

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет
науки і технологій**

Кафедра «Автоматизації виробничих процесів»

В авторській редакції

PHR-ПРОГРАМУВАННЯ

Навчально-методичні рекомендації
до виконання індивідуального завдання

Електронне видання



ДНПРО
2025

Упорядники:
М. О. Рибальченко, І. О. Маначин, А. А. Бурчак

Електронне видання

Схвалено Групою забезпечення якості освітньої програми
«Комп'ютеризовані системи управління та робототехніка»
Протокол № 6 від 07.01.2025 р.

Р 56 РНР-програмування : навчально-методичні рекомендації до виконання індивідуального завдання / упоряд. М. О. Рибальченко, І. О. Маначин, А. А. Бурчак ; Укр. держ. ун-т науки і технологій. – Електрон. вид. – Дніпро : УДУНТ, 2025. – 47 с.

Навчально-методичні рекомендації містять опис та методику виконання індивідуального завдання, що полягає у виконанні блокової HTML/CSS верстки шаблону дизайну для вебсайту. Наведені вихідні дані та рекомендації щодо його виконання, які підкріплені відповідним прикладом.

Призначаються для студентів другого (магістерського) рівня вищої освіти, які опановують вибіркову навчальну дисципліну «РНР-програмування» на освітній програмі «Комп'ютеризовані системи управління та робототехніка» за спеціальністю 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка».

Іл. 20, бібліогр. 2 назв.

ЗМІСТ

ПЕРЕДМОВА	4
1. КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ.....	5
1.1. Основи HTML	5
1.2. Основи каскадних таблиць стилів CSS	11
1.3. Основи верстки сайтів.....	24
2. ІНДИВІДУАЛЬНІ ЗАВДАННЯ.....	32
3. РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ.....	32
СПИСОК ЛІТЕРАТУРИ.....	46

ПЕРЕДМОВА

Навчальна дисципліна «PHP програмування», що вивчається в рамках освітньо-професійної програми «Комп'ютеризовані системи управління та робототехніка» другого (магістерського) рівня вищої освіти за спеціальністю 174 – «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» спрямована на формування теоретичних та практичних знань сучасних інструментів, мов програмування, технологій та підходів до розробки вебдодатків. Засвоєння можливостей використання HTML/CSS, PHP, MySQL для програмування динамічних вебсайтів і вебінтерфейсів доступу до баз даних.

При вивченні дисципліни «PHP програмування» студенти поглиблюють та розвивають таку передбачену освітньою програмою компетентність:

- здатність застосовувати спеціалізоване програмне забезпечення та цифрові технології для розв'язання складних задач і проблем автоматизації та комп'ютерно-інтегрованих технологій.

Відповідно до навчального плану спеціальності 174 – «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» лекційний матеріал програми дисципліни «PHP програмування» повинен бути закріплений студентами при виконанні індивідуального завдання. Це індивідуальне завдання передбачає виконання блокової HTML/CSS верстки шаблону дизайну для вебсайту. Розроблені навчально-методичні рекомендації спрямовані на розгляд основних теоретичних відомостей, пояснень та порад щодо виконання цього завдання.

1.КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1. Основи HTML

Мова HTML (від англ. *HyperText Markup Language* – "мова розмітки гіпер-тексту") служить для створення веб-сторінок. Більшість сайтів створено саме за допомогою HTML .

HTML-документи являють собою файли з текстом та додатковими інструкціями мови HTML, які називаються тегами. Теги дозволяють задавати форматування тексту, а також розміщувати в документі мультимедійні файли, гіпертекстові посилання інші документи, табличні дані, форми введення даних. HTML-документи мають розширення імені файлу html . Редагування HTML коду виконують у текстовому редакторі, а перегляд – у браузері.

Структура тега:

```
<ім'я тега атрибут1 атрибут2="значення2" ...>
```

Тег складається з імені тега, за яким може слідувати список атрибутів, що поміщаються між кутовими дужками (< >). Атрибути дозволяють керувати поведінкою тега. Вони можуть мати конкретні значення, які задаються після знаку рівності. Значення атрибутів поміщують в одиначні (') або подвійні лапки ("). Атрибути відокремлюються один від одного пробілом, порядок атрибутів значення не має. Імена тегів і атрибутів нечутливі до регістру.

Приклад: ``

Тег font призначений для керування зовнішнім виглядом тексту. У прикладі він задає зображення тексту шрифтом Arial , червоним кольором.

Теги поділяються на парні та непарні.

Непарні теги впливають на весь документ або визначають разовий ефект у місці своєї появи. Парні теги впливають на частину документа, укладену між відкриваючим і закриваючим (кінцевим) тегами. Закриваючий тег відрізняється від відкриваючого символом «/» перед ключовим словом - наприклад, `</title>`

Прикладом парного тега є тег `...` для жирного виділення тексту.

Прикладом непарного тега є тег переведення рядка `
`. Звичайне переведення рядку клавішею {Enter} браузер ігнорує (як і кілька пробілів, що поставлені поспіль, або знаків табуляції).

При вкладанні тегів один в одного закривати теги потрібно починаючи з останнього, у зворотному порядку.

Документ HTML складається з трьох частин:

1)декларація типу документа (англ. Document type declaration, Doctype), на початку документа, в якій визначається тип документа (DTD);

2)шапка документа (знаходиться в межах елемента head), в якій записано загальні технічні відомості або додаткова інформація про документ, яка не відтворюється безпосередньо в браузері;

3)тіло документа (може знаходитися в елементі body), в якому міститься основна інформація документа.

Нижче наведено приклад структури HTML-документа:

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Документ</title>
  </head>
  <body>
    Це приклад HTML коду.
  </body>
</html>
```

Мова розмітки веб сторінок HTML передбачає такі різновиди представлення кольору:

1.У вигляді рядку, що починається з символу «#» і включає три пари шістнадцяткових цифр, які відповідають відповідно за червоний (перша пара), зелений (друга пара) і синій (третя пара) кольори.

Наприклад:	#FF0000 -	яскраво-червоний (red)
	#00FF00 –	яскраво-зелений (green)
	#0000FF –	яскраво-синій (blue)
	#FFFF00 –	жовтий (yellow)
	#000000 –	чорний (black)
	#FFFFFF –	білий (white)

2.За допомогою ключових слів англійською мовою, наприклад green, black, при цьому слід мати на увазі, що браузери підтримують лише обмежену кількість таких слів, а також слово transparent (прозорий).

3.RGB (R,G,B) чи RGB (R,G,B,*) — перші три параметри це кольори червоний (Red), зелений (Green), синій (Blue), записуються числами від 0 до 255. Можливий четвертий параметр *, це число від 0 до 1, визначає ступінь непрозорості кольору. Наприклад, rgba(0,255,0,0.3); відобразить зелений колір із прозорістю 70 %.

Приклад використання кольору у HTML-документі:

`сірий текст`

Кожен тег призначений на вирішення певного завдання: роботи з текстом, посиланнями, графікою, таблицями тощо. Розглянемо найбільш поширені.

Теги структури документа призначені для визначення структури HTML документа та не впливають на його відображення у браузері. Проте правильно сформований документ обов'язково має їх містити.

`<html>...</html>` – включає весь вміст веб-сторінки, у тому числі теги `<head>` і `<body>`

`<head>...</head>` – містить теги зі службовою інформацією про сторінку, наприклад, назва в тегу `<title>`.

`<title>...</title>` – задає назву документа. Ця назва зазвичай відображається у заголовку вікна браузера.

`<body>...</body>` – зберігає вміст документа.

Атрибути:

`bgcolor = "колір"` – призначає колір тла документа

`text = "колір"` – вказує колір звичайного тексту в документі

Теги для роботи з текстом дозволяють керувати відображенням тексту на сторінці.

`...` – виділення тексту жирним

`<i>...</i>` – виділення тексту курсивом

`<u>...</u>` – підкреслення тексту

`_{...}` – формувати текст як підрядковий індекс

`^{...}` – формувати текст як надрядковий індекс

`<center>...</center>` – вирівнювання тексту по центру

`...` – встановлює розмір, колір та гарнітуру тексту

Атрибути:

`color = "колір"` – задає колір тексту

`face = "шрифт"` – визначає гарнітуру тексту; значенням атрибуту може бути список шрифтів, перерахованих через кому – у цьому випадку вибирається перший доступний шрифт

`size = "1-7"` – встановлює розмір шрифту (від 1 до 7)

`<p>...</p>` – задає початок та кінець параграфа

Атрибут:

`align = "..."` – визначає режим вирівнювання тексту

`left` – по лівому краю (за умовчанням)

`center` – по центру

`right` – по правому краю

`justify` – по ширині

`<hN>...</hN>` – вкладений текст, є заголовком документа рівня N , N

набуває значення від 1 до 6. Найбільшим заголовком є <h1>, найменшим <h6>.

 – перенесення рядка

Тег <hr> – виводить горизонтальну розділову лінію

Атрибути:

align = "..."- визначає режим вирівнювання лінії

left – по лівому краю

center – по центру (за замовчуванням)

right – по правому краю

noshade – використовувати суцільну лінію замість об'ємної

size = "N" - товщина лінії в пікселях

width = "N" – ширина лінії у пікселях або відсотках по відношенню до ширини екрана.

Розміри об'єктів HTML часто вказуються в пікселях. Піксель – найменший елемент зображення на екрані (крапка). Кількість пікселів на екрані по горизонталі та вертикалі називають роздільною здатністю (наприклад, 1024 по горизонталі на 768 по вертикалі).

Робота із списками. У HTML є можливість створювати нумеровані та марковані списки.

... – створює нумерований список елементів

Атрибути:

start = "N" – почати нумерацію з числа N

type = "..."- визначає формат нумерації

1 – арабські цифри (за замовчуванням)

A – великі літери (A , B , C)

a - малі літери (a , b , c)

I – великі римські цифри (I , II , III)

i – малі римські цифри (i , ii , iii)

... – створює маркований список елементів

Атрибут:

type = "..."- визначає формат маркера

disk – диск (за замовчуванням)

circle – коло

square – квадрат

... – задає елемент списку у нумерованому або маркованому списку.

Атрибути:

type = "..."- формат номера або маркера (див. опис < ol > та < ul >)

value = " N " – задає номер елемента списку

Приклад:

HTML -код:

