

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

ІНФОРМАТИКА В СФЕРІ КОМУНІКАЦІЙ

**Навчально-практичний посібник
у 3-х частинах**

**Частина 3. Використання web-технологій
у сфері комунікацій**

*За загальною редакцією
д-ра техн. наук, професора С. Г. Удовенка*

**Харків
ХНЕУ ім. С. Кузнеця
2020**

УДК 004(075.034)

I-74

Авторський колектив: д-р техн. наук, професор С. Г. Удовенко – вступ, теоретичні відомості за розділами 5 – 7, загальна редакція; канд. техн. наук, доцент В. А. Затхей – п. 16, 17, тестові завдання за розділом 6; канд. техн. наук, доцент О. В. Гороховатський – п. 18, тестові завдання за розділом 7; канд. техн. наук, ст. викладач О. О. Передрій – п. 19; канд. техн. наук, доцент О. В. Тесленко – п. 14, тестові завдання за розділом 5; канд. техн. наук, доцент Н. О. Бринза – п. 15; канд. техн. наук, доцент О. О. Тютюник, канд. екон. наук, доцент О. В. Вільхівська – п. 13.

Рецензенти: завідувач кафедри комп'ютерної математики і аналізу даних Національного технічного університету "Харківський політехнічний інститут", д-р техн. наук, професор *Л. М. Любчик*; завідувач кафедри електронних обчислювальних машин Харківського національного університету радіоелектроніки, д-р техн. наук, доцент *А. А. Коваленко*.

Рекомендовано до видання рішенням ученої ради Харківського національного економічного університету імені Семена Кузнеця.

Протокол № 2 від 25.06.2020 р.

Самостійне електронне текстове мережеве видання

Інформатика в сфері комунікацій [Електронний ресурс] : навчально-практичний посібник : у 3-х частинах. Частина 3. Використання web-технологій у сфері комунікацій / С. Г. Удовенко, В. А. Затхей, О. В. Гороховатський та ін. ; за заг. ред. д-ра техн. наук, професора С. Г. Удовенка. – Харків : ХНЕУ ім. С. Кузнеця, 2020. – 155 с.
ISBN 978-966-676-798-4

Подано теоретичні відомості про використання web-технологій для розроблення сайтів. На практичних прикладах розглянуто використання мови розмітки HTML, каскадних таблиць стилів CSS3 та системи керування контентом WordPress. Уміщено індивідуальні завдання для самостійного виконання та набуття практичних навичок і вмій створення web-сайтів, а також тестові завдання для перевірки рівня засвоєння теоретичного матеріалу та способів практичної роботи.

Рекомендовано для студентів економічних та гуманітарних спеціальностей.

УДК 004(075.034)

© Удовенко С. Г., Затхей В. А.,
Гороховатський О. В. та ін., 2020

© Заг. ред. Удовенка С. Г., 2020

© Харківський національний економічний
університет імені Семена Кузнеця, 2020

ISBN 978-966-676-798-4

Вступ

В умовах всебічної інформатизації суспільства і стрімкого проникнення інформаційно-комунікаційних технологій у різні сфери діяльності людини ідеологія інформаційної грамотності стає необхідною нормою в процесі підготовки майбутніх фахівців сфери комунікацій.

Використання web-технологій дозволяє створювати віртуальне інформаційне середовище, в якому користувачі отримують доступ до спільних даних, аналізу інформації та обміну думками.

Мережа "Інтернет" пропонує нові можливості для спілкування завдяки системі блогів (сайтів). Набуття компетентностей у сфері інформаційних технологій є необхідною складовою успішної професійної діяльності у сфері комунікацій.

Необхідним елементом успішного засвоєння навчального матеріалу є самостійна робота студентів із технічною літературою та сучасними програмними засобами розроблення сайтів (блогів).

Третя частина навчально-практичного посібника "Інформатика у сфері комунікацій" охоплює питання використання HTML5, CSS3 і WordPress та за змістом відповідає третьому змістовому модулю базової навчальної дисципліни "Інформатика в сфері комунікацій" для студентів спеціальності 061 "Журналістика" першого (бакалаврського) рівня.

Основною метою написання третьої частини навчально-практичного посібника є формування професійних компетентностей із володіння web-технологіями та вміння використовувати їхні можливості в сфері комунікацій, створювати власний сайт, розміщувати його в інтернеті та керувати його контентом.

На початку кожного розділу наведено базові теоретичні відомості про роботу з HTML5, CSS3 та WordPress і розглянуто основні елементи верстки сайтів.

Кожен із розділів подано як самовчитель за темами, що за назвою та змістом відповідають темам лабораторних робіт із навчальної дисципліни "Інформатика в сфері комунікацій". У навчально-практичному посібнику вміщено п'ять тем для опанування способів роботи з HTML5 і CSS3, набуття навичок адаптивної верстки, роботи з Bootstrap та дві теми для опанування професійних способів роботи з WordPress.

Для кожної теми визначено мету та наведено компетентності, яких студенти набувають у процесі опанування навчального матеріалу, подано стислі теоретичні відомості за матеріалом, що розглядають, та показано способи і методи професійної роботи з HTML5, CSS3 та WordPress у межах теми заняття. Для пояснення навчального матеріалу наведено скріншоти виконання завдань у відповідних додатках.

Для підвищення ефективності засвоєння теоретичного навчального матеріалу наприкінці кожної теми сформовано запитання для самоконтролю, а для набуття практичних навичок запропоновано індивідуальні завдання для практичного виконання.

Після засвоєння теоретичного матеріалу та відпрацювання практичного матеріалу в навчально-практичному посібнику для кожного із розділів передбачена система самодіагностики на базі тестових завдань. Для самостійної перевірки рівня остаточних знань у кінці навчально-практичного посібника наведено відповіді до тестових завдань за розділами.

Навчально-практичний посібник буде корисний для всіх, хто бажає набути професійних навичок у роботі з HTML5, CSS3 та WordPress. Матеріал навчально-практичного посібника може бути використаний для підготовки студентів економічних та гуманітарних спеціальностей.

Розділ 5. Основи web-дизайну

Метою розділу є формування у майбутніх фахівців системи компетентностей з використання мови розмітки гіпертексту HTML в процесі створення сайтів.

Професійні компетентності

У результаті вивчення навчальної дисципліни студент має:

знати:

принципи побудови сайтів;

технологію управління контентом сайтів;

вміти:

створювати, редагувати та формувати документи складної структури з використанням мови HTML5;

ставити завдання на розроблення сайту відповідно до професійної діяльності.

Ключові терміни: HTML, гіпертекст, web-сторінка, сайт, тег, атрибут, текстовий редактор, браузер, таблиця, об'єднання клітинок, таблична верстка, форма, елементи управління.

Теоретичні відомості

Для створення web-сторінок і web-додатків використовується мова гіпертекстової розмітки HTML (HyperText Markup Language). Роком її створення вважається 1990 рік, а творцем та ідеологом став Тімоті Джон Бернерс-Лі, який зумів, використовуючи нові мережеві технології й досвід своїх попередників, зробити Всесвітню павутину приємним і зручним засобом поширення інформації у мережі "Інтернет". Для позначення Всесвітньої павутини також використовують термін web (англ. web –"павутина") і аббревіатуру WWW (World Wide Web).

Одночасно з розвитком web-технологій, насиченням інтернету новими сервісами й можливостями розвивалася й мова розмітки гіпертексту. З моменту своєї появи стандарт HTML перетерпів безліч змін, останньою з яких є нова редакція HTML 5.3, анонсована W3C (World Wide Web Consortium) 18 жовтня 2018 року.

Виділимо базові елементи web-технологій:

- Мережа "Інтернет" – це всесвітня мережа різнорідних комп'ютерних мереж, які взаємодіють за протоколом TCP/IP (Transmission Control Protocol Internet Protocol);
- web є одним із застосунків в мережі "Інтернет" (поряд з електронною поштою, новинами й іншими електронними сервісами), призначеним для масового поширення різноманітної інформації;
- носіями інформації у web слугують web-сторінки, що містять текст, графіку, елементи мультимедіа й гіперпосилання на інші ресурси web або мережі "Інтернет". Такі файли мають розширення html;
- для передачі гіпертексту web-сторінок у мережі "Інтернет" використовується спеціально розроблений протокол HTTP (Hyper Text Transfer Protocol);
- для розроблення web-сторінок використовується спеціальна мова розмітки гіпертексту HTML;
- для форматування зовнішнього вигляду web-сторінок використовуються каскадні таблиці стилів CSS (Cascading Style Sheets);
- для формування коду web-сторінки використовуються текстові редактори;
- для перегляду web-сторінок використовується спеціальна клієнтська програма web-браузер. У вікні web-браузера відображаються результати інтерпретації мови HTML із web-сторінок.

13. Знайомство з web. Перша HTML-сторінка

Мета: вивчити основи мови HTML. Познайтися з текстовим редактором Notepad++. Створити першу HTML-сторінку.

Ключові терміни: HTML, гіпертекст, web-сторінка, сайт, тег, атрибут, текстовий редактор, браузер.

Теоретичні відомості

Базовим елементом мови розмітки гіпертексту HTML є тег (дескриптор, елемент, маркер).

Тегом (від англ. слова tag – поименована мітка) в HTML вважається текст із назвою елемента розмітки, який розміщується між трикутними дужками "<" та ">".

Розрізняють відкриваючі та закриваючі теги.

