані усіх полів, записи, яких відповідають умові. Тобто замість стмволу "*" можна вказувати потрібні поля.

4. Update назва_таблиці SET (назва_поля1='значення1', назва_поля2='значення2', ...) where вираз – у таблиці для записів, які відповідають умові зазначенні полянабувають відповідних значень.

Після виконання запиту запит виконаний і отриманий його ідентифікатор потрібно отримати власне дані – результат. Для цього найбільш часто використовується функція **array mysql_fetch_row(int $result)**, яка повертає масив-список із значеннями полів чергового рядка результату $result. Якщо вказівник поточної позиції результату був встановлений після останнього запису), то функція повертає значення false. Поточна позиція переміщується до наступного запису, так що черговий виклик mysql_fetch_row() поверне наступний рядок результату.

Функція mysql_fetch_array() повертає черговий рядок результату у вигляді асоціативного масиву, де кожному полю відповідає елемент з ключем, що співпадає з ім'ям поля.

Завдання для самостійної роботи

1. **Створити таблиці з наступними полями:**
   <перші 3 літери вашого прізвища>_user
   id - унікальний ідентифікатор
   login – прізвисько;
   password - пароль;
   name - ім'я;
   lastname – прізвище;

   <перші 3 літери вашого прізвища>_forum

id - унікальний ідентифікатор
tema – тема для обговорення;
autor - автор теми;
data - дата і час створення теми;

<перші 3 літери вашого прізвища>_comment
id - унікальний ідентифікатор
id_tema – тема до якоъ наледить коментар;
commentar - текст повыдомлення;
autor - автор коментаря;
data - дата і час створення коментаря;

Типи полів вибрати самостійно

2. Розробити PHP сценарії (в окремій папці **admin_user**), для виводу вмісту таблиці **<...>_user** з можливістю додавання, редагування і знищення її записів.

3. Розробити PHP сценарій (**index.php**) авторизації користувачів, відомості про яких містяться в таблиці**<...>_user**. В разі успішоного проходження авторизації відбувається перехід на сторінку **forum.php.**

4. Розробити PHP сценарій сторінки **forum.php**, в якій виводяться всі записи таблиці **<...>_forum** у такому вигляді:

Тема	**Автор**	**дата і час**	**Кількість коментарів**
Назва теми, яка є гіперпосилкою на сторінку**comment.php**			

і форму для додавання запису в цю таблицю (форма повинна містити лише одне текстове поле, інша інформація про дату, автора, та унікальний ідентифікатор, повинні шукатись)

5. Розробити PHP сценарій сторінки **comment.php**, яка виводитиме лише ті записи таблиці**<...>_commenet**, які стосуються обраної теми, і на якій знаходиться форма для введення повідомлення на обрану тему (для додавання запису в таблицю **<...>- comment** з **id_tema** на обрану тему).

Звіт до практичного заняття №26

1. Роздрукувати титульну сторінку практичної роботи, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Яким чином відбувається з’єднання з базою даних?
- Як відбувається вибір бази даних?
- Як одержати список полів таблиці?
- Як записати дані у базу даних?
- Запишіть синтаксис функції mysql_fetch_array.
- Які оператори використовуються для створення бази даних (БД).
- Які оператори використовуються для знищення бази даних.
- Які оператори використовуються для зміни бази даних.
- Як відбувається вибірка даних із БД.

Практичне заняття №27

Тема. Створити сайт, у якому для наповнення сторінок використовують інформацію з бази даних.

Мета: закріплювати знання та вміння теоретичного матеріалу по веброзробці на практиці.

Обладнання: ПК з ОС Windows або Linux.

Програмне забезпечення: браузери різних виробників - IE, Opera, Firefox, Chrome, Notepad++, Sublime Text 3.

Завдання для самостійної роботи

1. Створити сайт на вільну тему.

2. Ваш сайт повинен черпати інформацію з бази даних (назви пунктів меню, наповнення сторінок).

3. Окрім меню, на вашому сайті повинна бути кнопка «Змінити дизайн», при натисненні цієї кнопки дизайн сайту повинен змінитись, наприклад з вертикального, меню стає горизонтальним. При зміні дизайну може змінюватись і кольорова гама, шрифти, тощо. Дизайн повинен бути “гумовим”.

Звіт до практичного заняття №27

1. Роздрукувати титульну сторінку практичної роботи, зробити скріншоти виконаних завдань, роботу прикріпити у classroom.

Практичне заняття №28

Тема. Хостинг та FTP-доступ основні поняття, можливості, інструменти. Розміщення сторінки в Інтернет

Мета: ознайомлення з основними поняттями хостингу, одержання навичок у FTP-доступі до WEB-ресурсів на сервері, дослідження можливостей основних інструментальних засобів FTP-доступу. Практичне засвоєння розміщення ресурсів на хостингу.

Обладнання: ПК з ОС Windows або Linux.

Програмне забезпечення: браузери різних виробників - IE, Opera, Firefox, Chrome, Notepad++, Sublime Text 3.