```
<html>
  <head>
    <title> Робота зі списками </title>
  </head>
  <body>
    <ol>
      <li>арабські цифри (за замовчуванням)</li >
      <li type = "A">великі літери</li>
      <li type = "a">малі літери</li>
      <li type = "I">великі римські цифри</li>
      <li type = "i">малі римські цифри</li>
    </ol>
    <ul>
      <li>диск (за замовчуванням)</li>
      <li type = "circle">коло</li>
      <li type = "square">квадрат</li >
    </ul>
  </body>
</html>
```

У браузері (рис.1.1):

1. арабські цифри (за замовчуванням)
 - В. великі літери
 - с. малі літери
 - IV. великі римські цифри
 - v. малі римські цифри
- диск (за замовчуванням)
 - коло
 - квадрат

Рис.1.1. Відображення у браузері

Для створення посилань використовується тег <a>

Обов'язковий атрибут href вказує на абсолютну або відносну адресу, на яку веде посилання.

Абсолютна адреса містить у собі повний шлях до ресурсу. За допомогою абсолютної адреси можна посилатись на будь-який відкритий ресурс в Інтернеті. Якщо потрібно поставити посилання на головну сторінку сайту, вказують його адресу та символ «/».

Також для документів, розміщених на тому ж сайті, можна використовувати відносну адресу. Для файлів у межах одного сайту рекомендується використовувати лише відносні шляхи. У цьому випадку посилання зберігатиме працездатність при зміні адреси сайту, перенесенні в іншу папку тощо.

Для відкриття посилання у новому вікні використовується атрибут target зі значенням _blank .

Приклад:

```
<a href="http://google.com/" target="_blank"> Google </a>
```

Колір посилань у документі можна вказати атрибутами тега <a> :

alink = "колір" – встановлює колір активних посилань

link = "колір" – задає колір невідвіданих посилань

vlink = "колір" – визначає колір відвіданих посилань

Вставка зображень на сторінці здійснюється непарним тегом .
Обов'язковий атрибут src вказує на абсолютний або відносний URL зображення .

Стандартними форматами зображень є GIF , PNG та JPEG .

Атрибути тега :

align = "... " – визначає режим вирівнювання зображення щодо тексту в рядку:

top – по верхньому краю

middle – по центру рядка

bottom – по нижньому краю (за замовчуванням)

left – по лівому краю вікна

right – по правому краю вікна

alt = "... " – визначає текст, що описує зображення для браузерів без підтримки графіки (або з відключеною графікою), пошукових машин тощо.

border = "N" – встановлює товщину рамки навколо зображень, що дорівнює N пікселів, 0 – для відключення рамки

height = "N" – висота зображення в пікселях або відсотках

width = " N" – ширина зображення в пікселях або відсотках

Браузер визначає розмір зображення автоматично. Для прискорення завантаження рекомендується вказувати розмір зображення атрибутами height і width , щоб браузер не обчислював цей розмір автоматично після завантаження зображення. Також цими атрибутами можна розтягнути/стиснути зображення по горизонталі/вертикалі, але таке масштабування призведе до втрати якості.

Зображення може бути зроблено посиланням, поміщаючи всередину тег <a>. У цьому випадку навколо зображення автоматично з'являється рамка.

Товщина рамки визначається атрибутом border. Зазвичай рамку забирають, вказуючи border = "0" у тегу .

Приклади:

1. HTML -сторінка знаходиться в папці site , а зображення picture.jpg знаходиться в папці site /images/.

```

```

2. Зображення знаходиться на іншому сайті в Інтернеті

```
<img src = "http://example.com/pics/tree.gif" alt =  
"дерево">
```

При використанні зображень як фон важливо забезпечити читабельність тексту на сторінці.

1.2. Основи каскадних таблиць стилів CSS

CSS (Cascading Style Sheets – каскадні таблиці стилів) – технологія управління зовнішнім виглядом елементів (тегів) веб-сторінки. CSS надає набагато більше можливостей для оформлення сторінки, ніж HTML .

Розглянемо синтаксис CSS. У стилях задається набір правил відображення в парах «властивість – значення», і те, до яких елементів їх застосовувати (селектор):

```
Селектор  
{  
властивість1: значення1;  
властивість2: значення2;  
властивість3: значення3 значення4;  
}
```

Правила записуються всередині фігурних дужок і відокремлюються один від одного крапкою з комою. Між властивостями та їх значеннями ставиться двокрапка.

Можна додавати коментарі, укладаючи їх між /* та */.

Селектор визначає, до яких елементів (тегів) сторінки застосовуватимуться правила, задані парами «властивість – значення».

Як селектор можна використовувати:

- Назву тега – тоді стиль застосовується до всіх таких тегів.

Приклад:

```
A {font-size: 12pt; text-decoration: none}  
TABLE {border: black solid 1px}
```

Перший рядок цього CSS -коду задає всім посиланням 12-й розмір шрифту і прибирає підкреслення. На другому рядку вказується, що всі лінії таблиці будуть чорного кольору, суцільними (solid) і шириною 1 піксель.

- Декілька тегів через кому - тоді стиль застосовується для всіх перерахованих тегів.

Приклад:

```
/* робимо всі заголовки червоними */
H 1, H 2, H 3, H 4, H 5, H 6 { color : red }
```

- Декілька тегів через пробіл:

```
TABLE A { font - size : 120%}
```

Правило відноситься до всіх тегів A , вкладених у тег TABLE . Розмір шрифту збільшиться на 20% від базового.

- ID елемент.

У стилях унікальний ідентифікатор вказується після знака # – правила застосовуються до тега з атрибутом id = "ідентифікатор".

Приклад:

CSS

```
#supersize {font-size: 200%}
```

HTML

```
<a href="http://htmlbook.com" id="supersize">Довідник HTML та CSS</a>
```

Не можна вносити до документа кілька елементів з однаковим id ! Символ * – правила застосовуються до всіх елементів документа.

- Класи

Часто потрібно, щоб стиль застосовувався не до всіх тегів на сторінці, а лише до деяких елементів (наприклад, не до всіх посилань на сторінці, а лише до тих, що розташовані в меню сайту). Для цього використовуються класи:

```
ТЕГ.ім'я_класу { ... }
```

Правила, вказані після такого селектора, діятимуть лише на теги з атрибутом class = "ім'я_класу":

```
<ТЕГ class = "ім'я_класу"> ... </ТЕГ>
```

Можна не вказувати ім'я тега, тоді правила будуть застосовуватись до всіх тегів з відповідним значенням атрибута class .

Розглянемо приклад:

для всіх тегів з атрибутом `class = "class1"` додамо підкреслення тексту та зменшимо розмір шрифту, а для тега `<a>` приберемо підкреслення.

```
.class1 {text-decoration: underline; font-size: 80%}  
A .class1 {text-decoration: none;}
```

В HTML кодi вкажемо для тегів ім'я класу:

```
<h1 class ="class1">Мої улюблені сайти</h1> </br>  
<a href="http://google.com" class="class1">Google</a>
```

Стилі CSS можуть включатися в HTML -документ 3 різними способами. Розглянемо ці методи.

1. Зовнішні стилі .

Зберігаються в окремому файлі. `css` . Підключаються тегом

```
<link rel="stylesheet" type="text/css" href="адреса_стилю">
```

Основна перевага: один стиль може використовуватися відразу в кількох документах HTML .

Реалізуємо наш попередній приклад. Створимо файл `style.css` :

```
.class1 {text-decoration: underline; font-size: 80%}  
A.class1 {text-decoration: none;}
```

Тепер створимо саму сторінку `links.html` :

```
<html>  
  <head>  
    <link rel="stylesheet" type="text/css"  
href="style.css">  
  </head>  
  <body>  
    <h1 class =" class 1">Мої улюблені сайти</h1>  
    <a href="http://google.com" class="class1">  
Google</a><br>  
  </body >  
</html>
```

При відкритті цієї сторінки браузер клієнта також завантажить файл `style.css` та застосує правила CSS до документа.

Використання CSS дозволяє розділити оформлення та вміст документа.

У нашому прикладі правила оформлення містяться у файлі style.css , а зміст – у links.html . Такий поділ суттєво спрощує редагування сайту надалі.

У браузері відобразатиметься (рис.1.2):

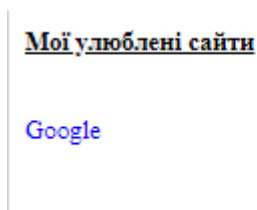


Рис.1.2. Відображення сторінки у браузері

2. Стили рівня документа.

Застосовуються до всього документа, записуються всередині тега <style>...</style>, який вкладається в тег <head>...</head> у документі HTML. Такий спосіб вказівки стилів застосовується, коли потрібно застосувати однакові стилі відразу до декількох HTML-елементів (тегів) в одному документі.

Додамо до нашого прикладу тег <style>:

```
<!DOCTYPE html>
<html>
<head>
  <title>Моя веб-сторінка</title>
  <style>
    .class1 {text-decoration: underline; font-size: 80%;}
    a.class1 {text-decoration: none;}
  </style>
</head>
<body>
  <h1 class="class1">Мои любимые сайты</h1> </br>
  <a href="http://google.com" class="class1">Google</a>
</body>
</html>
```

3. Внутрішні стилі

Використовуються, коли потрібно вказати стилі конкретного елемента. Внутрішній стиль записується в атрибуті тега **style** і застосовується лише до цього тега. Внутрішній стиль має більший пріоритет, ніж зовнішні стилі та стиль рівня документа.

Наприклад:

```
<a href="http://ust.edu.ua" style="color: green">УДУНТ</a>
```

При роботі з CSS необхідно пам'ятати, що більш специфічні правила мають пріоритет над менш специфічними, наприклад: стиль, вказаний в атрибуті style, перекриває стиль, вказаний у тегу <style> або зовнішньому файлі CSS .

Селектор ID (#) має більший пріоритет, ніж селектор класу (.), а той у свою чергу - більший, ніж звичайний селектор тега.

Наведемо приклад.