Відкриваючий тег "< >" слугує для вказівки програмі-браузеру початку якого-небудь об'єкта або завдання властивостей об'єктів, поміщених у контейнер.

Закриваючий тег "</ >" слугує для вказівки програмі-браузеру про кінець об'єкта або закінчення застосування властивостей, заданих у відкриваючому тегу.

Теги можуть бути двох видів:

а) одинарні (не потребують закриття). Ставляться перед текстом, якими вони управляють. Наприклад:
 Книга – текст буде написано з нового рядка;

б) парні (потребують закриття). Вказують інтервал дії команди. Наприклад:

<I> Мій перший сайт </I> – текст буде написано курсивом.

Теги зазвичай мають атрибути (параметри) – вказівки про те, як має браузер сприймати і обробляти теги.

Атрибут записується після назви тегу і подається, здебільшо, у вигляді пари: ім'я атрибута = "значення". Значення атрибута береться в подвійні (") або одинарні (') лапки. Якщо атрибут не містить пробілів, лапки можуть не застосовуватися. Наприклад: face="Times New Roman", align = center.

Тег може мати декілька атрибутів, які відокремлюються один від одного пробілами. Наприклад: <ТЕГ атрибут1 = "ЗНАЧЕННЯ" ... атрибутN = "ЗНАЧЕННЯ">

Коментарі – це фрагменти коду, які не обробляються браузером і слугують для службової розмітки документа і записуються в такому вигляді:

<! - - Текст коментаря - ->

У web-документі, відкритому в браузері, коментарі не відображаються.

Структура HTML-документу. HTML-документ становить звичайний текстовий файл, що містить розмічений тегами текст.

Документ HTML починається відкриваючим тегом <HTML> і закінчується закриваючим тегом </HTML>. Між цією парою контейнерних тегів розташовуються дві інші основні частини HTML документа: контейнер <HEAD>...</HEAD>, у якому розміщуються заголовок сторінки і службові теги (*meta*-теги, *link*-теги та ін.), і контейнер <BODY>...</BODY> – тіло документа.

Примітка. Мова HTML не чутлива до регістра. Теги можуть набиратися як малими, так і великими літерами, тобто команда <title> еквівалентна команді <TITLE> або <Title>.

Елемент <!DOCTYPE> призначений для вказівки типу поточного документа і для всіх документів задається у вигляді <!DOCTYPE HTML>. Це необхідно, щоб браузер розумів, як слід інтерпретувати поточну web-сторінку, оскільки HTML існує в декількох версіях.

Тег <HTML> визначає межі документа HTML, йому відповідає кінцевий тег </HTML>. Між цими двома тегами розташовується весь документ.

Тег <HEAD> – контейнер заголовка документа, який містить назву та службові дані.

У цьому контейнері обов'язковим є парний тег <TITLE> </TITLE>, контейнер із назвою документа, вміст якого відображається в рядку заголовка вікна браузера. Пошукові системи у першу чергу звертають на нього увагу.

Крім того, в заголовку сторінки розміщуються інші службові теги, вміст яких не відображається на сторінці, але вони мають важливе значення для правильного відображення web-документа.

Кодування HTML-сторінки потрібно вказувати для того, щоб web-браузер міг правильно відображати текст на сторінці. Якщо браузер

неправильно визначить кодування, то замість тексту будуть відображатися літери недрукованого тексту.

Щоб повідомити браузеру кодування HTML-сторінки, необхідно всередині тегу <HEAD> використовувати тег: <meta charset = "ім'я кодування">. Найпоширеніше сучасне кодування – utf-8. Таким чином, кодування можна задати таким чином: <meta charset = "utf-8">.

Доцільно вказати службові теги, які будуть визначати просування вашого сайту в пошукових системах, а саме:

```
<meta name = "description" content=" ">  
<meta name = "keywords" content=" ">
```

Мета-тег *description* (мета-опис) призначений для створення короткого опису (перелік назв) HTML-сторінки. Його вміст може використовуватися пошуковими системами для формування позиції (місця) сайту під час пошуку. Цей тег не впливає на зовнішній вигляд сторінки, оскільки є службовою інформацією. Для формування переліку назв сайту доцільно використовувати 50 – 300 символів.

Keywords – це мета-тег, що вписується в заголовок HTML-коду веб-сторінки з метою вказівки пошуковій системі набору ключових слів, релевантних тексту на сторінці. Для формування ключових слів доцільно використовувати 150 – 200 символів.

У контейнері заголовка розміщуються також інші службові теги, призначення яких буде розглянуто далі.

Таким чином, структуру HTML-документа наведено на рис. 5.13.1.

```
<!doctype html>  
<html>  
  <head>  
    <title> Заголовок документа </title>  
    <meta charset = "utf-8">  
    <meta name = "description" content="">  
    <meta name = "keywords" content="">  
  </head>  
<body>  
Тіло документа (відображається в браузері)  
</body>  
</html>
```

Рис. 5.13.1. Структура HTML – документа

Парний тег <BODY> призначається для виділення тієї частини документа, яка буде видима користувачу та має декілька атрибутів. Атрибути визначають загальний вигляд документа та наведені в табл. 5.13.1.

Таблиця 5.13.1

Атрибути елемента <BODY>

Атрибут	Призначення
bgcolor	Визначає колір фону документа. Синтаксис: <Body bgcolor="#ff0000"> або <Body bgcolor="red">
background	Вказує на URL-адресу зображення, що використовується як фон. Синтаксис: <Body background="URL/ім'я файлу">
text	Задає колір тексту на сторінці. За замовчуванням колір тексту є чорним. Синтаксис: <Body text="колір">
link	Визначає колір гіперпосилання. Синтаксис: <Body link="колір">
alink	Задає колір активного гіперпосилання (в момент клацання по ньому мишею). Синтаксис: <Body alink="колір">
vlink	Задає колір відвіданого гіперпосилання. Синтаксис: <Body vlink="колір">
leftmargin	Визначає відступ (у пікселях) від лівого краю вікна браузера до контенту сторінки
rightmargin	Визначає відступ (у пікселях) від правого краю вікна браузера до контенту сторінки
topmargin	Визначає відступ (у пікселях) від верхнього краю вікна браузера до контенту сторінки
bottommargin	Визначає відступ (у пікселях) від нижнього краю вікна браузера до контенту сторінки

Визначення кольору складових частин документа – перший крок до його створення. Якщо це не зроблено, використовуються кольори за замовчуванням (визначаються налаштуваннями браузера).

Потрібно намагатися підтримувати високу контрастність тексту і фону та уникати сусідства областей із близькими відтінками кольорів, для зручності сприймання інформації користувачем.

У HTML колірна система базується на трьох основних кольорах: червоному, синьому й зеленому. Кожний основний колір задається двома розрядами шістнадцяткового числа у межах від 00 до FF, що відповідає діапазону 0 – 255 у десятковому вираженні. Колір можна задавати шестирозрядним шістнадцятковим числом, якому передує символ "#", наприклад, #0000FF – визначає синій колір.

Колір можна також задавати назвою кольору англійською мовою – red, blue і т. д. (рис. 5.13.2).

```
<!doctype html>
<html>
<head>
<title>Тег Body</title>
<meta charset = «utf-8»>
</head>
<body bgcolor="yellow" leftmargin="100" topmargin="50"
rightmargin="50" bottommargin="50" text="blue"
link=" black" alink="red" vlink="green">
Всім привіт!
</body>
</html>
```

Рис. 5.13.2. Приклад застосування атрибутів тегу <BODY>

Атрибут BACKGROUND вказує на зображення, яке буде використуватися як фоновий рисунок. Цей рисунок заповнює собою весь видимий простір вікна. Якщо рисунок є меншим за розмір вікна браузера, то він буде повторюваним, утворюючи мозаїку з однакових картинок. На стику цих картинок виникають видимі переходи. Для усунення цього недоліку доцільно для форматування фонового рисунка використовувати CSS.

Обов'язково після назви рисунка потрібно через крапку вказати розширення рисунка (jpg, png, gif та ін.). Наприклад: <body background="весна.jpg">. Якщо рисунок знаходиться в іншій папці, то необхідно обов'язково вказати ім'я папки. Наприклад: <body background="рисунки/весна.jpg">.

Для форматування тексту існує багато тегів. Деякі з них використовуються часто (вони швидко запам'ятовуються), інші – нечасто (їх і не потрібно запам'ятовувати). Розглянемо ті, які використовуються часто.

Теги поділу тексту на абзаци й переносу рядка

Тег
 – тег примусового переносу рядка. Текст, що розміщений після цього тегу, починається з нового рядка. Тег
 одинарний і не потребує закриття. Можна вставляти підряд декілька тегів
 для переміщення тексту та інших об'єктів вниз сторінки.

Парний тег <P></P> розділює текст на абзаци. Перед початком кожного абзацу варто помістити тег <P>. Кінець абзацу позначається тегом </P>. На відміну від тегу
, абзаци відділяються один від одного порожнім рядком.

У тегу <P> є атрибут ALIGN, що задає спосіб вирівнювання тексту усередині абзацу (рис. 5.13.3).