Теоретичні відомості

Вибір доменного імені

Після створення сайту постає питання реєстрації домену, або доменного імені. В Інтернеті адреси прийнято поділяти на домени за тематичною (організаційною) чи географічною (національною) належністю. Географічні зони можуть поділяться на організаційні (com.ua, gov.ua, org.ua тощо).

Перш ніж зареєструвати домен необхідно визначиться, в якій доменній зоні реєструватися. Якщо компанія орієнтована на український ринок, то необхідно реєструватися в одній з українських організаційних зон (.com.ua, biz.ua, info.ua) або в одній з українських обласних зон (kiev.ua, lviv.ua, zp.ua). Кожен обласний центр має свою доменну зону.

Домени першого рівня у свою чергу поділяються на домени другого рівня (наприклад, в адресі www.company.kiev.ua - .ua є зоною першого рівня, яка вказує, що компанія знаходиться в Україні, .kiev - це зона другого рівня, яка вказує, що це київська кампанія).

Компанія може зареєструвати домен як в обласній, так і в організаційній українській доменній зоні. Якщо в компанії немає філій в інших областях України, то можна зареєструвати домен в обласній зоні, тоді у користувачів компанія асоціюватиметься з містом в якому знаходиться компанія, якщо компанія має філії, то краще реєструватися в українській організаційній зоні, тоді користувачі будуть сприймати компанію як всеукраїнську. Якщо компанія має зареєстровану торгову марку, тоді можна зареєструвати домен «торговамарка.ua».

Як правило, українські компанії реєструють одне доменне ім'я в певній українській доменній зоні, хоча адрес в одного сайту може бути множина. Якщо для сайту потрібно зареєструвати кілька доменних адрес (company.ua, company.com.ua, company.kiev.ua, company.com), то варто одне з доменних імен вибрати як основне. Воно буде використовуватися в логотипі, рекламі, візитках, бланках тощо, решта доменів буде скеровуватися на «основний» домен.

Правила вибору доменного імені:

1. Доменне ім'я повинне бути коротким.
2. Доменне ім'я повинно запам'ятовуватися.
3. Доменне ім'я не повинне допускати різночитань.
4. Доменне ім'я може складатися з букв латинського алфавіту, цифр і символу «-»(дефіс).

Доменне ім'я повинне бути коротким

Чим же добре коротке доменне ім'я? По-перше коротке доменне ім'я легше запам'ятовується і його простіше розмістити в логотипі, по-друге в

короткому доменному імені важче допустити помилку при наборі в адресному полі браузера.

За часів становлення Інтернет можна було зареєструвати практично будь-яке доменне ім'я, але вданий час більшість коротких благозвучних доменів вже є зайнятими, і тепер підібрати коротке ім'я стає все складніше.

Доменне ім'я повинно запам'ятовуватися

В першу чергу доменне ім'я повинне асоціюватися з назвою компанії, тому кращим варіантом буде домен company.ua або company.com.ua, щоб клієнти за найменуванням компанії швидко знайшли її сайт.

Доменне ім'я не повинне допускати різночитань.

Доменна адреса може бути транскрипцією латинськими буквами кириличних назв і тут можуть бути неприємні сюрпризи. Наприклад, компанія «Киев Пассажирский» в транскрипції може виглядати як KievPassazhirsky і як KievPassazhirskiy, а в транскрипції з української як KyivPasazhirsky. Деякі користувачі можуть помилитися і писати як з двома «s» так і з однією, тому для такої компанії краще підібрати щось просте, наприклад, www.kpas.ua.

Небезпечно використовувати в доменному імені цифру «0» і букву «О», оскільки користувачі часто їх плутають (коли логотип виконано у вигляді адреси zOO, то не всім зрозуміло які символи використано в назві домену). Якщо все ж таки необхідно використовувати нуль або букву «О», слід зареєструвати домен з правильним і неправильним написанням і переадресувати домен з неправильним написанням на основний домен (правильний).

Доменне ім'я може складатися з букв латинського алфавіту, цифр і символу (дефіс), починатися і закінчуватися буквою латинського алфавіту або цифрою. Довжина імені кожного домену (між розділовими точками) не може перевищувати 63 знаки. Мінімальна довжина доменного імені (без врахування розділових точок і назви домену) прийнята рівною 2 (для зон .biz і .info - 3 символи).

Поради з вибору і використання доменних імен:

1. Реєструйте кілька доменів

2. Використовуйте ключові слова в назві домену.

3. Одночасно реєструйте домен в національній та організаційній зонах (kiev.ua, com.ua

4. Використовуйте «чужі» географічні зони (.tv, .co, .cc, .to).

5. Використовуйте міжнародні домени (.com, .biz, .info).

6. Використовуйте назву поєднання організаційних або географічних зон і назву домену.

7. В назві домену краще уникати дефіси.

8. Не використовуйте як основний доменне ім'я, що схоже на домен конкурента.

9. Реєструйте домени, що звільняються.

1. Реєструйте декілька доменів.

Якщо назва компанії українською та російською мовою пишеться по-різному, то варто зареєструвати кілька доменів в різних варіантах написання. Таким чином, будь-яке написання назви компанії приведе користувача на сайт.

2. Використовуйте ключові слова в назві домену.