```
<!DOCTYPE html>
<html>
<head>
  <title>Моя веб-сторінка</title>
  <style>
    A{color: red; text-decoration: none; font-size: 120%;}
    .links {color: blue; text-decoration: underline;}
    #greenlink {color: green;}
  </style>
</head>
<body>
  <h1 class = "class1">Мої улюблені сайти</h1></br>
  <a href="http://google.com" class="class1">Google</a></br>
  <a href="http://ust.edu.ua" style="color: green">УДУНТ</a>
</body>
</html>
```

У браузері посилання буде зеленим і підкресленим, розмір шрифту збільшений на 20% (рис.1.3).

Мої улюблені сайти

Google
УДУНТ

Рис.1.3. Відображення веб-сторінки

Розміри CSS можна задавати в різних одиницях вимірювання, наприклад:

- em – поточна висота шрифту;
- pt – пункти (типографська одиниця виміру шрифту);
- px – піксель;
- % – відсоток;

Одиниця виміру записується відразу за значенням без пробілу:

```
TABLE {font-size: 12pt;}
```

У CSS колір задається як і в HTML - 6 шістнадцятковими цифрами: по 2 на кожен базовий колір (червоний, зелений, синій). Також можна використовувати стандартні назви кольорів англійською.

Наприклад:

```
.content {color: black;}  
#menu {color: #3300AA;}
```

Рідше використовується конструкція rgb (...), яка дозволяє задавати червону, зелену та синю компоненти в десятковому або відсотковому вигляді:

```
.content {color: rgb (0%,0%,0%)}
```

Посилання задаються конструкцією URL (...). Наприклад, наступний CSS -код додає фонове зображення для сторінки:

```
body {background-image: url(images/bg.jpg);}
```

Шрифт – набір букв та знаків. У комп'ютері шрифт являє собою файл, в якому описано, як мають відобразитися на моніторі або принтері різні символи: букви, цифри, знаки пунктуації та ін. Існують Unicode - шрифти, які містять символи всіх мов. Основні формати файлів шрифтів: TTF – TrueType та його розширення OTF – OpenType .

Типи шрифтів:

- serif – шрифти із засічками (антиквенні), наприклад: Times New Roman , GEORGIA .
- sans-serif – рубані шрифти (шрифти без засічок або гротески), типові представники - Arial , IMPACT , Tahoma , VERDANA ;
- cursive – курсивні шрифти: COMIC SANS MS ;
- fantasy – декоративні шрифти, наприклад: CURLZ MT .
- monospace – моноширинні шрифти, ширина кожного символу однакова.

Приклади: Courier New , LUCIDA CONSOLE .

Засічками називають елементи на кінцях штрихів літер. Порівняємо букву шрифту Times New Roman і літеру шрифту Arial (рис.1.4) .



Рис.1.4. Порівняння літери М антиквенного та рубаного шрифту (пунктирними лініями обведені засічки)

Використання шрифтів із засічками полегшує читання тексту з паперу, тому такі шрифти зазвичай використовують для набору основного тексту в книгах. Для веб-сайтів основний текст частіше набирають шрифтом без засічок.

CSS дозволяє керувати властивостями шрифту та тексту.

`font - family` - задає шрифт. Можна вказати кілька значень через кому. Браузер перевірить перший шрифт зі списку: якщо шрифт встановлений на комп'ютері користувача, браузер застосує його, якщо ні - перейде до другого шрифту і т.д. Останнім у списку зазвичай вказується загальний тип шрифту `serif`, `sans-serif`, `cursive`, `fantasy` або `monospace`. Приклад:

```
font - family : Georgia , 'Times New Roman', serif;
```

Якщо на комп'ютері користувача встановлено шрифт `Georgia`, то використовуватиметься він, якщо ні – `Times New Roman`. Якщо ж і `Times New Roman` виявиться відсутнім, то браузер буде використовувати шрифт із засічками, який встановлений на комп'ютері.

`font - size` - розмір шрифту. Може задаватися абсолютним значенням у пунктах (pt) чи пікселях (px); або відносним – у відсотках (%) чи `em`. Приклад:

```
font-size : 12 pt;
```

`font - style` - задає зображення тексту:

- `normal` (звичайне),
- `italic` (курсивне).

`font - weight` – дозволяє змінити рівень жирності тексту:

- `normal` (звичайний),
- `bold` (напівжирний).

`color` - задає колір тексту.

Наприклад, задаємо червоний колір для всіх заголовків:

```
h1, h2, h3, h4, h5, h6 {color: #ff0000;}
```

або

```
h1, h2, h3, h4, h5, h6 {color: red;}
```

`line - height` - міжрядковий інтервал, вказує відстань між рядками

тексту. Може задаватися числом, як множник від поточного розміру шрифту, у відсотках, а також у пунктах (pt), пікселях (px). Приклад:

```
line-height: 1.5; /* Полуторний інтервал */
```

text - decoration - задає оформлення тексту. Варіанти:

- line - through (перекреслений),
- overline (лінія над текстом),
- underline (підкреслення),
- none (вимкнення ефектів).

Наприклад:

```
a {text - decoration: none;}
```

text - align - вирівнювання тексту:

- left (лівий край),
- center (по центру),
- right (правий край),
- justify (по ширині).

Приклад:

```
p {text-align: justify;}
```

text - indent - відступ першого рядка ("червоний рядок"). Довжина відступу може задаватися у відсотках (%) від ширини текстового блоку, пікселях (px), пунктах (pt) та ін.

Приклад :

```
p {text-indent: 1.25cm}
```

У CSS кожен елемент розташовується в блоці, якому можна задати значення полів (margin), заповнення (padding) та межі (border). Поле є відступом елемента від сусідніх, а заповнення – порожньою областю між кордоном та вмістом (див. рис. 1.5).

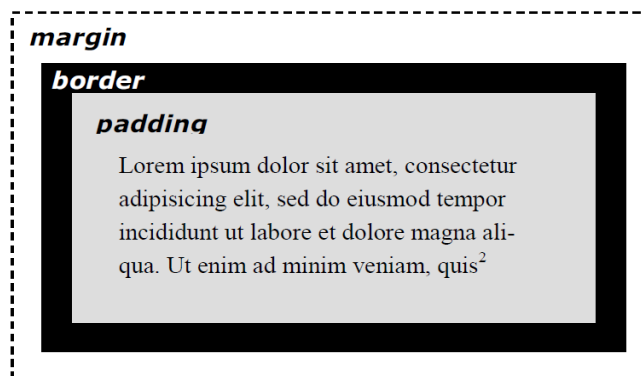


Рис.1.5.Бокс (box) елемента

Ширина полів та заповнення задається наступними CSS властивостями:

margin-top, margin-right, margin-bottom, margin-left - для верхньої, правої, нижньої, лівої сторони поля.

margin - скорочений запис. Задає значення відразу всім сторін.
Приклад:

```
p { margin : 10 px;}
```

аналогічно до запису

```
p {  
margin-top: 10px;  
margin-right: 10px;  
margin-bottom: 10px;  
margin-left: 10px;  
}
```

padding - top, padding - right, padding - bottom, padding - left - встановлюють ширину заповнення зверху, праворуч, знизу та зліва від вмісту відповідно.

padding - встановлює значення відразу всім сторін.

Для margin і padding можна задавати значення auto. У цьому випадку браузер автоматично розрахує величину полів і заповнення.

Для кордонів можна задати товщину, колір та стиль:

border-width – товщина кордону;

border-color – колір кордону (за замовчуванням – чорний);

border-style – стиль кордону.

Може приймати значення solid (за замовчуванням), dotted, dashed, double, groove, ridge, inset чи outset.

На рис. 1.6 представлені всі види меж, border - width встановлений у 5 пікселів.

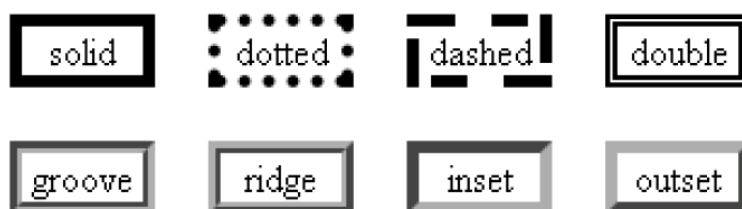


Рис.1.6 . Види меж

Існує скорочений запис: властивість border задає одночасно товщину, колір та стиль. Значення вказуються через пробіл у будь-якому порядку.
Наприклад:

```
<p style="border: solid 1 px green">Текст</p>
```

Загальні розміри елемента складаються так:

$Ширинa = width + padding + border + margin$

$Висота = height + padding + border + margin$

Тобто. `width` та `height` задають тільки розміри вмісту, не включаючи поля, заповнення та межі.

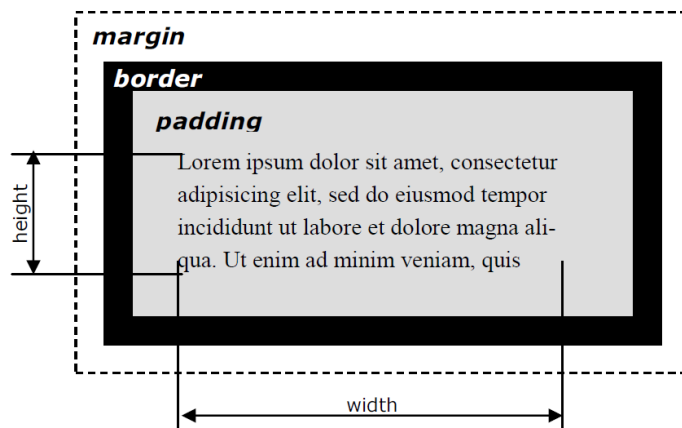


Рис.1.7. Бокс елемента та розміри вмісту

Як і в мові HTML, у CSS фоном служить заливка кольором або зображення. Фонове зображення може бути повторюваним.

`background - color` - встановлює колір фону.

Приклад:

```
td.head {background-color: #ffff00}
```

`background - image` - встановлює як фон зображення.

```
body {background-image: url(images/bg.jpg)}
```

`background - attachment` - задає поведінку фонового зображення при прокручування. За замовчуванням задається значення `scroll` – фон прокручується разом із вмістом. Значення `fixed` робить фон нерухомим.

`background - position` - початкове положення фонового зображення по горизонталі (`left` , `center` , `right`) і вертикалі (`top` , `center` , `bottom`). Замість ключових слів можна вказувати відстань у пікселях або відсотках.

`background - repeat` - вказує, в якому напрямку має розмножуватися фонове зображення:

- `repeat` – по горизонталі та вертикалі (за умовчанням);
- `repeat - x` – лише по горизонталі;
- `repeat - y` – тільки по вертикалі;
- `no - repeat` – вимкнути повторення.

Іноді потрібно застосувати стилі до фрагмента вмісту, що не включений в окремий тег. Наприклад, виділити фоном кілька слів у тексті.

Існують спеціальні теги `<div>...</div>` і `...`. Вони використовуються там, де не підходить інший тег. Самі собою вони не визначають ніякого форматування, але зручні для прив'язки до них стилів. При цьому DIV є блоковим елементом, а SPAN - рядковим.

Основна відмінність між блоковими і рядковими елементами полягає в наступному: рядкові елементи йдуть один за одним у рядку тексту, а блокові - розташовуються один за одним. До рядкових елементів відносяться такі теги, як `<a>`, ``, `<input>`, `<select>`, ``, `<sub>`, `<sup>` та ін.

До блокових: `<div>`, `<form>`, `<h1>` ... `<h6>`, ``, `<p>`, `<table>`, `` та деякі інші.

Розглянемо приклад для блокових тегів:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Блочні елементи</title>
    <style>
      h3, div, table
      {
        border: black dotted 1px;
        margin: 5px;
        padding: 5px;
      }
    </style>
  </head>
  <body>
    <h3>Заголовок</h3>
    <div>Вміст &#x3C; div &gt;
    <div>Вкладений &lt; div &gt; №1</div>
    <div>Вкладений &lt; div &gt; №2</div>
  </div>
  <table>
    <tr><td>Таблиця з одного середку</td></tr>
  </table>
  </body>
</html>
```

У браузері :

Заголовок
Вміст < div >
Вкладений < div > №1
Вкладений < div > №2
Таблиця з одного осередку

Рис.1.8. Поведінка блокових елементів

Зауваження. У HTML передбачено механізм вставки у документ будь-яких символів Юні-код – підстановки чи сутності (англ. *entities*). Підстановки дозволяють використовувати символи, відсутні на клавіатурі або навіть

використовуючи моє кодування (тобто навіть використовуючи кодування Windows -1251 можна вставити букву грецького алфавіту). Підстановки починаються з символу амперсанда і записуються у вигляді `&# DDDD` ; де DDDD – код символу в Юнікодi у десятиричній системі числення. Також можна записувати код у шістнадцятковій системі числення у формі `&# xHHHH` . Наприклад, знак копірайту © може бути заданий кодом `©` або `©` .

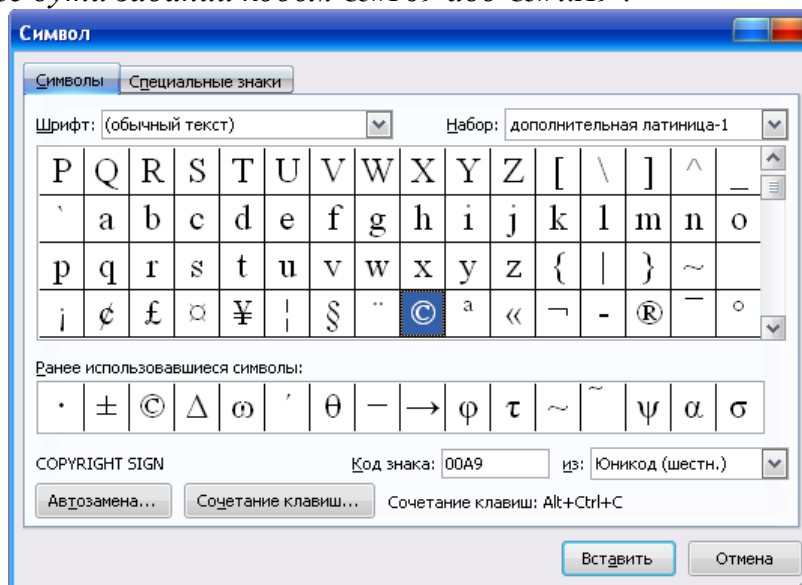


Рис.1.9.Інтерфейс «Таблиця символів»

Часто використовуються підстановки:
`& laquo` ; `i & raquo` ; – для фігурних дужок;
`& mdash` ; - для тире;
нерозривний пробіл `& nbsp` ;

`& lt` ; `i & gt` ; – для символів менше (`<`) та більше (`>`), які використовуються для завдання тегів HTML .

Блокові елементи розташовуються один під одним, багато хто займає всю можливу ширину. Блокові елементи можуть включати малі та інші блокові елементи. Але малі елементи не можуть містити блокові.

Ще однією відмінністю є те, що для малих елементів не працюють такі властивості, як `margin - top` , `margin - bottom` , `padding - top` і `padding - bottom` . Винятком є теги `` , `<input>` , `<textarea>` та `<select>` – для них можна задавати відступи `padding - top` та `padding - bottom` .

За замовчуванням блокові елементи йдуть строго один під одним. Змінити цей порядок можна, зробивши елементи «плаваючими». Для цього служить CSS атрибут `float` . Він задає, з якого боку вирівнюватиметься елемент: лівої (`left`) чи правої (`right`) . Плаваючий елемент буде прагнути до лівої або правої сторони батьківського елемента, а з інших сторін може обтікати текстом або іншими елементами.

Наочно робота `float` видно на прикладі:

```
<html>
<head>
```

```

<title>Плаваючі елементи</title>
<style>
DIV#floating {float: left;}
</style>
</head>
<body>
    Щоб правильно вибрати датчики температури, необхідно
    розуміти, для яких вимірювань вони призначені, і наскільки
    достовірним буде результат виміру . <br>
    <div id="floating"></div>
    Найчастіше для вимірювання температури застосовують такі
    типи датчиків: термістори, цифрові напівпровідникові
    датчики, термопари та термоперетворювачі опору. Далі
    докладніше зупинимось на їхніх перевагах та недоліках.<br>
    Термістори використовують принцип зміни опору провідника
    від температури. Чутливі елементи для термісторів
    виконуються з урахуванням різних оксидів металів. <br>
    Перевагами термісторів є висока чутливість, малий розмір
    та низька ціна. Основні недоліки - нелінійність
    характеристики, слабка завадостійкість, відсутність
    взаємозамінності. На останньому хотілося б зупинитись
    окремо. Така характеристика, як взаємозамінність, особливо
    важлива в тих випадках, коли датчики застосовують для
    вимірювання температури ґрунту, компосту або готового
    продукту, де використовується, як правило,
    низькокваліфікований персонал, де датчики часто тиснуть
    навантажувачами або ламають по необережності. Власне, у
    реальних робочих умовах ці датчики стають витратним
    матеріалом. У цих процесах використання термісторних
    датчиків температури не рекомендується. Справа в тому, що
    кожен виробник термісторів випускає їх із суто
    індивідуальною характеристикою, і у разі поломки такого
    датчика доведеться звертатися саме до оригінального
    виробника. Уявіть, якщо такий виробник перебуває в Китаї,
    скільки коштуватиме нашому українському господарству імпорт
    одиначної партії таких датчиків? < br >
    </body>
</html>

```

Контейнер DIV із зображенням прагне лівого краю документа, і з трьох сторін він отікається текстом (рис. 1.10).