Код HTML	<pre><!doctype html> <html> <head> <title>Форматування web-сторінки</title> <meta charset = "utf-8"> <meta name = "description" content=""> <meta name = "keywords" content=""> </head> <body> <!--Тег переносу рядка--> В деяких випадках виникає необхідність перенесення тексту
 на інший рядок <!--Тег поділу на абзаци--> <p align="center">Абзац вирівняно по центру</p> <p align="left">Абзац вирівняно по лівому краю</p> <p align="right">Абзац вирівняно по правому краю</p> </body> </html></pre>
Браузер	<p>В деяких випадках виникає необхідність перенесення тексту на інший рядок</p> <p style="text-align: center;">Абзац вирівняно по центру</p> <p>Абзац вирівняно по лівому краю</p> <p style="text-align: right;">Абзац вирівняно по правому краю</p>

Рис. 5.13.3. Перенесення рядка та вирівнювання абзаців

Атрибут ALIGN має параметри:

- left – по лівому краю;
- right – по правому краю;
- center – по центру;
- justify – по ширині.

Слід відмітити, що окремо існує тег `<center></center>`, який вирівнює текст по центру.

Теги форматування заголовків тексту

Для створення заголовків використовують теги `<Hn> </ Hn>`, де *n* – число від 1 до 6.

`<H1></H1>`;

`<H2></H2>`;

`<H3></H3>`;

`<H4></H4>`;

`<H5></H5>`;

`<H6></H6>`.

Ці теги виділяють текст у вигляді заголовків. Кожний заголовок починається з нового рядка, виділяється напівжирним шрифтом і має свій розмір (заголовок першого рівня H1 найбільший, шостого – найменший). Теги заголовків можуть використовуватися з параметром горизонтального вирівнювання ALIGN (рис. 5.13.4).

Код HTML	<pre><!--Теги заголовків--> <h1 align="center">Заголовок 1-го рівня</h1> <h3 align="left">Заголовок 3-го рівня</h3> <h6 align="right">Заголовок 6-го рівня</h6></pre>
Браузер	<p style="text-align: center;">Заголовок 1-го рівня</p> <p>Заголовок 3-го рівня</p> <p style="text-align: right;">Заголовок 6-го рівня</p>

Рис. 5.13.4. Перенос рядка та вирівнювання абзаців

Теги форматування креслення тексту

У табл. 5.13.2 наведено теги, що використовуються для форматування креслення тексту.

Теги форматування креслення тексту

Креслення тексту			
Курсив	Напівжирний	Підкреслений	Закреслений
<code><cite></cite></code>	<code></code>	<code><ins></ins></code>	<code><strike></strike></code>
<code><dfn></dfn></code>	<code></code>	<code><u></u></code>	<code><s></s></code>
<code></code>			
<code><i></i></code>			

Теги, що виділяють текст курсивом, мають різне семантичне навантаження, проте візуально результат відображення в браузері відрізнити дуже важко. На рис. 5.13.5 наведено приклад сумісного використання тегів для форматування тексту.

Код HTML	<pre> <!--Теги форматування креслення тексту--> <p align="left">Для розроблення сайтів використовується мова розмітки гіпертексту <ins>HTML</ins></p> </pre>
Браузер	<p>Для розроблення <i>сайтів</i> використовується мова розмітки гіпертексту <u>HTML</u></p>

Рис. 5.13.5. Перенос рядка та вирівнювання абзаців

Для відображення тексту вище та нижче від основного використовуються теги: `` – відображення вище, `` – відображення нижче. Наприклад, для створення фрази "Я_{студент}^{XHEУ}" необхідно записати такий HTML-код:

Я `_{` студент `}` `^{` XHEУ `}`.

Слід відмітити, що на сьогодні для форматування тексту часто застосовуються стилі CSS.

Тег збереження авторського форматування

Ви, напевно, помітили, що скільки б пробілів ви не зробили у розмітці, на екрані завжди відображається тільки один пробіл. Решта пробілів браузером ігноруються, що не завжди зручно. Наприклад, під час відображення віршів хочеться зберегти всі відступи і пробіли. У цьому випадку і використовується парний тег `<PRE>` `</PRE>` (рис. 5.13.6).

Код HTML	<pre><!--Використання тегу pre--> Ой у гаю, при Дунаю Соловей щебече. Він свою всю пташину До гніздечка кличе. <pre> Ой у гаю, при Дунаю Соловей щебече. Він свою всю пташину До гніздечка кличе. </pre></pre>
Браузер	<pre>Ой у гаю, при Дунаю Соловей щебече. Він свою всю пташину До гніздечка кличе. Ой у гаю, при Дунаю Соловей щебече. Він свою всю пташину До гніздечка кличе.</pre>

Рис. 5.13.6. Застосування тегу `<PRE>`

Створення списків на сторінці

На web-сторінці можуть бути розміщені списки таких типів:

- нумерований – для перерахування елементів, які слідуєть у строго визначеному порядку;
- маркований – для перерахування елементів, які слідуєть у довільному порядку;
- багаторівневий – для деталізації інформації елементів списку різних рівнів;
- списки визначень – для формування словників, а також списків термінів та їхніх визначень.

Для створення *нумерованих* списків використовуються теги `` і ``.

Тегом `` визначається початок/кінець усього списку.

Тегом визначають початок/кінець елемента списку.

За замовчуванням елементи списку нумеруються по порядку 1, 2, 3. За допомогою атрибута TYPE можна змінювати стиль нумерації. Водночас атрибуту TYPE можуть присвоюватися значення "1", "A", "a", "i" або "l".

За допомогою атрибута START можна змінювати порядок нумерації списку, задаючи початковий номер списку, відмінний від 1.

Для не пронумерованих списків можуть використовуватися різні маркери для вказівки на елемент списку.

Маркований список починається тегом і завершується тегом . Кожен елемент списку вказується тегом .

За замовчуванням елементи списку маркуються чорним колом. За допомогою атрибута TYPE можна змінити стиль маркування. Цьому атрибуту можна присвоювати значення "circle", "disk" та "square".

У межах одного списку можна використовувати різне маркування елементів списку.

Для створення *багаторівневих* списків можна використовувати як марковані, так і нумеровані списки, а також їхні поєднання.

Багаторівневий список виходить шляхом вкладення одного списку в інший.

Головне завдання – не заплутатися зі вкладеннями. Щоб не допустити помилок, доцільно робити різні відступи окремих списків (структурованість програмного коду – захист від помилок).

Для створення *списків визначень* (наприклад, словників термінів із їхніми визначеннями) використовують три теги:

<DL></DL> – початок/кінець списку;

<DT></DT> – початок/кінець конкретного терміна;

<DD></DD> – початок/кінець тексту, який пояснює термін (визначення).

Теги <DT> і <DD> не обов'язково чергувати, тобто можна "прив'язати" до одного терміна кілька визначень і навпаки.

На рис. 5.13.7 наведено приклади формування на web-сторінці списків різних типів.

Код HTML	<pre> <!--Створення нумерованого списку--> <OL type="I" start="3"> Три Чотири П'ять <!--Створення маркованого списку--> <UL type="disc"> елемент 1 <li type="circle">елемент 2 <li type="square">елемент 3 <!--Створення багаторівневого списку--> Звірі <li type="disc">Вовк <li type="square">Лисиця Квіти <li type="circle">Троянди <li type="disc">Тюльпани <!--Створення списку визначень--> <DL> <DT>Термін 1</DT> <DD>Анотація1 до терміна 1</DD> <DD>Анотація2 до терміна 1</DD> </DL> </pre>
Браузер	<p>III. Три IV. Чотири V. П'ять</p> <ul style="list-style-type: none"> • елемент 1 ◦ елемент 2 ▪ елемент 3 <p>1. Звірі</p> <ul style="list-style-type: none"> • Вовк ▪ Лисиця <p>2. Квіти</p> <ul style="list-style-type: none"> ◦ Троянди • Тюльпани <p>Термін 1</p> <p>Анотація1 до терміна 1 Анотація2 до терміна 1</p>

Рис. 5.13.7. Формування на web-сторінці списків різних типів

Створення рухомого рядка

Для створення рухомого рядка використовується тег <MARQUEE>.

Встановлювати порядок руху дозволяє атрибут BEHAVIOR (поведінка) з такими значеннями:

scroll – рядок з'явиться з одного боку екрана і зникне з іншого;

slide – рядок з'явиться з одного боку екрана і зупиниться з іншого;

alternate – забезпечує переміщення рядка туди-назад (вліво-вправо).

Інші атрибути:

DIRECTION – задає напрямок переміщення рядка;

LOOP – визначає кількість циклів переміщення;

BGCOLOR – дозволяє змінювати колір фону рухомого рядка;

SCROLLAMOUNT – задає крок переміщення рядка в екранних пікселях;

SCROLLDELAY – вказує час затримки в мілісекундах між окремими тактами переміщення рядка.

Вставка рисунків на сторінку

Для вставки рисунків у HTML-документ використовують тег з такими атрибутами:

SRC="адреса до рисунка";

BORDER="Товщина рамки";

ALIGN="Вирівнювання";

WIDTH="Ширина";

HEIGHT="Висота";

HSPACE="Відступ по горизонталі";

VSPACE="Відступ по вертикалі";

ALT="Альтернативний текст при відсутності зображень";

TITLE="Підказка (назва рисунка під час наведення миші)".

Атрибути, що визначають розмір і відступи рисунка, задаються в пікселях.

Вимоги до формування значення атрибута SRC аналогічні вимогам до атрибута BACKGROUND тегу <BODY>.

Для вирівнювання рисунка по центру доцільно використовувати тег <CENTER></CENTER>.

Приклад вставки рисунка на сторінку наведено на рис. 5.13.8.