За багатьма дослідженнями деякі користувачі вгадують адресу, тобто беруть назву того, що хочуть знайти і додають .cot, відповідно українські користувачі додають .com.ua або .kiev.ua. Тому, якщо компанія продає монітори, то окрім домену company.com.ua, бажано зареєструвати домен monitor.com.ua чи monitor.kiev.ua. Також, не варто забувати про назви доменів в множині, наприклад, monitors.com.ua чи monitors.kiev.ua.

Крім того, в пошукових системах за відповідними ключовими словами сайт буде вищим в результатах пошуку, оскільки в назві домену присутнє ключове слово.

3. Одночасно реєструйте домен в кількох зонах.

Як правило, визначившись з назвою домену багато київських компаній, реєструють домен в зоні com.ua, або .kiev.ua, хоча, слід реєструвати доменне ім'я в обох зонах, оскільки компанія конкурент може зареєструвати друге доменне ім'я і скерувати його на свій сайт. Користувачі часто плутають .com.ua і kiev.ua (так вже історично склалося, що Київ став флагманом розвитку Інтернет в Україні, тому зони .com.ua і .kiev.ua для українського користувача стали рівнозначними) тому реєструйте домен в обох зонах, тоді навіть допустившись помилки користувач все одно потрапить на сайт.

1. Не реєструйте домен, якщо він зайнятий хоча б в одній із зон .com.ua або .kiev.ua компанією, яка займається тим же бізнесом.

2. Якщо компанія конкурент зареєструвала домен лише в одній із зон .com.ua або .kiev.ua, то необхідно зареєструвати цей домен в іншій зоні і скерувати його на свій основний домен тоді користувачі помилково зайдуть на ваш сайт і можливо знайдуть там те, що шукали.

3. Якщо кампанія є власником торгової марки, то слід зареєструвати домени в трьох зонах: .ua, .com.ua і .kiev.ua.

4. Якщо в компанії є філії в обласних центрах, то слід зареєструвати домени і у відповідних українських обласних доменних зонах.

Використовуйте «чужі» географічні домени (.tv, .ag, .co).

Домен .ag (держава Антигуа) як альтернатива домену .com може особливо зацікавити акціонерні компанії орієнтовані на німецький ринок. В

німецькій мові скорочення Ag позначає «акціонерне суспільство» (Die Aktiengesellschaft) і широко використовується як позначення типу компанії, подібно до англійських скорочень Inc. і Corp.

Компанія VeriSign купила права на управління доменною зоною .tv (маленька острівна держава Тувалу) і тепер адмініструє домен .tv - телевізійним, і тепер реєстрація домену в цій зоні вельми приваблива для телекомпаній і сайтів, що мають телевізійну скерованість.

Національний домен Колумбії .co добре асоціюється із словами Company та commercial і може, таким чином, стати альтернативою для переповненої зони .com.

Слід пам'ятати, що користувачі розділяють сайти за географічною і організаційною ознакою, тому, перш ніж зареєструвати домен в «чужій» доменній зоні - необхідно все ретельно зважити. Реєстрація в «чужій» доменній зоні буде виправданою, якщо отримана адреса добре запам'ятовується і асоціюється з родом діяльності компанії.

У поєднанні із закінченням (організаційним доменом) назва домену може дати несподівано добрий результат. Наприклад, kuvyr.com - така адреса дуже добре запам'ятовується, оскільки домен kuvyr спільно із зоною .com читається як «кувирком». Також добре запам'ятовуються словосполучення daleko.edu (далеко еду), доменне ім'я daleko само по собі добре запам'ятовується, але спільно із закінченням .edu (освітня доменна зона) - запам'ятовується ще краще. Такий домен міг би підійти як основний для туристичної фірми, хоча в даному випадку потрібно буде переконати реєстратора, що домен реєструється для освітнього сайту, оскільки .edu це організаційна зона для освітніх сайтів.

Використовуйте міжнародні домени (.com, .biz, .info)

Якщо компанія працює на міжнародному ринку, то не слід обмежуватися лише українським доменом - зареєструйте домен в зоні .com, .biz або .info. Скеровуйте міжнародні домени на англійську версію сайту, а українські на українську (або російську - якщо немає української версії). Відповідно для роботи на міжнародному ринку використовуйте міжнародний домен (у візитках, рекламі, бланках), а на внутрішньому ринку український домен. Тоді, іноземні користувачі відразу потраплять на англійську версію сайту, а українські відповідно на українську версію.

Дефіси рідко зустрічаються, тому, краще обмежити їх використання в назвах доменів, хоча цього не завжди можна уникнути, оскільки в зоні .com всі благозвучні домени вже зайняті.

Не використовуйте основне доменне ім’я, що схоже на домен конкурента

Користувачі набирають ту адреса, яка краще запам'ятовується, і можуть потрапити на сайт конкурента. Виключенням можуть бути випадки,

коли назви компаній конкурентів є схожими. Тоді, слід подумати про використання в назві основного домену ключового слова.

Реєструйте домени, що звільняються

Багато доменів в пошукових системах вже мають певний авторитет і тому, покупка такого домену, від якого з певних причин відмовився колишній власник, може бути корисною для просування вашого основного сайту.