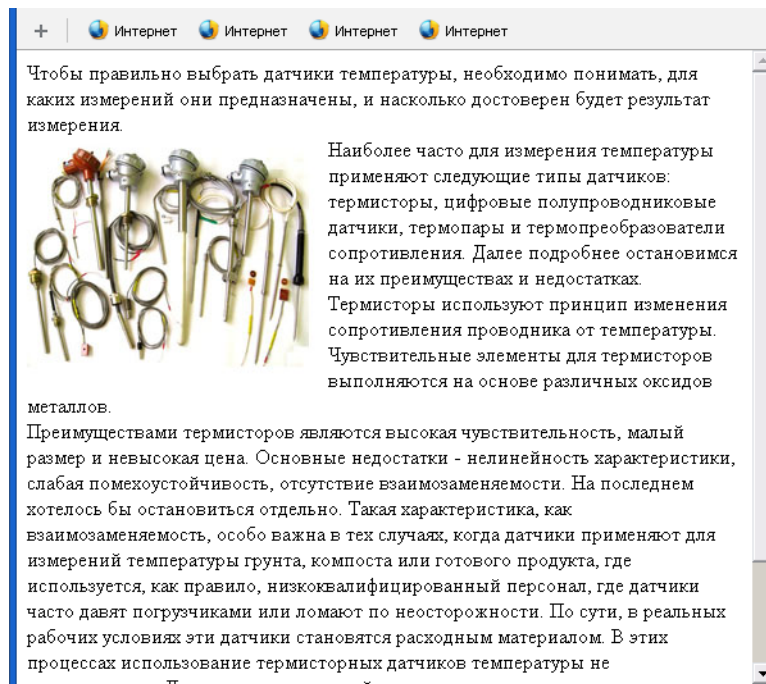


Рис.1.10. Обтікання текстом блокового елемента

1.3.Основи верстки сайтів

Під версткою веб-сторінки розуміють процес її створення шляхом компонування текстових та графічних елементів. Під час створення дизайну сайту дизайнер розробляє макет у графічному редакторі. Макет є звичайним зображенням. Тому, щоб використовувати дизайн на сайті, верстальник "перетворює" макет на веб-сторінку, де розташування елементів задається з використанням HTML і CSS .

Як правило, веб-сторінку можна розглядати як набір прямокутних блоків.

Можливі блоки на сторінці: логотип, верхня частина (header), ліва колонка, центральна колонка, права колонка, нижня частина (footer), загальний фон сторінки. Блоки в свою чергу можуть містити в собі інші дрібніші блоки: пункти меню, панелі, тощо. Як приклад, у верхній частині може розташовуватися основне навігаційне меню і форма пошуку, у лівій колонці - меню розділу, у правій колонці - важливі оголошення, в центральній - основний зміст сторінки, а в нижній частині дублюється верхнє меню (щоб для переходу в інший розділ не потрібно прокручувати всю сторінку знизу вгору) та контакти. Така верстка сторінки називається триколонною. Приклад на рис. Зрозуміло, існують інші варіанти компонування елементів, наприклад в одну колонку.



Рис.1.11. Триколонкова сторінка

Крім способу компонування блоків найважливішою характеристикою сторінки є спосіб її ширини. Існує два основні підходи:

1. Визначення строго фіксованої ширини сторінки. Якщо ширина сайту перевищує ширину вікна браузера, з'явиться горизонтальна смуга прокручування. Якщо ж ширина сайту буде меншою за ширину вікна, то з'явиться порожній простір з краю сторінки.

2. Прив'язка ширини сторінки до ширини екрана. У цьому випадку розмір блоків сторінки залежить від розмірів екрана. Якщо вікно звужується, звужуються і блоки. Якщо вікно розтягується, блоки розширюються. Така верстка часто називається гумовою.

Між фахівцями точаться суперечки, який із підходів кращий. Верстка, що «розтягується», може адаптуватися під різні дозволи екранів, але, з іншого боку, на невеликому моніторі блоки сторінки можуть занадто звужитися, а на широкоформатному екрані – стати занадто широкими. За таких умов користувачеві незручно читатиме текст на сайті. Щоб цього уникнути, необхідно задавати мінімальну та максимальну ширину гумової сторінки. Тому такий спосіб верстки є технічно складнішим, ніж фіксований.

Найбільш простим способом є верстка таблицями. Ідея полягає в тому, що всі елементи сторінки розміщуються у таблиці з невидимими межами.

Наприклад створимо триколоночну сторінку:

Каркасом такої сторінки буде таблиця із п'ятьма осередками. У таблиці буде два стовпці і три рядки, причому в першому та останньому рядку один осередок буде розтягнутий на 2 стовпці за допомогою атрибуту `colspan`.

Використовуючи таблиці, зручно задавати розміри осередків (блоків сторінки). У прикладі ширина бічних колонок дорівнюватиме 200 пікселям, центральної – 500 пікселям. Таким чином ми отримаємо жорсткий «каркас» сторінки. Висота блоків залежатиме від кількості вмісту в них. Висоту «шапки» та «підвалу» задамо 60 і 20 пікселів відповідно.

Для наповнення сторінки будемо використовувати класичний варіант умовного беззмістовного тексту, що вставляється в макет сторінки, Lorem ipsum.

Код сторінки:

```
<!DOCTYPE html>
<html>
  <head>
    <title> Таблична верстка </title>
    <style>
      BODY {
        margin: 0; /* обнулюємо поля біля сторінки */
        text-align: center; /* вирівнювання по центру */
      }

      #main {
        margin: 0 auto; /* центруємо таблицю */
        border-collapse: collapse; /* стуляємо осередки */
      }

      TD { /* задаємо стиль для всіх осередків */
        vertical-align: top;
        margin: 0;
        padding: 5px;
      }

      #header { /* стиль заголовка */
        background-color: #999999;
        text-align: center;
        height: 60px;
      }

      #left_col { /* стиль лівої колонки */
        width: 200;
        background-color: #bbbbbb;
      }

      #center_col { /* стиль центральної колонки */
        width: 500px;
      }

      #right_col { /* стиль правої колонки */
        width: 200px;
        background-color: #bbbbbb;
      }

      #footer { /* стиль нижнього блоку */
        background-color: #999999;
        text-align: center;
        height: 20px;
      }
    </style>
```

```

</head>
<body>
  <table id="main">
    <tr>
      <td colspan="2" id="header">
        HEADER
      </td>
    </tr>
    <tr>
      <td id="left_col">
        Lorem ipsum dolor sit amet, consectetur
        adipiscing elit, sed do eiusmod tempor incididunt ut labore et
        dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
        exercitation ullamco laboris nisi ut aliquip ex ea commodo
        consequat. Duis aute irure dolor in reprehenderit in voluptate
        velit esse cillum dolore eu fugiat nulla pariatur. Excepteur
        sint occaecat cupidatat non proident, sunt in culpa qui officia
        deserunt mollit anim id est laborum.
      </td>
      <td id="center_col">
        Lorem ipsum dolor sit amet, consectetur
        adipiscing elit, sed do eiusmod tempor incididunt ut labore et
        dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
        exercitation ullamco laboris nisi ut aliquip ex ea commodo
        consequat. Duis aute irure dolor in reprehenderit in voluptate
        velit esse cillum dolore eu fugiat nulla pariatur. Excepteur
        sint occaecat cupidatat non proident, sunt in culpa qui officia
        deserunt mollit anim id est laborum.
      </td>
    </tr>
    <tr>
      <td colspan="2" id="footer">
        FOOTER
      </td>
    </tr>
  </body>
</html>

```

Результат показано на рис.1.12.

HEADER	
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
FOOTER	

Рис.1.12 Верстка у дві колонки за допомогою таблиці

Якщо внести невеликі зміни в CSS стилі, можна зробити сторінку, що розтягується. Наприклад, можна залишити ширину бічних колонок фіксованою, а загальну ширину таблиці прив'язати до ширини вікна. Тоді розмір центральної колонки браузер буде обчислювати, віднімаючи із ширини таблиці ширину бічних колонок. Для цього достатньо для #main додати правило width : 90%, а у #center_col прибрати властивість width .

Для компонування складних сторінок можна використовувати вкладені таблиці.

Блокова верстка набагато складніша в освоєнні, ніж таблична. Зважаючи на неможливість опису всіх прийомів блокової верстки в навчально-методичних рекомендаціях надається лише короткий вступ до цієї теми.

Верстка за допомогою блоків є більш сучасною технологією, ніж таблична верстка. Основною ідеєю блокової верстки є використання елементів DIV та CSS -стилів.

Реалізуємо триколонкову сторінку за допомогою блокової верстки.

Спочатку необхідно визначити в HTML-кодї сторінки основні блоки:

```
<div id="wrap">
  <div id="header">...</div>
  <div id="left_col">...</div>
  <div id="center_col">...</div>
  <div id="right_col">...</div>
  <div id="footer">...</div>
</div>
```

Блок wrap є контейнером («оберткою») для всіх інших блоків сторінки: заголовка, лівої, середньої та правої колонок і «підвалу».