Код HTML	<pre> <!--Вставка малюнків на сторінку--> <center> </center> </pre>
Браузер	

Рис. 5.13.8. Вставка рисунків на сторінку

Якщо рисунок вирівняний по лівому краю, то текст (інші об'єкти) будуть обтікати його з правої сторони. Під час вирівнювання рисунка по правому краю текст буде розміщуватися з лівої сторони від рисунка. За наявності 2-х рисунків, один із яких вирівняний по лівому краю, а другий по правому – текст буде розташовуватися посередині між рисунками.

Вставка гіперпосилання на сторінку

Гіпертекстові посилання HTML дозволяють зв'язати фрагмент тексту або інший об'єкт (наприклад, рисунок) з іншими гіпертекстовими документами. Для його створення використовується тег <A> (від слова anchor – якір) з атрибутом HREF, значення якого – це адреса переходу під час активізації цього гіперпосилання.

Конструкція гіперпосилання така:

```

<a href="url" target="Вкладка" title="Підказка">
    Назва гіперпосилання
</a>

```

За замовчуванням, під час переходу за посиланням документ відкривається в поточній вкладці. За необхідності, ця умова може бути змінена атрибутом TARGET тегу <A>:

`_SELF` – завантажує сторінку в поточну вкладку вікна браузера (значення за замовчуванням);

`_BLANK` – завантажує сторінку в нову вкладку браузера.

Якщо значення `target="_blank"` задати для тегу <BASE>, то всі гіперпосилання на сторінці будуть відображатися в новій вкладці. Наприклад: `<head> <base target="_blank"></head>`.

Для організації посилання на закладку в поточному документі використовується така конструкція:

`Назва розділу`

Ім'я закладки вставляється в необхідному місці на сторінці (куди має бути здійснений перехід) і організовується таким чином:

``

Для створення посилання на закладку в іншому документі необхідно додати ім'я файлу. Наприклад:

` Біографія Т. Г. Шевченко`

На рис. 5.13.9 наведено приклади створення гіперпосилань.

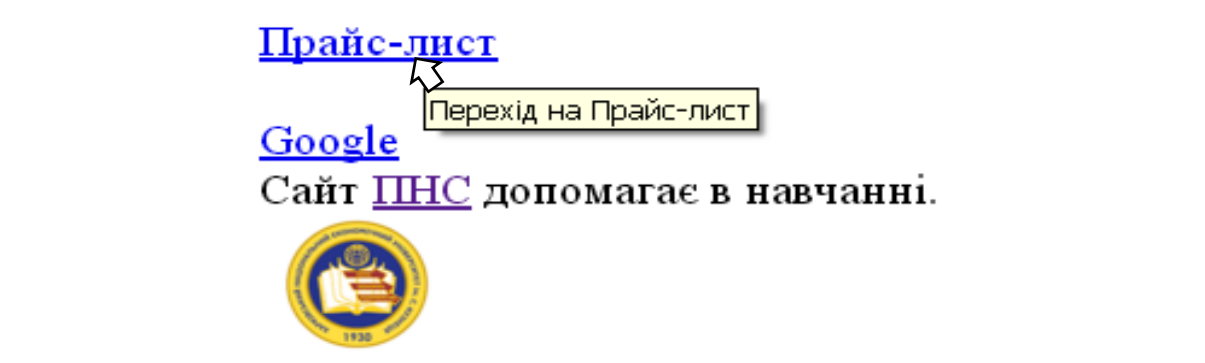
Код HTML	<pre> <!--Гіперпосилання на іншу сторінку--> Прайс-лист
 <!--Гіперпосилання на web-ресурс--> Google
 <!--Гіперпосилання на web-ресурс всередині тексту--> Сайт ПНС допомагає в навчанні.
 <!--Рисунок як гіперпосилання--> </pre>
Браузер	

Рис. 5.13.9. Приклади гіперпосилань

Вставка ліній на сторінку

У деяких випадках виникає необхідність розділення сторінки лініями на складові.

Тег <HR> рисує горизонтальну лінію, відокремлену зверху і знизу порожніми рядками. У цього тегу є такі атрибути:

ALIGN – вирівнює лінію по центру (center), ліворуч (left) або праворуч (right);

WIDTH – встановлює довжину лінії в пікселях або у відсотках від ширини вікна;

SIZE – встановлює товщину лінії в пікселях;

COLOR – встановлює колір лінії;

NOSHADE – прибирає рельєфність лінії.

Без задання атрибутів тегу <HR> лінія буде сірого кольору і займатиме всю ширину вікна браузера.

Приклад використання тегу <HR> наведено на рис. 5.13.10.

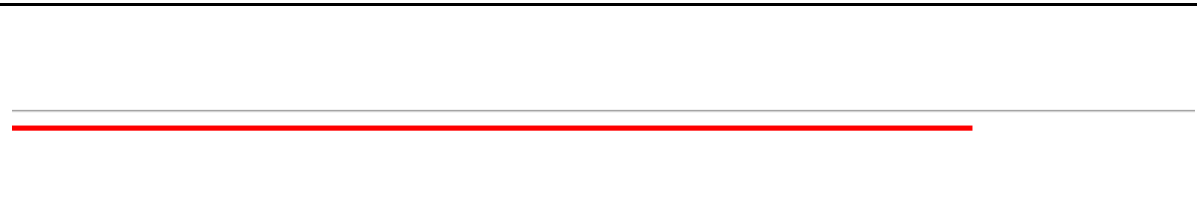
Код HTML	<pre><!--Вставка ліній--> <hr> <hr color="red" width="70%" align="left" size="3"></pre>
Браузер	 The image shows a browser window with a white background. A single horizontal red line is positioned in the lower half of the window, extending across most of its width. The line is solid and has a consistent thickness.

Рис. 5.13.10. Вставка ліній на сторінку

Слід відмітити, що атрибут WIDTH тегу <HR> доцільно задавати у відсотках. Це дозволить масштабувати довжину лінії відповідно до розміру вікна браузера.

Практичне виконання роботи

Завдання 13.1. Знайомство з текстовим редактором Notepad++

Створіть у редакторі Notepad++ новий файл *форматування.html* та наберіть в ньому код простої web-сторінки.


Вирішення

Для написання коду використовуються текстові редактори. Найбільш відомими є: Блокнот, Notepad++, Sublime Text, WebStorm та інші.

У цій роботі скористаємося редактором Notepad++. Це популярний безкоштовний редактор для текстових файлів, який є чудовою альтернативою вбудованому стандартному блокноту. Зазвичай цю програму використовують не тільки звичайні користувачі, але і багато web-майстрів.

Завдання 13.2. Створення HTML-файлу в редакторі Notepad++

Створіть на робочому столі папку **WEB**. У цю папку будемо зберігати HTML-файли, рисунки, аудіо- і відеофайли.

Двічі натисніть на піктограму . У вікні редактора Notepad++ задайте мову сторінки (вкладка "Мова" → **H** → **HTML**) і тип кодування (вкладка "Кодування" → **UTF-8**) (рис. 5.13.11).

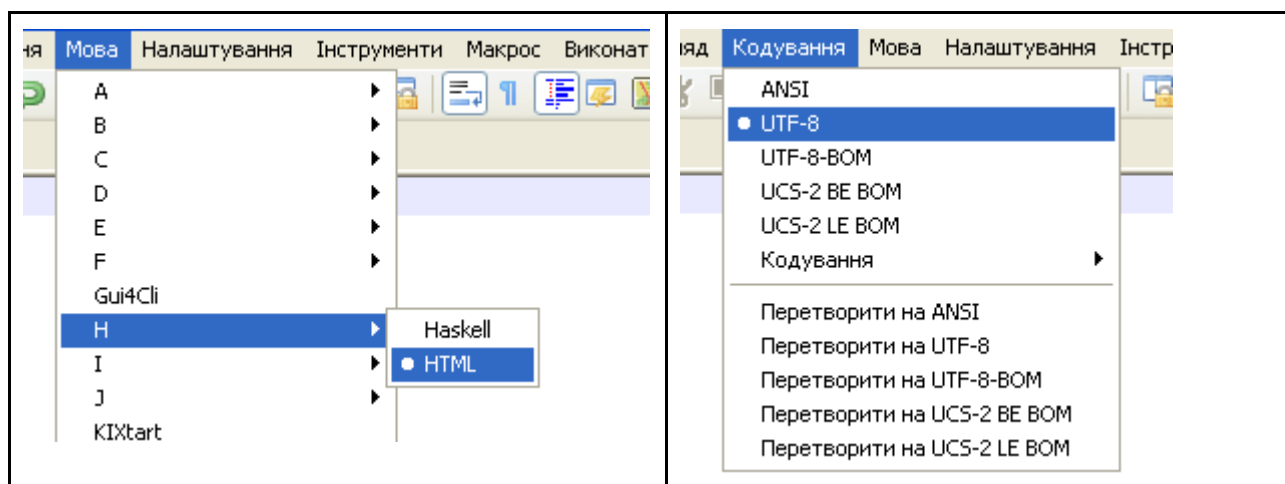


Рис. 5.13.11. Створення HTML-файлу в редакторі Notepad++

Завдання 13.3. Набір коду web-сторінки

Редактор Notepad++ має службову розмітку, можливість автозавершення та ін., що суттєво відрізняє його в кращий бік, порівнюючи з Блокнотом.

Колір службової розмітки Notepad++ за замовчуванням:

теги – синій;

атрибути – червоний;

значення атрибутів – фіолетовий;

авторський текст – чорний;

коментар – зелений.

Наберіть код web-сторінки, як показано на рис. 5.13.11.