Реєстрація домену

Перевірити чи є вільним обраний домен можна на сайтах:

✓ whois.com.ua (для перевірки доменів в українських обласних зонах, в зоні .ua, .com.ua, та інших українських доменних зонах).

✓ www.networksolutions.com або www.register.com (для перевірки в міжнародних зонах).

Скористатися перевіркою на сайті любого українського хостинг-провайдера.

Підтримку більшості доменів необхідно оплачувати щорічно, вартість реєстрації та підтримки коливається у межах:

✓ Від 11 у.о. до 18 у.о. (українські організаційні і обласні домени).

✓ Від 80 (у зоні ,ua).

✓ Від 14 у.о. (міжнародні домени).

За рік (сплачений термін обчислюється з моменту реєстрації домену) необхідно оплатити підтримку наступного року, у разі несплати домен відключається і може бути зареєстрований іншим користувачем.

Для реєстрації домену можна звернутися до одного з українських реєстраторів (http://www.hostmaster.net ua/reqistrators/). або сплатити реєстрацію кредитною карткою на www.networksolutions. coma6owww.register.com (для міжнародних доменів).

Вебхостинг. Розміщення сайту в Інтернеті

Вебхостинг (host - власник готелю) - це фізичне розміщення веб-сторінок на сервері. Від того, де буде розміщено сайт, залежить багато якісних характеристик, тому важливо вибрати оптимальний майданчик для сайту, що відповідає критеріям надійності та стабільності.

Види хостингу

Хостинг - це віртуальний аналог оренди приміщення, але орендується місце на диску, яке обчислюється мегабайтами. Хостинг умовно можна поділити на платний і безкоштовний.

Безкоштовний хостинг

Безкоштовний хостинг передбачає надання хостинг-провайдером безкоштовного дискового простору для розміщення сайту в Інтернеті. Безкоштовний хостинг, зазвичай, існує за рахунок реклами, що розміщується на сторінках сайтів. Ця реклама може бути у вигляді банерів, текстових

посилань, рекламних фреймів, спливаючих вікон, хоча існують безкоштовні хостинги, які не розміщують на сайтах жодної реклами.

Поважні компанії, зазвичай, не користуються послугами безкоштовного хостингу, бо він має особливості, що є неприйнятними для серйозного Інтернет проекту.

Основні недоліки безкоштовного хостингу:

✓ Невеликий об'єм, що надається для сайту.

✓ Низька надійність і стабільність серверного майданчика.

✓ Повільне завантаження сайтів.

✓ Присутність реклами.

✓ Часто відсутня підтримка PHP, баз даних та інших даних, що необхідні для повноцінного функціонування сайту.

✓ Відсутність гарантій якісного та постійного надання послуг.

Плюси безкоштовного хостингу:

Хостинг є досить привабливим для малобюджетних, любительських чи тимчасових сайтів. Великим попитом користується серед юних розробників-початківців чи щойно створених спільнот.

Зрештою, інших плюсів безкоштовний хостинг не має, тому, якщо сайт скеровано на довге і стабільне існування, варто задуматися про надійний і швидкий комерційний хостинг.

Залежно від країни розташування, хостинг може бути, наприклад: українським (технічний майданчик оозташовано в Україні), російським (в Росії), американським (у США) тощо.

Платний хостинг

У платному хостингу, власник сайту оплачує певну суму за використання дискового простору та сервіси, що йому надаються.

Хостинг можна класифікувати залежно від:

✓ Програмного забезпечення, що встановлено на сервері.

✓ За типом вмісту об'єктів, що розміщуються на серверах хостинг провайдеру

✓ Типу сервера, на якому розміщуються файли сайтів або програми

✓ Місця розташування технічного майданчика.

За типом програмного забезпечення

Залежно від встановленого на сервері програмного забезпечення хости: с mon-з _ NDOWS-хостинг.

RedHat LINUX (UNIX-платформа) - традиційна платформа веб - хостингу. Це дуже надійна і міцна платформа для підтримки сайтів з великим об'ємом інформації і великою кількістю звернень (трафіком).

Windows NT (WINDOWS-платформа) забезпечує певні опції, які не доступні на платформі хостингу Unix, такі як ASP, Microsoft SQL 2000, Access databases і Cold Fusion 5. Тому, якщо для сайту потрібне одне або

кілька з цих програмних застосувань, варто обирати WINDOWS-платформу.

За типом об'єктів:

- ✓ Веб-хостинг. На серверах розміщують файли сайту.
- ✓ Application-хостинг. На серверах розміщують програми.

За типом сервера:

- ✓ Віртуальний сервер
- ✓ Віртуальний виділений сервер
- ✓ Виділений сервер
- ✓ Колокація

Віртуальний сервер

Віртуальний сервер передбачає розміщення на одному фізичному сервері сайтів та відповідних веб- програм кількох власників. Для кожного ресурсу відводиться місце на твердому диску. Ці ресурси спільно використовують процесорний час і пам'ять сервера. Всі проекти знаходяться на одному комп'ютері-сервері, кожен в своєму каталозі.