Тепер визначимо правила CSS . Рекомендується спочатку скинути параметри відступів для всіх елементів:

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

Задамо ширину блоку wrap і відцентруємо його:

```
#wrap {  
width : 1000 px ;  
margin : 0 auto ;  
}
```

Тепер задаємо параметри інших блоків. Для того, щоб ліва і права колонки зайняли своє місце, використовуємо правило float . Щоб опустити "підвал" вниз використовуємо clear.

```
#header {  
padding: 5px;  
  
background-color: #999999;  
text-align: center;  
height: 60px;  
}  
  
#left_col {  
float: left;  
padding: 5px;  
width: 200px;  
background-color: #bbbbbb;  
}  
  
#center_col { float: left;  
padding: 5px;  
width: 570px;  
background-color: #ffffff;  
}  
  
#right_col { float: right;  
padding: 5px;  
width: 200px;  
background-color: #bbbbbb;  
}  
  
#footer {clear: both;  
padding: 5px;  
background-color: #999999;  
text-align: center;  
height: 20px;  
}
```

Заповнимо блоки текстом Lorem ipsum.

Повний код:

```
<!DOCTYPE html>
<html>
  <head>
    <title> Use div </title>
    <style>
      * {
        margin : 0;
        padding : 0;
      }
      #wrap {
        width : 1000 px ;
        margin : 0 auto ;
      }
      #header {
        padding: 5px;
        background-color: #999999;
        text-align: center;
        height: 60px;
      }

      #left_col {
        float: left;
        padding: 5px;
        width: 200px;
        background-color: #bbbbbb;
      }

      #center_col { float: left;
        padding: 5px;
        width: 570px;
        background-color: #ffffff;
      }

      #right_col { float: right;
        padding: 5px;
        width: 200px;
        background-color: #bbbbbb;
      }

      #footer {clear: both;
        padding: 5px;
        background-color: #999999;
        text-align: center;
        height: 20px;
      }
    </style>
  </head>
  <body>
    <div id="wrap">
```

```

        <div id="header">The title of page</div>
        <div id="left_col">Lorem ipsum dolor sit
amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi
ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id
est laborum.</div>
        <div id="center_col">Lorem ipsum dolor sit
amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi
ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id
est laborum.</div>
        <div id="right_col">Lorem ipsum dolor sit
amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi
ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id
est laborum.</div>
        <div id="footer">&copy; 2024 All rights
reserved.</div>
    </div>
</body>
</html>

```

Отриманий результат представлений на рис.1.13

THE TITLE OF PAGE		
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
© 2024 All rights reserved.		

Рис.1.13.Верстка у три колонки за допомогою блоків div

2. ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Індивідуальне завдання – виконання блокової html/css верстки шаблону дизайну для вебсайту.

Вихідні дані завдання надаються в електронному вигляді та містять зображення веб-сторінки, яку потрібно розробити. Крім того, надаються зображення, з яких складається дизайн сторінки.

За кожним варіантом необхідно створити окремо файл html та css із кодом, що відповідно до завдання створює сторінку.

Варіант індивідуального завдання визначає викладач в період аудиторних занять.

3. РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Крок 1. Аналіз макету

Розглядаємо зображення-шаблон як пазл. Розділяємо її на окремі блоки: заголовок, меню, контент, футер тощо. Кожен блок буде відповідати певному HTML-елементу.

Звертаємо увагу на шрифти, кольори, розміри, відступи та інші стилістичні елементи. Це в майбутньому допоможе визначити необхідні CSS-властивості.

Крок 2: Створення HTML-структури

Необхідно створити файл з розширенням .html та написати базову структуру документа.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Індивідуальна робота</title>
</head>
<body>
  </body>
</html>
```

Важливим тегом для структурування сторінки є тег <div>.

Використовуючи його, можна ділити свою сторінку на блоки. У середині даного тега можливо розмістити інші HTML теги, які будуть потрібні для створення вмісту сторінки (посилання, текст, зображення і т.п.). Всі ці

елементи згодом знайдуть свій стиль за допомогою каскадних таблиць стилю CSS.

Нехай маємо такі блоки нашої веб сторінки (рис.3.1) :

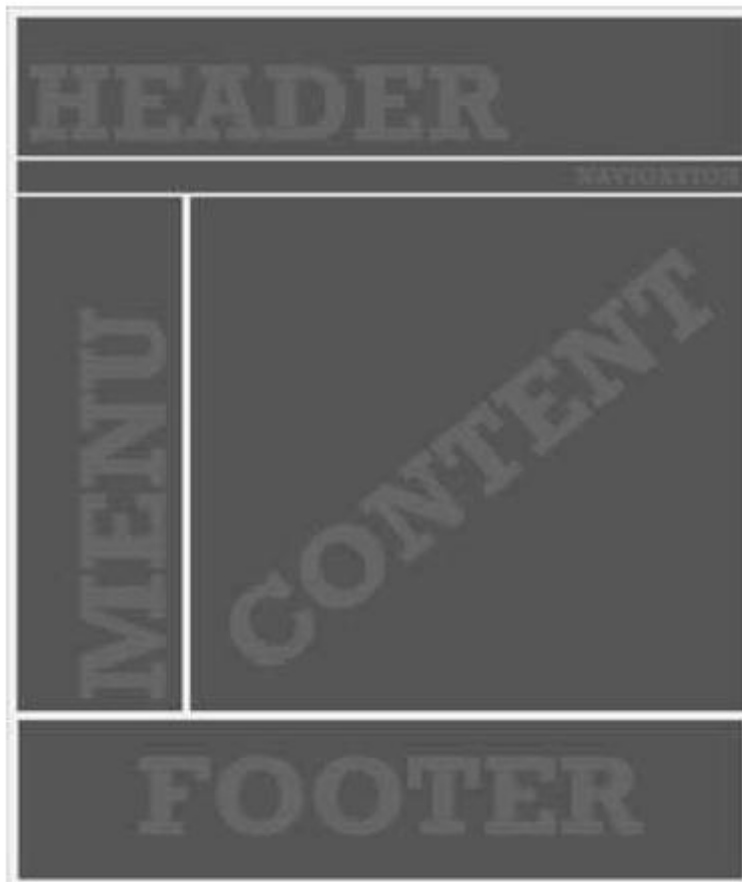


Рис.3.1. Розмітка сторінки

Код HTML цієї сторінки буде виглядати, як показано нижче:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Індивідуальна робота</title>
  </head>
  <body>
    <div id="container">
      <div id="header">
      </div>
      <div id="navigation">
      </div>
      <div id="menu">
      </div>
      <div id="content">
      </div>
      <div id="footer">
      </div>
    </div>
  </body>
</html>
```

```
</body>  
</html>
```

Кожному тегу div надано певний селектор ID. Тобто кожному окремому блоку сторінки надано свою назву. Це необхідно для того, щоб пізніше, задаючи стиль блокам у файлі CSS, можливо було б відрізнати один блок від іншого.

Існує два види селекторів: ID - використовується для унікальних блоків, таких, які більше ніде на сторінці не повторюються. У той час як селектор CLASS - застосуються для позначення повторюваних блоків. Також, всі блоки було укладено всередину блоку "container ". Це зроблено для того, щоб згодом можна було помістити сторінку цілком так, як потрібно розробнику.

Крок 3: Написання CSS-стилів

Як вже було сказано, CSS таблиця стилів визначає дизайн елементів веб сторінки. Її можливо розмістити її як всередині html документа, так і у зовнішньому файлі.

Якщо відкрити щойно зроблено веб-сторінку в браузері, то нічого там виявити не буде можливим. Це тому, що вона не наповнена будь-яким вмістом, а блоки не мають будь-якої форми та кольору. Контент для блоків можливо додати між відповідними тегамі. А ось форму, розташування колір та інші зовнішні прояви блоків потрібно завдавати за допомогою файлу CSS.

Створюємо відповідний файл. Назвемо його style.css і розмістимо в тій же папці, що і документ html. Після цього в html файлі, між тегамі <head> необхідно додати наступний рядок :

```
<link rel="stylesheet" href="styles.css">
```

Таким чином ці два файли пов'язуються разом. Тепер при відображенні вебсторінки браузер буде використовувати відповідний файл стилів.

Синтаксис CSS складається з селектора, властивості і значення.

Використовуємо селектори для вибору елементів і задаємо їм необхідні стилі.

У нас є наступні селектори :

- body
- container
- header
- navigation
- menu
- content
- footer

Код файлу стилів style.css виглядатиме так :

```
body {
  background : # f3f2f3 ;
  color : # 000000 ;
  font - family : Trebuchet MS, Arial, Times New Roman
;
  font - size : 12px ;
}
#container {
  background : # FFFFFFFF ;
  margin : 30px auto ;
  width : 900px ;
  height : 1000px ;
}
#header {
  background : # 838283 ;
  height : 200px ;
  width : 900px ;
}
#navigation {
  background : # a2a2a2 ;
  width : 900px ;
  height : 20px ;
}

#menu {
  background : # 333333 ;
  float : left ;
  width : 200px ;
  height : 600px ;
}
#content {
  background : # d2d0d2 ;
  float : right ;
  width : 700px ;
  height : 600px ;
}
#footer {
  background : # 838283 ;
  height : 180px ;
  width : 900px ;
}
```

Для коректного відображення створеної веб сторінки (після використання властивості float) використовують додатковий блок, який слід розмістити відразу за блоком content.