Завдання 13.4. Збереження HTML-файлу

HTML-файли мають розширення **.html**. Для збереження файлу натисніть вкладку **"Файл"→Зберегти як...** У вікні, що відкрилося, виберіть із випадного списку тип файлу (**Тип→All types (*.*)**), призначте файлу ім'я *Форматування.html* та збережіть у папку **WEB** (рис. 5.13.12).

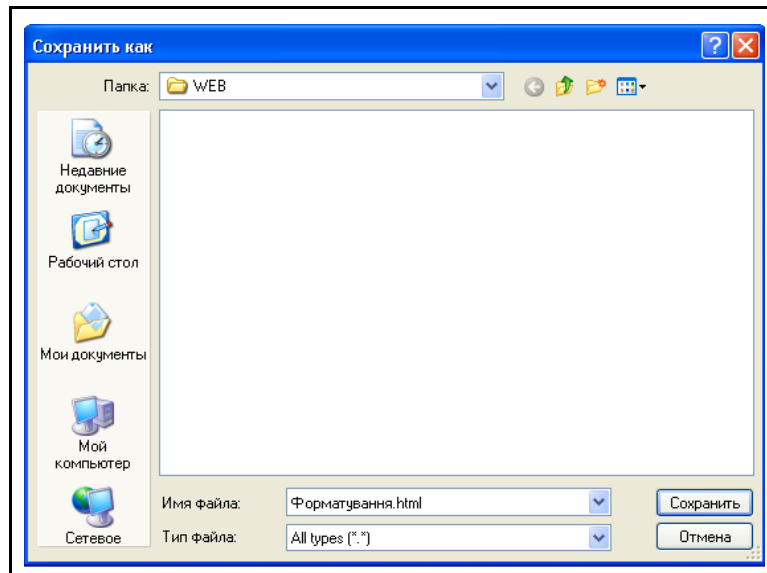


Рис. 5.13.12. Збереження HTML-файлу

За правильного збереження HTML-файл буде відобразитися зі значком браузера, що встановлений на комп'ютері за замовчуванням (рис. 5.13.13).

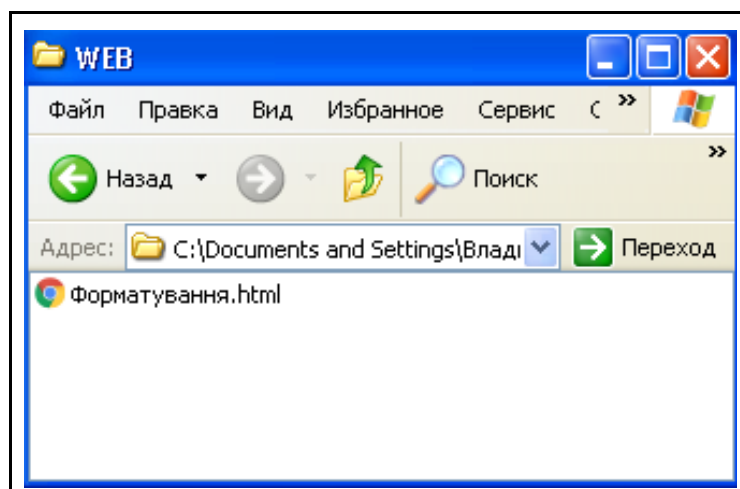


Рис. 5.13.13. Вигляд HTML-файлу після збереження

Для подальшого збереження вмісту файлу використовуйте поєднання клавіш "**Ctrl+S**".

Завдання 13.5. Відкриття HTML-файлу в браузері

Відкрийте файл у браузері (вкладка "**Виконати**"→**Launch in Chrome**) (рис. 5.13.14).

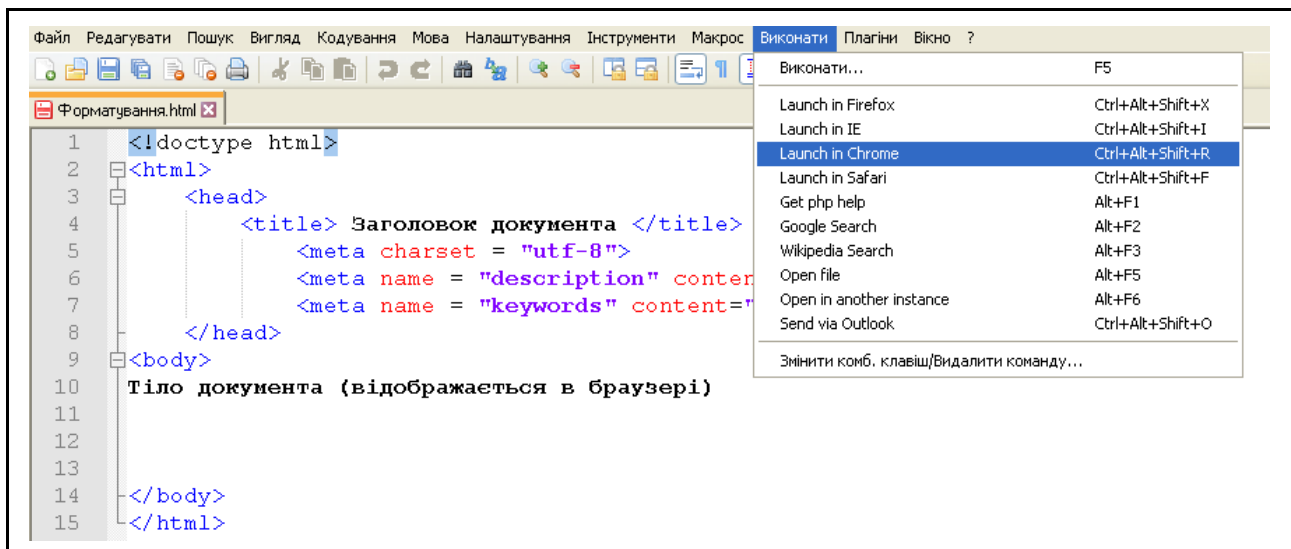


Рис. 5.13.14. Відкриття HTML-файлу в браузері

Файл для перегляду можна також відкрити шляхом натискання правої кнопки миші на HTML-файлі, розміщеному в папці (**Відкрити за допомогою...**).

Проаналізуйте зміст того, що відображається (це вміст контейнера **BODY**). Зверніть увагу на назву вкладки браузера (це вміст контейнера **TITLE**). Змініть вміст контейнерів **BODY** та **TITLE**, збережіть файл (**Ctrl+S**) і перевірте вміст сторінки, що відображається в браузері. Для цього за умови відкритого браузера натисніть клавішу **F5** або кнопку "**Оновити цю сторінку**" на панелі браузера.

Завдання 13.6. Форматування web-сторінок

Використовуючи раніше наведені приклади (рис. 5.13.3 – 5.13.10), послідовно (шляхом накопичення) відпрацюйте особливості використання тегів для форматування web-сторінок.

Індивідуальні завдання

Створіть web-сторінку про улюбленого письменника (актора, спортсмена), на якій відобразіть такі матеріали:

- прізвище, ім'я, по батькові (використовуйте теги заголовків та форматування шрифту);
- декілька фотографій різних періодів життя (одна із фотографій має бути гіперпосиланням на Вікіпедію);
- основні етапи життя та досягнення (використовуйте різні типи списків та роздільні лінії);
- створіть гіперпосилання на web-ресурси та всередині сторінки;
- заповніть сторінку повністю;
- для форматування web-сторінки використовуйте тільки засоби HTML.

Запитання для самоконтролю

1. Що таке HTML?
2. Що таке "тег"? У чому відмінність одиночних і парних тегів?
3. Що таке "атрибут"? Наведіть приклад.
4. Які частини HTML-документа ви знаєте?
5. Які типи гіперпосилань ви знаєте?
6. У чому відмінність фізичного й логічного форматування?
7. Які текстові редактори ви знаєте? У чому їхня відмінність?
8. Як перетворити рисунок у гіперпосилання?
9. Чому довжину ліній доцільно задавати у відсотках

14. Таблична верстка сайту. Створення форм на web-сторінці

Мета: вивчити порядок роботи з таблицями в HTML. Створити сайт за допомогою табличної верстки. Познайтися з технологією створення форм на сторінці.

Ключові терміни: таблиця, об'єднання клітинок, таблична верстка, форма, елементи управління.

Теоретичні відомості

За допомогою таблиць зручно верстати макети сторінок, розташувати потрібним чином фрагменти тексту та зображення.

Таблиці в HTML створюються рядок за рядком. У кожному рядку задається певна кількість клітинок.

Для створення таблиці застосовують парний тег `<TABLE>`, `</TABLE>`.

Тег `<CAPTION>` призначений для створення заголовка до таблиці та розміщується всередині контейнера `<TABLE>` відразу після відкриваючого тегу. Такий заголовок становить текст, за замовчуванням відображається перед таблицею і описує її зміст.

Для створення рядків застосовують контейнер `<TR>` та `</TR>`.

Для створення клітинок у рядках використовуються парні теги `<TH>` або `<TD>`.

Тег `<TH>` використовується для створення клітинок у рядку заголовка, в яких дані відображаються напівжирним шрифтом із вирівнюванням по центру, а тег `<TD>` – для створення звичайних клітинок таблиці.