Кожний проект обмежено певною дисковою квотою, в залежності від вибраного тарифного плану. Керують узгодженою роботою всіх проектів системні адміністратори хостинг-провайдера.

Віртуальний виділений сервер

Віртуальний виділений сервер також передбачає розміщення Інтернет ресурсів на одному фізичному сервері, але за допомогою спеціальних програм ресурси фізичного сервера розбиваються на кілька віртуальних виділених серверів. Кожен віртуальний виділений сервер є незалежним від інших проектів в сенсі використання процесорного часу і пам'яті, він має свій певний ліміт, що відведено йому програмою- сервер.

Відмінності від віртуального сервера:

✓ Користувачу надається повне управління сервером, доступ до ядра операційної системи. У віртуальному сервері доступ є лише до певних програм-серверів за вибором провайдера (www-, mail-, DNS-, ftp-серверів);

✓ Власнику сайту надаються фіксовані ресурси сервера: пам'ять, процесорний час. У віртуальному сервері фіксованим є лише дисковий простір.

Виділений сервер

Виділений сервер - це оренда фізичного сервера у хостинг-провайдера власником великого Інтернет ресурсу. Сервер знаходиться на технічному майданчику хостинг-провайдера, є постійно під єднаним до безперебійної мережі електроживлення і високошвидкісних каналів передачі інформації.

Виділений сервер відрізняється від віртуального хостингу (віртуального виділеного сервера) тим, що ніхто окрім орендаря не має доступу до сервера, навіть хостинг-провайдер.

Технічний персонал Дата-центра стежить лише за зовнішніми показниками сервера: нормальною роботою складових частин сервера, електроживленням тощо.

Відомості про встановлене на сервері програмне забезпечення та про розташовані там Інтернет ресурси, відомі лише орендареві і більше нікому.

Колокація

Колокація (Co-location - сумісне розміщення) передбачає розміщення власного сервера клієнта на технічному майданчику хостинг-провайдера. Для сервера виділяється місце в сітці, він під'єднується до високошвидкісних каналів передачі інформації і до мережі надійного і безперебійного електроживлення. Відмінністю колокації від виділеного сервера є в тому, що технічний персонал Дата-центра стежить лише за електроживленням сервера і за каналами зв'язку, до яких він під'єднаний. Якщо певна складова частина сервера виходить з ладу, несправність усувається засобами клієнта.

Інші послуги хостинг-провайдера

Окрім послуг з розміщення сайтів, хост-компанії надають і супутні послуги. Це - реєстрація доменних імен, паркування доменів, реселінг.

Реєстрація доменних імен

Реєстрація доменних імен - це отримання офіційної адреси для сайту. В результаті реєстрації сайт отримує своє ім'я і доменну адресу, за якою можна доступитися до сайту.

Реєстрацію доменних імен проводять спеціалізовані організації - реєстратори. Провайдери можуть лише допомогти в реєстрації, бути посередником між власником сайту і реєстратором.

Доменне ім'я - це унікальний набір символів, який дозволяє асоціювати ресурс, що працює в мережі Інтернет, з сервером, зокрема, з його IP-адресою, на якому він розташований. Доменне ім'я - це унікальна адреса, за якою користувач може знайти любий ресурс в мережі інтернет.

Адресацією ресурсів в Інтернеті займається Служба Доменних Імен (Domain Name Service). За загальну координацію і управління мережною службою імен (DNS) і особливо за делегування доменів верхнього рівня, відповідає організація Internet Assigned Numbers Authority (IANA).

Паркування домену

Якщо власник майбутнього сайту хоче зарезервувати певне доменне ім'я, але роботи зі створення сайту тривають або ще не розпочалися, можна скористатися послугою паркування домену. Користувачеві надаються права на дане ім'я, повноцінна поштова адреса, але блокуються повноцінні можливості хостингу, такі як FTP-доступ, панель управління, субдомени, CGI, SSH, SSL, MYSQL. Згодом, коли сайт буде готовий, власник може обрати хостинг, що відповідає його потребам.

Перевагою паркування домену є лише дешевизна послуги у порівнянні з традиційним віртуальним хостингом.

Основні вимоги до майбутнього хостингу

Вибір хост-провайдера є основою успішного представництва в мережі. Тому, варто знати, які вимоги слід висувати до провайдера хостингу.

1. Багато про провайдера свідчить його сайт - візитна картка компанії. Хост-провайдери мають надавати на своїх сайтах повну і однозначну інформацію про себе і послуги, щоб у потенційного клієнта не виникало додаткових питань. Якщо на сайті немає інформації з основних питань - це показник недбалого відношення до послуг, що надаються.

2. Хостинг компанії мають бути професіоналами не лише в наданні послуг, але і у відносинах з клієнтами. Тому, слід перевірити чи існує служба технічної підтримки і чи задовольняє вона всім можливим вимогам.