Ось його код HTML:

```
<div id="clear"> </div>
```

I CSS:

```
#clear {
    clear: both;
}
```

Між тегами head записати, яке кодування необхідно використовувати браузеру, якщо буде потрібно додати український текст в свої блоки :

```
<meta charset="UTF-8">
```

Фінальний код HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Індивідуальна робота</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <div id="container">
      <div id="header"> Заголовок
    </div>
    <div id="navigation"> Навігація
    </div>
    <div id="menu"> Меню
    </div>
    <div id="content"> Основний контент
    </div>
    <div id="clear">
    </div>
    <div id="footer"> Футер
    </div>
  </div>
</body>
</html>
```

CSS:

```
body {
    background: #f3f2f3;
    color: #000000;
    font-family: Trebuchet MS, Arial, Times New Roman;
    font-size : 14px;
}
#container {
    background : #FFFFFF ;
    margin : 30px auto ;
    width : 900px ;
    height : 1000px ;
}
```

```
    }  
#header {  
    background : #838283 ;  
    height : 200px ;  
    width : 900px ;  
    }  
#navigation {  
    background : #a2a2a2 ;  
    width : 900px ;  
    height : 20px ;  
    }  
  
#menu {  
    background : #333333 ;  
    float : left ;  
    width : 200px ;  
    height : 600px ;  
    }  
#content {  
    background : #d2d0d2 ;  
    float : right ;  
    width : 700px ;  
    height : 600px ;  
    }  
#clear {  
    clear: both;  
    }  
#footer {  
    background : #838283 ;  
    height : 180px ;  
    width : 900px ;  
    }
```

Ось як сторінка буди виглядати в браузері :

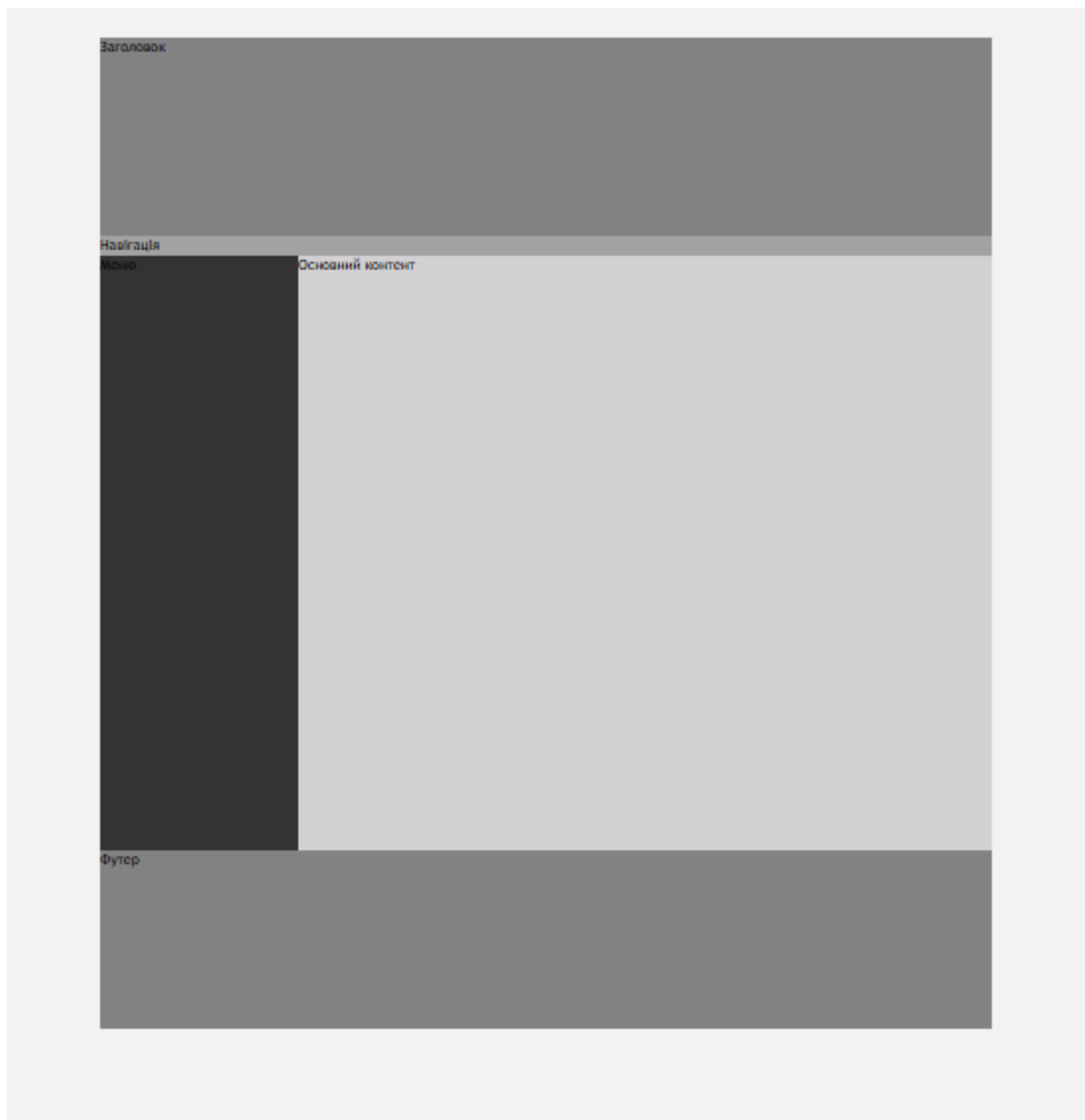


Рис.3.2. Фінальна сторінка

Наповнення блоків контентом (корисною інформацією) здійснюється або прямо текстом, або із використанням тегів параграфу `<p></p>`, зображенням – тег ``, гіперпосиланнями – тег `<a>`. При створенні меню сайту слід користуватися неупорядкованим списком – теги `` та `` для сумісності меню із сучасними системами управління контентом (CMS).

Для прикладу створимо веб сторінку, що представлена на рис. 3.3.

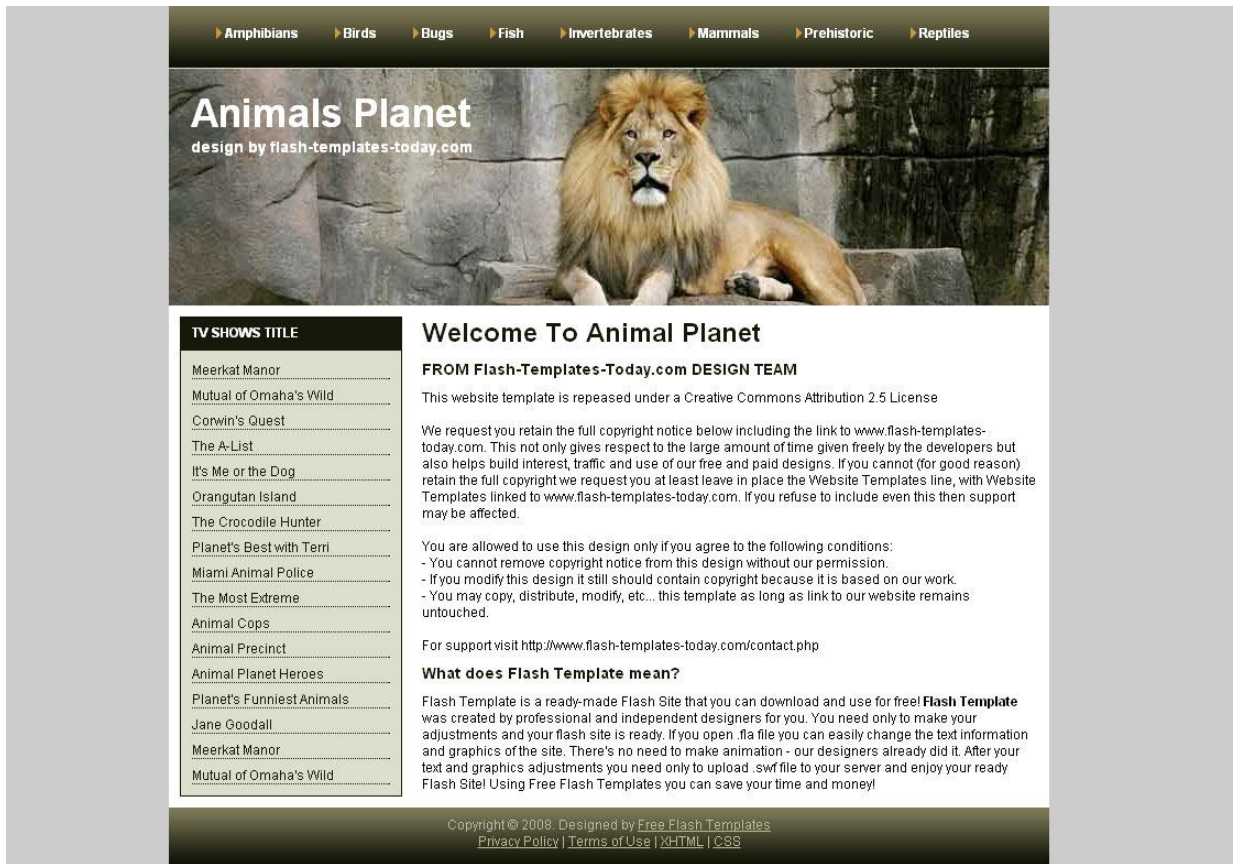


Рис.3.3. Mainpage.jpg

В робочій папці створюємо папку images, в якій розміщуємо наступні рисунки:



Рис.3.4. headerBackground.jpg



Рис.3.5. mainMenuList.gif



Рис. 3.6.menuBackground.gif

Крок 1. Аналіз макету

Розглядаємо зображення-шаблон як пазл. Розділяємо її на окремі блоки: заголовок, меню, контент, футер тощо. Кожен блок буде відповідати певному HTML-елементу.

Крок 2: Створення HTML-структури

Створюємо файл з розширенням `.html` та створюємо базову структуру документа.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Animals Planet</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8"/>
  </head>
  <body>
  </body>
</html>
```

Всередині тега `<body>` вставляємо необхідні блоки.

```
<div id="container">
  <div id="layoutMenu"></div>
  <div id="layoutHeader"></div>
  <div id=" ">
    <div id="layoutNavigation"></div>
    <div id="layoutBodyContent"></div>
  </div>
  <div class="clearer"></div>
  <div id="layoutFooter"></div>
</div>
```

Блок `container` є контейнером («оберткою») для всіх інших блоків сторінки: верхнього меню `layoutMenu`, заголовка `layoutHeader`, основного контенту `layoutCenterBody` (навігаційне меню зліва `layoutNavigation` та основний блок `layoutBodyContent`) та «підвалу» `layoutFooter`.