Тег `<TABLE>` може мати такі атрибути:

`BORDER` – для створення видимої межі таблиці із зазначенням товщини зовнішньої лінії;

`WIDTH` – ширина таблиці в пікселях або відсотках. З метою масштабування ширини таблиці під час зміни розміру вікна браузера значення цього тегу доцільно задавати у відсотках;

`ALIGN` – вирівнювання таблиці по горизонталі. Набуває значень: `left`, `right`, `center`;

`CELLSPACING` – задає відстань між клітинками таблиці (у пікселях), `CELLPADDING` – задає відстань між текстом і внутрішньою межею клітинки таблиці (у пікселях).

Для зміни кольору межі таблиці застосовується атрибут BORDER-COLOR із зазначенням кольору ліній межі. Крім того, для таблиці можна задавати колір фону або фоновий рисунок (атрибути BGCOLOR, BACKGROUND).

Теги <TR> та <TD> мають такі атрибути:

ALIGN – вирівнює дані по горизонталі (набуває значень: left, center, right);

VALIGN – вирівнює дані по вертикалі (набуває значень: top, center, bottom).

Для кожного рядка (клітинки) можна окремо задати колір фону або фоновий рисунок.

За допомогою атрибутів COLSPAN і ROWSPAN тегів <TH> або <TD> можна об'єднувати кілька сусідніх клітинок стовпчика або рядка в одну клітинку.

Атрибут COLSPAN забезпечує об'єднання клітинок кількох стовпців у межах одного рядка, а атрибут ROWSPAN – об'єднання клітинок кількох рядків у межах одного стовпчика. Значення атрибутів задається цифрою, яка показує кількість клітинок, які необхідно об'єднати.

Формування зовнішньої межі таблиці

Атрибут FRAME – задає вигляд межі навколо таблиці та може набувати таких значень:

void – межі немає;

above – тільки верхня межа;

below – тільки нижня межа;

hsides – тільки верхня і нижня межі;

vsides – тільки ліва і права межі;

lhs – тільки ліва межа;

rhs – тільки права межа;

box – усі чотири частини межі (за замовчуванням).

Формування внутрішніх меж таблиці

Атрибут RULES задає вигляд внутрішніх меж таблиці та може набувати таких значень:

void – межі немає;

above – тільки верхня межа;

below – тільки нижня межа;

hsides – тільки верхня і нижня межі;
vsides – тільки ліва і права межі;
lhs – тільки ліва межа;
rhs – тільки права межа;
box – усі чотири частини межі (за замовчуванням).

На сьогодні таблиці рідко використовуються для верстання сайтів, тому що мають жорстку структуру і не дозволяють створювати адаптивні сайти.

Проте відмова від таблиць є недоцільною, оскільки використовувати таблиці дуже зручно під час відображення чисел та інших табличних даних. Крім того, використання таблиць дозволяє наочно і компактно створювати форми на web-сторінці.

Форми призначені для отримання даних від користувача із подальшим пересиланням їх на сервер, де ці дані обробляються спеціальною програмою. Для створення форм на web-сторінці використовується мова HTML, а для оброблення даних на формі використовуються мови web-програмування PHP, Javascript та ін.

Для вставки форми використовується парний тег `<FORM>` і `</FORM>`, між якими розміщуються різні елементи управління.

Тег `<FORM>` має такі атрибути:

NAME – ім'я форми. Необхідно, якщо на сторінці кілька форм;

ACTION – визначає URL-адресу, за якою буде відправлена інформація, введена користувачем;

METHOD – визначає спосіб відправки інформації. Може мати одне з двох значень:

- get – передає дані серверу як частину URL-адреси;
- post – передає дані, використовуючи тіло http-запиту (передані дані не видно).

Основні елементи форми

Для створення елементів управління використовується тег `<INPUT>` з атрибутом TYPE – тип елемента управління, який має такі значення:

text – текстове поле (за замовчуванням);

password – текстове поле для введення пароля;

search – текстове поле для організації пошуку по сайту;

checkbox – прапорець;

radio – перемикач;

submit, image, reset, button – види кнопок;

file – поле для файлів.

Крім того, на формі можуть створюватися інші елементи:

<TEXTAREA> </ TEXTAREA> – багаторядкове текстове поле;

<SELECT> </ SELECT> – списки, що розкриваються;

<FIELDSET> </ FIELDSET> – групування елементів форми;

<LABEL> </ LABEL> – встановлення зв'язку між текстом і елементом форми.

Текстове поле

Для створення текстового поля використовується тег <INPUT> з такими атрибутами:

TYPE – тип елемента (type="text");

NAME – ім'я елемента;

SIZE – розмір текстового поля в символах (кількість одночасно відображуваних символів);

MAXLENGTH – максимальна кількість символів (за відсутності цього атрибута кількість символів буде необмеженою);

VALUE – текст за замовчуванням (за відсутності цього атрибута поле буде порожнім);

DISABLED – блокує поле від змін;

READONLY – поле використовується тільки для читання.

Замість атрибута VALUE можна використовувати атрибут PLACEHOLDER. У цьому випадку під час введення тексту вихідний текст пропадає.

Багаторядкове текстове поле використовується для великих текстів (повідомлень).

Створюється парним тегом <TEXTAREA></TEXTAREA> з такими атрибутами:

NAME – ім'я поля;

COLS – ширина поля в символах;

ROWS – кількість рядків тексту, видимих на екрані;

WRAP – спосіб перенесення слів:

- off – перенесення не відбувається;
- virtual – перенесення відображається, але на сервер надходить неподільний рядок;
- physical – перенесення і на екрані, і під час надходження на сервер;

DISABLED – блокує поле від змін;
READONLY – поле є тільки для читання.

Зв'язок тексту та елементів форми

Тег `<LABEL></LABEL>` встановлює зв'язок між певною міткою (текстом) і елементом форми (`<INPUT>`, `<SELECT>`, `<TEXTAREA>`). Такий зв'язок необхідний, щоб змінювати значення елементів форми під час натискання курсором миші на текст.

Існують два способи зв'язування об'єкта і мітки.

Перший спосіб полягає у використанні ідентифікатора *id* всередині елемента форми і надання його імені як значення атрибута *for* тегу `<LABEL>`.

Приклад 1:

```
<Label for="T1">Текстове поле: </Label>
```

```
<input type="text" name="text1" size="20" maxlength="50" id="T1">
```

Другий спосіб полягає в розміщенні елемента форми всередині контейнера `<LABEL> </LABEL>`.

Приклад 2:

```
<Label> Текстове поле: <input type="text" name="text1" size="20" maxlength="50"></Label>
```

Установка прапорців

Для створення прапорців використовується тег `<INPUT>` з такими атрибутами:

TYPE – тип елемента (`type="checkbox"`);

NAME – ім'я елемента;

VALUE – вказує програмі, що буде обробляти форму, значення пункту, який вибрав користувач;

CHECKED – найбільш імовірні значення.

Установка перемикачів

На відміну від прапорців, за допомогою перемикачів можна вибрати тільки один пункт. У зв'язку з цим значення параметра NAME мають бути однакові для всіх елементів групи.

Для створення перемикачів використовується тег `<INPUT>` із такими атрибутами:

TYPE – тип елемента (`type="radio"`);

NAME – ім'я елемента;

VALUE – вказує програмі, що буде обробляти форму, значення пункту, який вибрав користувач;

CHECKED – найбільш імовірні значення.

Кнопки

Існує чотири види кнопок:

SUBMIT – кнопка відправки вмісту форми web-сервера. Її атрибути:

TYPE – тип елемента (type="submit");

NAME – ім'я кнопки;

VALUE – напис на кнопці.

RESET – кнопка, що дозволяє відновити всі значення за замовчуванням на формі. Її атрибути:

TYPE – тип елемента (type="reset");

NAME – ім'я кнопки;

VALUE – напис на кнопці.

IMAGE – графічна кнопка відправки вмісту форми web-сервера.

Її атрибути:

TYPE – тип елемента (type="image");

NAME – ім'я кнопки;

SRC – адрес картинки для кнопки.

BUTTON – довільна кнопка, її дії призначаються користувачами.

Її атрибути:

TYPE – тип елемента (type="button");

NAME – ім'я кнопки;

VALUE – напис на кнопці;

ONCLICK – вказує дії під час натискання на кнопку.

Для створення кнопок можна використовувати тег `<BUTTON>` `</BUTTON>`. Він має такі атрибути:

TYPE – тип кнопки (набуває значень: submit, reset, button);

NAME – ім'я кнопки;

VALUE – напис на кнопці.

Додаткові можливості:

- розміщення на кнопці рисунків;
- використовуючи CSS, змінювати вигляд, колір, форму кнопки;
- кнопка може перебувати окремо від форми (Javascript).

Список із прокруткою

У разі необхідності на формі можна розмістити список із прокруткою або "випадний список". Для створення списку, що розкривається, використовується тег `<SELECT> </SELECT>`, усередині якого розташовуються елементи значень, які задаються тегом `<OPTION> </OPTION>`.

Атрибути тегу `<SELECT>`:

NAME – ім'я списку. Кожен обраний елемент списку під час передачі на сервер буде мати вигляд: `name.value`, де значення (`value`) береться з тегу `<OPTION>`;

SIZE – визначає кількість видимих елементів у списку:

- 1 – простий список, що розкривається;
- більше 1 – список зі смугою прокрутки;

MULTIPLE – дозволяє вибрати декілька елементів списку.

Атрибути тегу `<OPTION>`:

SELECTED – найбільш імовірний для вибору елемент списку;

VALUE – значення, яке буде відправлено на сервер, якщо пункт вибраний.

Групування елементів форми

Тег `<FIELDSET></FIELDSET>` призначений для групування елементів форми. Таке групування полегшує роботу з формами, що містять велику кількість елементів управління.