3. Потужні хостинг-компанії надають своїм клієнтам можливість тестування зручності і якості наявних сервісів за допомогою демо-входу в панель управління аккаунтом (особистої сторінки користувача). Слід перевірити швидкість доступу до сайтів, які розміщені у даного провайдера в різні години доби і дні тижня, щоб переконатися в якості запропонованих послуг. Слід дізнатися про процедуру повернення грошей, якщо якість послуг провайдера виявиться незадовільною.

4. Чи надає хостинг у розпорядження піддомени? В середньому, середньостатистичний хост- провайдер надає близько 6 піддоменів у розпорядження користувача, не стягуючи додаткової плати. Слід з'ясувати умови користування піддоменами. Піддомени можуть виявитися корисним доповненням до сайту користувача. Наприклад, якщо на сайті відкривається форум, то можна дати йому зручну адресу: http://forum. site.ua/.

5. Перевірити швидкість передачі даних. Бажано відвідати сайти, які розташовані на серверах компанії і прослідкувати час завантаження сторінок. Також, варто дізнатися враження клієнтів і замовників, що користуються послугами хостингу вданій компанії.

6. Дізнатися про систему захисту серверів, які використовує компанія.

7. Якщо користувач має свій домен, провайдер зобов'язаний надати у його розпорядження необмежену кількість поштових імен в межах його домену (name@site.ua).

8. Більшість хост-провайдерів надають FTP-доступ та доступ до бази даних. Обов'язково з'ясувати у провайдера, чи надає він наступні сервіси: CGI-сервіси, підтримку Perl, C, PHP MYSQL? Чи не стягується за використання цих послуг додаткова плата?

9. Вартість хостингу також грає важливу роль. Наприклад, ціна на оренду виділеного сервера може досягати порядку 200 $ у місяць, тоді як звичайний хостинг для сайту в місяць може складати всього 2-3 $.

Основні критерії вибору хостингу:

- ✓ Вартість хостингу
- ✓ Сервісні послуги
- ✓ Параметри хостингу
- ✓ Якість послуг, що надаються

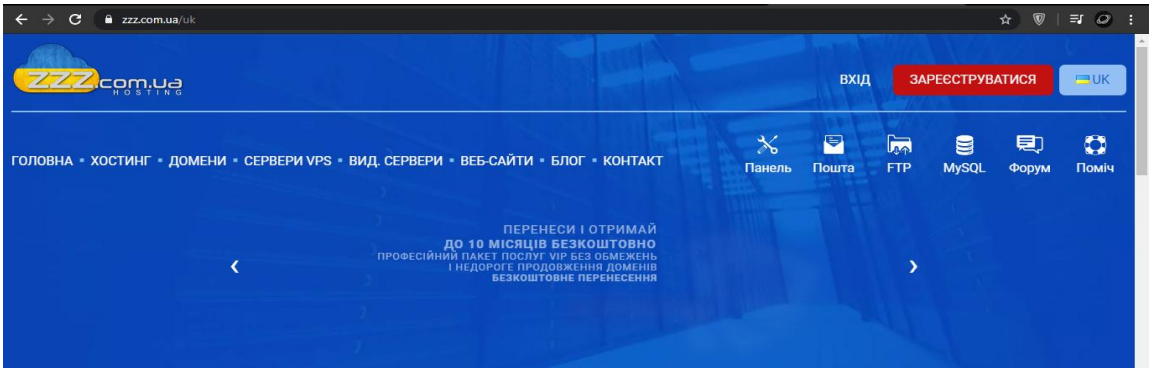
Очевидно, що ці критерії є важливими при виборі хостингу: співвідношення ціни, наявність сервісних послуг зумовлюють вибір користувача на користь того або іншого хостинг-провайдер. а значить, і надійну роботу сайту, і його окупність.

Отже, хостинг - це не лише оренда місця на твердому диску сервера і під'єднання до швидкісних каналів інформації. Сучасні сайти мають різний ступінь складності: від домашніх сторінок до великих порталів. Для їх нормального відображення і безперебійної роботи необхідно мати цілий ряд додаткових функцій, які повинен виконувати хост-сервер. Ці функції реалізовано в багатьох хостинг-провайдерів шляхом встановлення додаткових допоміжних програм на серверах.

Дізнатися про хостинг-провайдерів просто, для цього достатньо задати в пошуковій системі запит типу «хостинг» або «купити хостинг», або «хостинг для сайту» і зазвичай, на перші позиції буде виведено сайти відомих, потужних та надійних хостинг-провайдерів.