Для коректного відображення створеної вебсторінки (після використання властивості `float`) використовуємо додатковий блок `clearer`.

Усередині блоків можливо розмістити інші HTML теги, які потрібні для створення вмісту сторінки (посилання, текст, зображення і т.п.).

Код HTML сторінки стає таким:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Animals Planet</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-
8"/>
  </head>
  <body>
```

```

<div id="container">
<div id="layoutMenu">
  <ul>
    <li><a href="#">Amphibians</a></li>
    <li><a href="#">Birds</a></li>
    <li><a href="#">Bugs</a></li>
    <li><a href="#">Fish</a></li>
    <li><a href="#">Invertebrates</a></li>
    <li><a href="#">Mammals</a></li>
    <li><a href="#">Prehistoric</a></li>
    <li><a href="#">Reptiles</a></li>
  </ul>
</div>
<div id="layoutHeader">
  <h1><a href="#">Animals Planet</a></h1>
</div>
<div id="layoutCenterBody">
<div id="layoutNavigation">
  <div class="navTitle">
    <h3>TV SHOWS TITLE</h3>
  </div>
  <ul>
    <li><a href="#">Meerkat Manor</a></li>
    <li><a href="#">Mutual of Omaha's Wild </a></li>
    <li><a href="#">Corwin's Quest</a></li>
    <li><a href="#">The A-List</a></li>
    <li><a href="#">It's Me or the Dog</a></li>
    <li><a href="#">Orangutan Island</a></li>
    <li><a href="#">The Crocodile Hunter </a></li>
    <li><a href="#">Planet's Best with Terri</a></li>
    <li><a href="#">Miami Animal Police</a></li>
    <li><a href="#">The Most Extreme</a></li>
    <li><a href="#">Animal Cops</a></li>
    <li><a href="#">Animal Precinct</a></li>
    <li><a href="#">Animal Planet Heroes</a></li>
    <li><a href="#">Planet's Funniest Animals</a></li>
    <li><a href="#">Jane Goodall</a></li>
    <li><a href="#">Meerkat Manor</a></li>
    <li><a href="#">Mutual of Omaha's Wild</a></li>
  </ul>
</div>
<div id="layoutBodyContent">
  <h1>Welcome To Animal Planet</h1>
  <h2>FROM DESIGN TEAM</h2>
  <p>
  <br/>

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```

    <br/>
  </p>
  <h2>What does Lorem Ipsum mean?</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit
in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui
officia deserunt mollit anim id est laborum.</p>
</div>
</div>
<div class="clearer"></div>
<div id="layoutFooter">
  <p>Copyright © 2024. Designed by <a href="#" title="Free
Flash Templates">design team</a><br />
  <a href="#">Privacy Policy</a> | <a href="#">Terms of
Use</a> | <a target="_blank" href="#"><abbr title="eXtensible HyperText
Markup Language">XHTML</abbr></a> | <a target="_blank" href="#"><abbr
title="Cascading Style Sheets">CSS</abbr></a></p>
</div>
</div>
</body>
</html>

```

Крок 3: Написання CSS-стилів

Створюємо файл з розширенням .css та підключаємо його до html. Для його між тегами <head> додаємо наступний рядок :

```
<link href="style.css" rel="stylesheet" type="text/css">
```

Код файлу стилів style.css виглядатиме так:

```

* {
  margin:0;
  padding:0;
}
body {
  font-family:Arial, Helvetica, sans-serif;
  font-size:12px;
  background:#CCCCCC;
}
#container {
  width:800px;
  margin:auto;
  background:#FFFFFF;
}
/* Задаємо стилі заголовка */
#layoutHeader {
  width:800px;
  height:216px;
}

```

```

        background:url(images/headerBackground.jpg) no-repeat;
    }
    #layoutHeader h1 {
        color:#FFFFFF;
        font-size:36px;
        font-weight:bold;
        padding:20px 20px 0 20px;
    }
    #layoutHeader h2 {
        color:#FFFFFF;
        font-size:14px;
        font-weight:bold;
        padding:0 20px 0 20px;
    }
    #layoutHeader h1 a {
        color:#FFFFFF;
        font-size:36px;
        font-weight:bold;
        text-decoration:none;
    }
    #layoutHeader h2 a {
        color:#FFFFFF;
        font-size:14px;
        font-weight:bold;
        text-decoration:none;
    }
    /* Задаємо стилі основного блоку */
    #layoutCenterBody {
        margin:10px 10px 0 10px;
    }
    #layoutNavigation {
        border:1px solid #15190A;
        width:200px;
        float:left;
        background:#DCDECC;
    }
    #layoutNavigation .navTitle {
        background:#15190A;
        height:30px;
    }
    #layoutNavigation .navTitle h3 {
        color:#FFFFFF;
        font-size:12px;
        font-weight:bold;
        padding:6px 10px;
    }
    #layoutBodyContent {
        margin-left:220px;
    }
    #layoutBodyContent a {
        color:#7A7D5F;
        font-size:12px;
        font-weight:bold;
    }

```

```

}
#layoutBodyContent h1{
    color:#15190A;
    font-weight:bold;
    font-size:24px;
}
#layoutBodyContent h2{
    color:#15190A;
    font-weight:bold;
    font-size:14px;
    margin-top:10px;
}
#layoutBodyContent p {
    margin-top:10px;
}
.clearer {
    clear:both;
    height:10px;
}
/* Задаємо стилі "підвалу" */
#layoutFooter {
    height:56px;
    background:url(images/menuBackground.gif) repeat-x;
    text-align:center;
}
#layoutFooter p {
    padding-top:10px;
    color:#BCBA9E;
}
#layoutFooter a {
    color:#BCBA9E;
    text-decoration:underline;
}
#layoutFooter a:hover {
    color:#BCBA9E;
    text-decoration:none;
}
/* Задаємо стилі основного меню */
#layoutMenu {
    height:56px;
    background:url(images/menuBackground.gif) repeat-x;
}
#layoutMenu ul {
    text-align:center;
    padding-top:17px;
}
#layoutMenu li {
    list-style: none;
    display: inline;
}
#layoutMenu li a {
    padding: 3px 30px 3px 8px;
    text-decoration: none;
}

```

```

        background:url(images/mainMenuList.gif) no-repeat left
5px;
        font-size:12px;
    }
#layoutMenu li a:link {
    color: #ffffff;
    font-weight:bold;
    font-size:12px;
}
#layoutMenu li a:visited {
    color: #ffffff;
}
#layoutMenu li a:hover {
    color: #CA9847;
}

/* Задаємо стилі лівого навігаційного меню */
#layoutNavigation ul {
    list-style-type: none;
    margin:10px 0;
}
#layoutNavigation a {
    display: block;
    width: 180px;
    background-color: #DCDECC;
    border-bottom: 1px dotted #15190A;
    margin:7px 10px;
}
#layoutNavigation a:link, #layoutNavigation a:visited {
    color: #15190A;
    text-decoration: none;
}
#layoutNavigation a:hover {
    background-color: #B7BA9E;
    color: #fff;
}

```

Вигляд створеної вебсторінки у браузері представлено на рис.3.7.



Рис.3.7. Створена вебсторінка

СПИСОК ЛІТЕРАТУРИ

1. HTML Підручник. Початок. *W3Schools українською. Безплатні уроки онлайн для початківців, школярів та студентів.* URL: <https://w3schoolsua.github.io/html/index.html#gsc.tab=0> (дата звернення: 25.07.2024).
2. CSS Підручник. Каскадна таблиці стилів. Стилізація сайту. *W3Schools українською. Безплатні уроки онлайн для початківців, школярів та студентів.* URL: <https://w3schoolsua.github.io/css/index.html#gsc.tab=0> (дата звернення: 25.07.2024).

Навчально-методичне видання

Рибальченко Марія Олександрівна,
Маначин Іван Олександрович,
Бурчак Андрій Анатолійович

РНР-ПРОГРАМУВАННЯ

Навчально-методичні рекомендації до виконання індивідуального завдання

Електронне видання

Експертний висновок склав канд. техн. наук, доц. В. Шибакінський

Зареєстровано НМВ УДУНТ (№ 14 від 22.01.2025)

В авторській редакції
Комп'ютерна верстка М. О. Рибальченко

Формат 60x84 _{1/16}. Ум. друк. арк. 2,72. Обл.-вид. арк. 2,76.
Зам. № 7

Видавець: Український державний університет науки і технологій
вул. Лазаряна, 2, ауд. 2216, м. Дніпро, 49010.
Свідоцтво суб'єкта видавничої справи ДК № 7709 від 14.12.2022

Адреса видавця та дільниці оперативної поліграфії:
вул. Лазаряна, 2, Дніпро, 49010