Браузери для підвищення наочності відображають результат використання тегу `<FIELDSET>` у вигляді рамки.

Для задавання назви групи елементів використовується тег `<LEGEND></LEGEND>`.

Практичне виконання роботи

Завдання 14.1. Створення таблиці на web-сторінці

Створіть web-сторінку, на якій буде представлена таблиця 3x3. У клітинках таблиці розмістіть рисунки, гіперпосилання, а також для декількох клітинок задайте фоновий колір та фонове зображення.

Вирішення

1. У редакторі Notepad++ створіть файл *таблиці.html* та наберіть у ньому код таблиці 3x3 (рис. 5.14.1).

Код HTML	<pre> <table border="1" align="center" width="90%" cellspacing="0" cellpadding="10"> <tr> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td>4</td> <td>5</td> <td>6</td> </tr> <tr> <td>7</td> <td>8</td> <td>9</td> </tr> </table> </pre>									
Браузер	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td>4</td> <td>5</td> <td>6</td> </tr> <tr> <td>7</td> <td>8</td> <td>9</td> </tr> </table>	1	2	3	4	5	6	7	8	9
1	2	3								
4	5	6								
7	8	9								

Рис. 5.14.1. Створення таблиці на web-сторінці

2. Зробіть копію таблиці та вставте нижче на сторінці. Для розділення таблиць між собою використайте декілька тегів
. У другій таблиці замість цифр вставте в клітинки таблиці рисунки, гіперпосилання, а також для декількох клітинок задайте фоновий колір та фонове зображення (рис. 5.14.2 та 5.14.3).

Код HTML	<pre> <table border="1" align="center" width="90%" cellspacing="0" cellpadding="10"> <tr> <td><h2>Google</h2></td> <td bgcolor="blue"></td> <td align="center"></td> </tr> <tr> <td bgcolor="yellow"></td> <td align="center"></td> <td bgcolor="pink"></td> </tr> <tr> <td align="center"></td> <td background="весна.jpg"></td> <td><h2>Сайт ПНС</h2></td> </tr> </table> </pre>
----------	--

Рис. 5.14.2. Наповнення клітинок таблиці (HTML-код)



Рис. 5.14.3. Наповнення клітинок таблиці (вигляд у браузері)

Завдання 14.2. Об'єднання клітинок у таблиці

Створіть web-сторінку, на якій буде представлена таблиця 5x5. Використовуючи атрибути COLSPAN і ROWSPAN тегів <TH>, <TD>, створіть макет web-сторінки.

Вирішення

1. Зробіть копію таблиці, представлену на рис. 5.14.1, і доповніть HTML-код для формування таблиці 5x5 (рис. 5.14.4). Доцільно використовувати тимчасову нумерацію клітинок, це дозволить швидше орієнтуватися у кодї сторінки.

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

Рис. 5.14.4. Вигляд таблиці 5x5

2. Необхідно об'єднати всі клітинки у першому рядку.

Для цього тегу <TD>, який формує клітинку 11, задаємо атрибут COLSPAN і вказуємо кількість клітинок, які потрібно об'єднати. Наприклад: <td colspan="5">11</td>.

Під час перегляду в браузері видно, що клітинки 12, 13, 14 та 15 стали зайві, оскільки об'єдналися у клітинку 11. Тому теги <TD>, що формують

ці клітинки, потрібно видалити. Таким чином, HTML-код для формування першого рядка таблиці буде мати такий вигляд:

```
<tr><td colspan="5">11</td></tr>.
```

Для дослідження властивостей тегу <TH> замініть у першому рядку тег <TD> на <TH>, а цифру 11 на фразу "Таблиця №1":

```
<tr><th colspan="5">Таблиця №1</th></tr>.
```

Під час відображення у браузері фраза "Таблиця №1" буде відображена жирним кресленням і вирівняна по центру.

3. Об'єднайте всі клітинки у першому стовпчику, починаючи з клітинки 21. Для цього тегу <TD>, який формує клітинку 21, задаємо атрибут ROWSPAN і вказуємо кількість клітинок, які потрібно об'єднати. Наприклад: <td rowspan="4">21</td>.

Переглядаємо у браузері та видаляємо зайві клітинки в кожному із рядків. У цьому випадку, це клітинки: 35, 45, 55.

4. Об'єднайте клітинки всередині таблиці: 32, 33, 42, 43.

Для цього тегу <TD>, який формує клітинку 32, необхідно одночасно задати атрибути COLSPAN та ROWSPAN і вказати кількість клітинок, які потрібно об'єднати.

Наприклад: <td colspan="2" rowspan="2">32</td>. Переглядаємо у браузері та видаляємо зайві клітинки в кожному із рядків. У цьому випадку, це клітинки: 34, 43, 44.

5. Вставте в об'єднану клітинку 32 таблицю 3x3, що наведена на рис. 5.14.1. Оскільки вміст клітинки визначається текстом, що знаходиться між відкритим тегом <TD> і закритим тегом </TD>, то замість цифри 32 вставимо HTML-код таблиці 3x3.

Підсумковий HTML-код завдання 14.2 і зображення таблиці в браузері подано на рис. 5.14.5 та 5.14.6.

Тепер замість цифр можна наповнювати таблиці текстом, рисунками, гіперпосиланнями та ін. Водночас межі таблиць можна прибрати повністю (атрибут BORDER тегу <TABLE>) або частково (атрибути FRAME і RULES тегу <TABLE>).

Код HTML

```
<table border="1" align="center" width="90%" cellspacing="0" cellpadding="10">
  <tr>
    <th colspan="5">Таблиця №1</th>
  </tr>
  <tr>
    <td rowspan="4">21</td>
    <td>22</td>
    <td>23</td>
    <td>24</td>
    <td>25</td>
  </tr>
  <tr>
    <td>31</td>
    <td colspan="2" rowspan="2">
      <table border="1" align="center" width="90%" cellspacing="0" cellpadding="10">
        <tr><td>1</td><td>2</td><td>3</td></tr>
        <tr><td>4</td><td>5</td><td>6</td></tr>
        <tr><td>7</td><td>8</td><td>9</td></tr>
      </table>
    </td>
    <td>33</td>
  </tr>
  <tr>
    <td>41</td>
    <td>42</td>
  </tr>
  <tr>
    <td>51</td>
    <td>52</td>
    <td>53</td>
    <td>54</td>
  </tr>
</table>
```

Рис. 5.14.5. Об'єднання клітинок у таблиці (HTML-код)

Браузер

Таблиця №1				
21	22	23	24	25
	31	1 2 3		33
	41	4 5 6		42
	51	7 8 9		54

Рис. 5.14.6. Об'єднання клітинок у таблиці (вигляд у браузері)

Завдання 14.3. Створення форми на web-сторінці

Створіть web-сторінку, на якій послідовно розмістіть елементи форми.

Завдання 14.3.1. Створення текстових полів на формі

Створіть на формі текстові поля різних типів і зв'яжіть назви текстових полів з елементами форми.

Вирішення

1. У редакторі Notepad++ створіть файл *форми.html* та наберіть у ньому код для формування текстових полів різних типів. Використайте тег `<FIELDSET></FIELDSET>` для групування елементів форми, а також тег `<LABEL></LABEL>` для встановлення зв'язку між певним текстом і елементом форми (рис. 5.14.7 та 5.14.8).

Код HTML	<pre><!DOCTYPE HTML> <HTML> <HEAD> <TITLE>Приклад створення форми</TITLE> <meta charset="UTF-8"> <meta name = "description" content=" "> <meta name = "keywords" content=" "> </HEAD> <BODY BGCOLOR="lightblue"> <FORM action=" " method="post" name="form1"> <fieldset> <legend><H2>Приклад форми</H2></legend> <!--Просте текстове поле--> <Label for="T1">Текстове поле: </Label> <input type="text" name="text1" size="20" maxlength="50" value="Введіть текст" id="T1">

<!--Текстове поле не активне--> <input type="text" name="text1" size="20" maxlength="50" value="не активне поле" disabled>

<!--Текстове поле тільки для читання--> <input type="text" name="text1" size="20" maxlength="50" value="тільки для читання" readonly>

<!--Текстове поле для для пошуку по сайту--> <input type="search" name="q" placeholder="Пошук по сайту"> <input type="submit" value="Знайти">

<!--Багаторядкове текстове поле--> <Label> Багаторядкове текстове поле:
 <textarea cols="30" rows="5" name="areal" ></textarea></Label>
<!--Текстове поле для введення паролю--> Введіть пароль:
 <INPUT type="password" name="pass1" size="20"> <input type="submit" value="Ввести пароль"> </fieldset> </form> </BODY> </HTML></pre>
----------	--

Рис. 5.14.7. Створення текстових полів на формі (HTML-код)