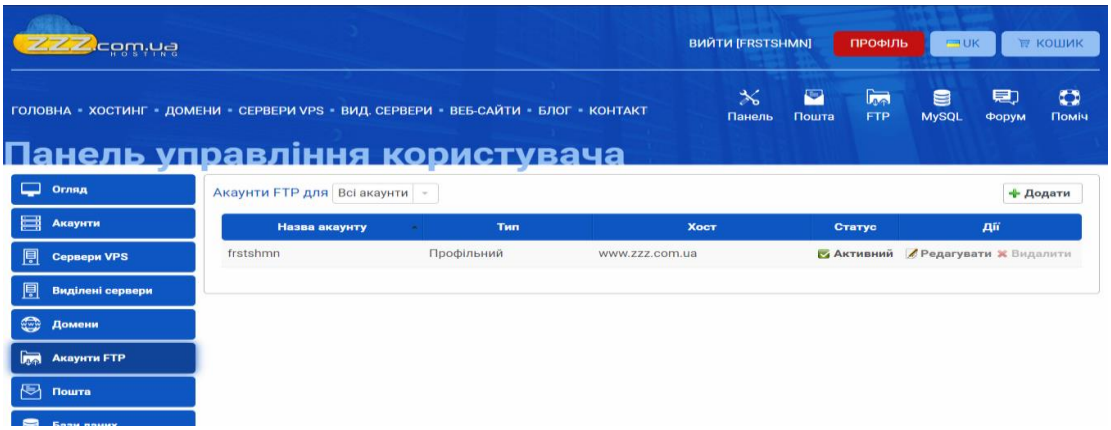
Завдання для самостійної роботи

1. Розглянути роботу по управлінню хостингом на прикладі сервісу **www.zzz.com.ua**. Вам потрібно провести реєстрацію на сайті та перейти на вкладку Домены, де створити безкоштовне доменне ім'я. Для управління хостингом перейти на вкладку Панель, де ви можете побачити створене нове доменне ім'я.



2. Натиснути на кнопку Огляд і переглянути всі можливі служби для роботи з вашим хостингом, а саме служби для роботи з файлами, базами даних, електронною поштою, доменними іменами, онлайн-конструктором сайтів тощо (рисунки 2-3). Розглянути деякі з найбільш важливих служб:

1) **FTP-доступ**



Після створення акаунту автоматично створюється один акаунт FTP, його назва має формат: admin@adresa (наприклад, admin@happyuser.zzz.com.ua). Це водночас логін до акаунту FTP. Пароль до цього акаунту за замовчуванням такий же, як пароль до панелі керування ZZZ.com.ua.

В профільному акаунті FTP є глобальний доступ до всіх каталогів зі всіх хостингових акаунтів у вашому профілі. Акаунтами FTP можна керувати в панелі керування ZZZ.com.ua в закладці «Акаунти FTP». Там є можливість їх додавати, видаляти і змінювати паролі до них.

На сервер FTP можна потрапити за допомогою нашого файлового менеджера або за допомогою зовнішньої програми, наприклад, FileZilla.

Ви також можете додати акаунт FTP з доступом виключно у межах конкретного каталогу домену — в закладці «Акаунти FTP» у процесі додавання нового акаунту можна вибрати зі списку домен, для котрого створюється акаунт зі вказаним логіном. Логін до акаунту FTP тоді матиме такий вигляд: login@adresastorinky.zzz.com.ua.

Дані для доступу до FTP:

Сервер FTP: адреса вашого сайту (наприклад, happyuser.zzz.com.ua)

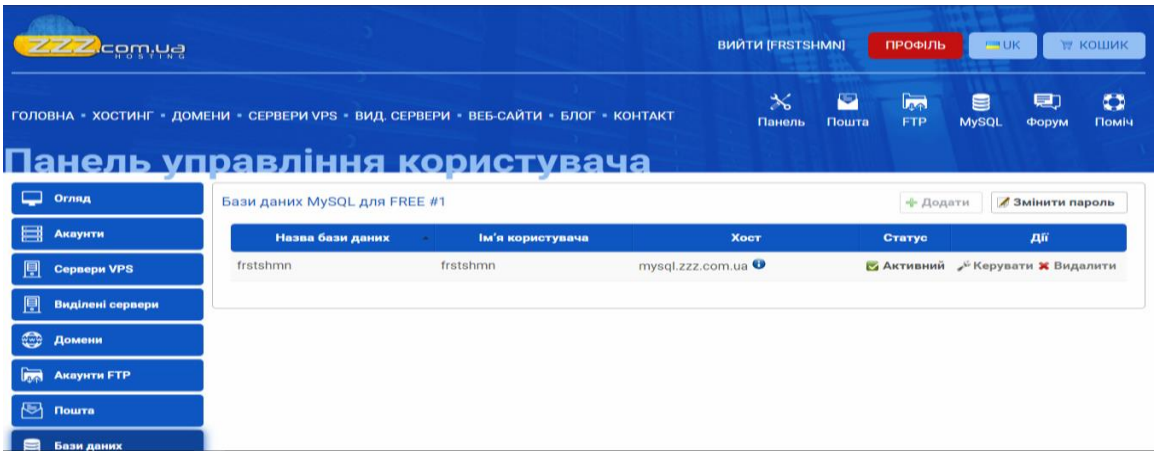
Логін: за замовчуванням admin@adresa (наприклад, admin@happyuser.zzz.com.ua)

Пароль: за замовчуванням такий же, як до панелі керування ZZZ.com.ua (якщо ви його не змінювали)

Порт: 21 або 210

Режим: пасивний

2) **Бази даних**



Базу даних можна створити в панелі керування ZZZ.com.ua в закладці «Бази даних». Потрібно тільки клікнути «+ Додати» і встановити ім'я користувача (логін) і пароль. Ім'я бази даних буде створено автоматично на основі адреси вашого сайту (спеціальні символи будуть замінені на символ «_»). Наприклад, для сайту happyuser.zzz.com.ua база даних матиме такі параметри:

- **Сервер MySQL**: mysql.zzz.com.ua;
- **Сервер MySQL для підключення ззовні**: Ваш домен (наприклад, happyuser.zzz.com.ua). *Увага: підключення ззовні доступні тільки для платних акаунтів;*
- **Порт**: 3306 (phpBB By Przemo: 80);
- **Логін, пароль**: такі, які були встановлені при створенні бази;
- **Назва бази даних**: буде створена автоматично на основі адреси сайту, наприклад, happyuser_zzz_com_ua.

Звіт до практичного заняття №28

1. Роздрукувати титульну сторінку практичної роботи, результати виконання завдання: зазначення і обґрунтування вибору доменної адреси і хост-майданчика. Print screen адміністративної панелі і зазначення можливостей, які надані для сайту.

2. Підготувати відповіді на контрольні запитання в електронному і роздрукованому вигляді:

- Які фактори потрібно враховувати при виборі доменної адреси?
- Як можна зареєструвати доменну адресу для сайту?
- Що таке хостинг?
- Які бувають умови надання хостингу?
- Які типи хостингів існують? Якій вибір буде доцільним для вашого сайту?
- У чому відмінності варіантів повнофункціонального хостингу?
- Що входить до послуг обмеженого хостингу? Недоліки.
- Охарактеризуйте найбільш відомі безкоштовні хостинги?
- Які основні служби забезпечує хостинг?
- Що таке FTP?
- Які переваги має FTP-клієнт у порівнянні з можливостями адмінок?
- Які інструментальні засоби забезпечують альтернативні можливості створення FTP-клієнта?
- Які параметри потрібні для налаштування FTP-з'єднання?
- Основні можливості програми Krusader та її відмінності від інших FTP-менеджерів?

ЛІТЕРАТУРА

1. Глибовець М. М., Завадський І. О. Вступ до ведтехнологій: посіб. для вищих навчальних закладів. – К. : «ІТ-книга», 2023. – 160с.
2. Бородкіна І.Л., Бородкін Г.О. Web-технології та Web-дизайн: застосування мови HTML для створення електронних ресурсів. К.: Ліра, 2020. 212 с.
3. Трофименко О.Г., Козін О.Б., Задерейко О.В., Плачінда О.Є. Веб-технології та веб-дизайн: навчальний посібник. Одеса: Фенікс, 2019. 284 с.
4. Пасічник О. В., Пасічник В. В. Веб-дизайн: Підручник. – Львів: Магнолія 2006, 2017. – 570 с.
5. Пасічник О. В., Пасічник В. В., Угрин Д. І. Веб-технології: Підручник. – Львів: Магнолія 2006, 2018. – 336 с.
6. Електронний навчальний посібник з дисципліни «Web-технологій та web-дизайн», укладач Людмила Мелещук, 2019р.
7. Конспект лекцій з дисципліни «Web-технології та Web-дизайн» для здобувачів освітньо-кваліфікаційного рівня фаховий молодший бакалавр IV курсу спеціальності 122 Комп’ютерні науки денної форми навчання / уклад. Л. Мелещук – Ковель: КПЕФК ЛНТУ, 2022. – 92 с.
8. Методичні вказівки до практичних занять з дисципліни «Web-технологій та web-дизайн» для здобувачів освітньо-кваліфікаційного рівня молодший спеціаліст IV курсу спеціальності 122 Комп’ютерні науки денної форми навчання / уклад. Л. Мелещук – Ковель: КПЕФК ЛНТУ, 2023. – 92 с.

Допоміжна

1. Зубик Л.В., Карпович І.М., Степанченко О.М. Основи сучасних web-технологій: навчальний посібник. Рівне: НУВГП, 2020. 290 с.
2. Для вивчення курсу використовується програмне забезпечення, яке встановлене на робочих станціях в ауд. №17 та №18 КПЕФК ЛНТУ: Chrome, Brainycp, phpMyAdmin.

Інтернет-ресурси:

1. www.nbuv.gov.ua Національна бібліотека України ім. І.І. Вернадського.
   www.britannica.com Он-лайн енциклопедія «Британіка».

Web-технології та Web-дизайн [Текст]: Методичні вказівки до практичних занять для здобувачів освітньо-професійного ступеня фаховий молодший бакалавр IV курсу спеціальності 122 Комп'ютерні науки денної форми здобуття освіти / уклад. Л.В. МЕЛЕЩУК. – Ковель : ВСП «КПЕФК ЛНТУ», 2025. – 154

Комп'ютерний набір і верстка: Людмила МЕЛЕЩУК

Редактор: Людмила МЕЛЕЩУК

Підп. до друку _____2025. Формат 60х84/16.Папір офс. Гарн. Таймс.
Ум. друк. арк.3,4.
Обл.-вид. арк. 7,4. Тираж _____прим. Зам.188.

ВСП «Ковельський промислово-економічний фаховий коледж
Луцького національного технічного університету»
45000 м. Ковель, вул. Заводська, 23
Друк – ВСП «КПЕФК ЛНТУ»