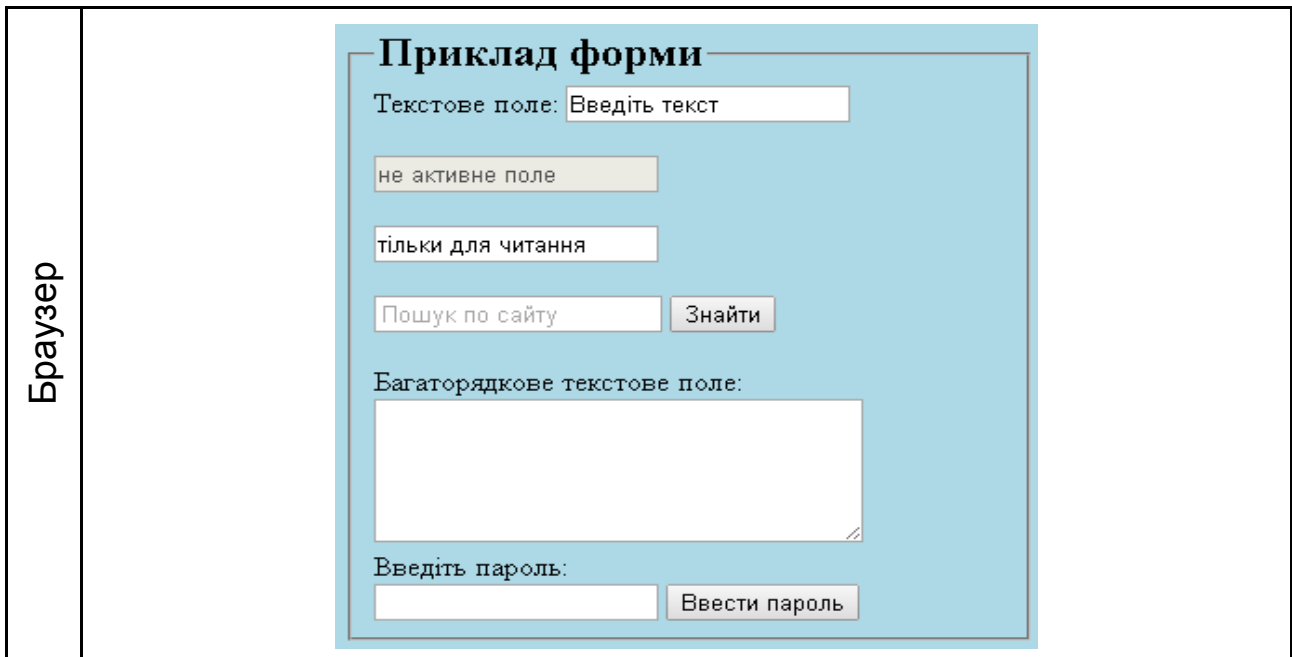


Рис. 5.14.8. Створення текстових полів на формі

Завдання 14.3.2. Установка прапорців на формі

Створіть на формі прапорці для визначення однієї із чотирьох мов: української, англійської, німецької та іспанської.

Вирішення

1. Додайте у файл *форми.html* HTML-код для формування прапорців на формі (рис. 5.14.9).

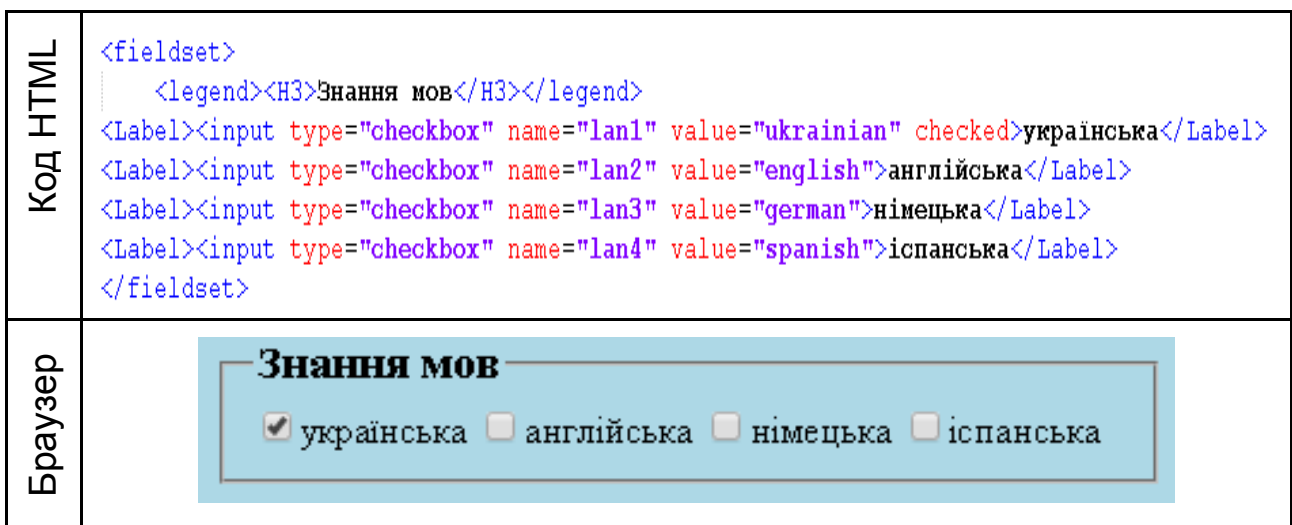


Рис. 5.14.9. Створення прапорців на формі

Завдання 14.3.3. Установка перемикачів на формі

Створіть на формі перемикачі для визначення статі користувача.

Вирішення

1. Додайте у файл *форми.html* HTML-код для формування перемикачів на формі (рис. 5.14.10). Зверніть увагу, що в цьому випадку атрибут NAME має однакове значення для різних елементів. Це дозволяє відправити на сервер тільки одне вибране значення.

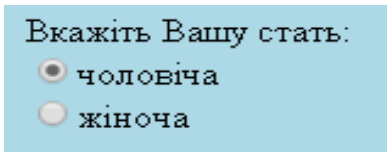
Код HTML	<pre>Вкажіть Вашу стать:
 <input type="radio" name="st" value="man" checked>чоловіча
 <input type="radio" name="st" value="woman">жіноча</pre>
Браузер	

Рис. 5.14.10. Створення перемикачів на формі

Завдання 14.3.4. Створення кнопок на формі

Створіть на формі кнопки різного типу.

Вирішення

1. Додайте у файл *форми.html* HTML-код для створення кнопок на формі (рис. 5.14.11 та 5.14.12).

Код HTML	<pre><!--Кнопка для відправки даних--> <INPUT type="submit" name="s1" value="Віправка даних">
 <!--Малюнок для відправки даних--> Клік по даному малюнку приведе до відправки даних: <INPUT type="image" src="смайлик.png" name="img1" width="50" height="40">
 <!--Кнопка для очистки форми-->
<INPUT type="reset" name="reset" value="Очистка полів">
 <!--Довільна кнопка-->
<INPUT type="button" name="open" onClick='location.href="https://pns.hneu.edu.ua/"' value="ПНС"> <!--Кнопка з малюнком-->

<button type="submit"></button></pre>
----------	--

Рис. 5.14.11. Створення кнопок на формі (HTML-код)

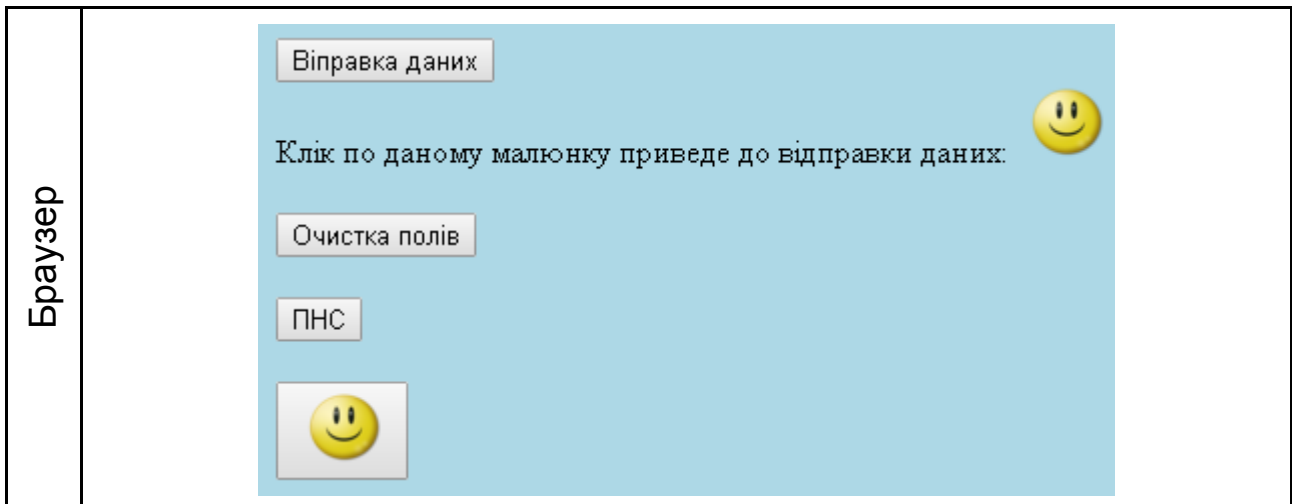


Рис. 5.14.12. Створення кнопок на формі (вигляд у браузері)

Завдання 14.3.5. Створення на формі списку, що розкривається

Створіть на формі список, що розкривається.

Вирішення

Додайте у файл *форми.html* HTML-код для створення на формі списку, що розкривається (рис. 5.14.13 та 5.14.14).

Код HTML	<pre> Які дні тижня для занять Вас влаштують:
 (вибирайте натиснувши клавішу Ctrl)
 <select name="day" size="7" multiple> <option value="mon" selected>понеділок</option> <option value="tue">вівторок</option> <option value="wen">середа</option> <option value="thu" selected>четвер</option> <option value="fri">п'ятниця</option> <option value="sat">субота</option> <option value="san">неділя</option> </select> </pre>
----------	--

Рис. 5.14.13. Створення списку, що розкривається (HTML-код)

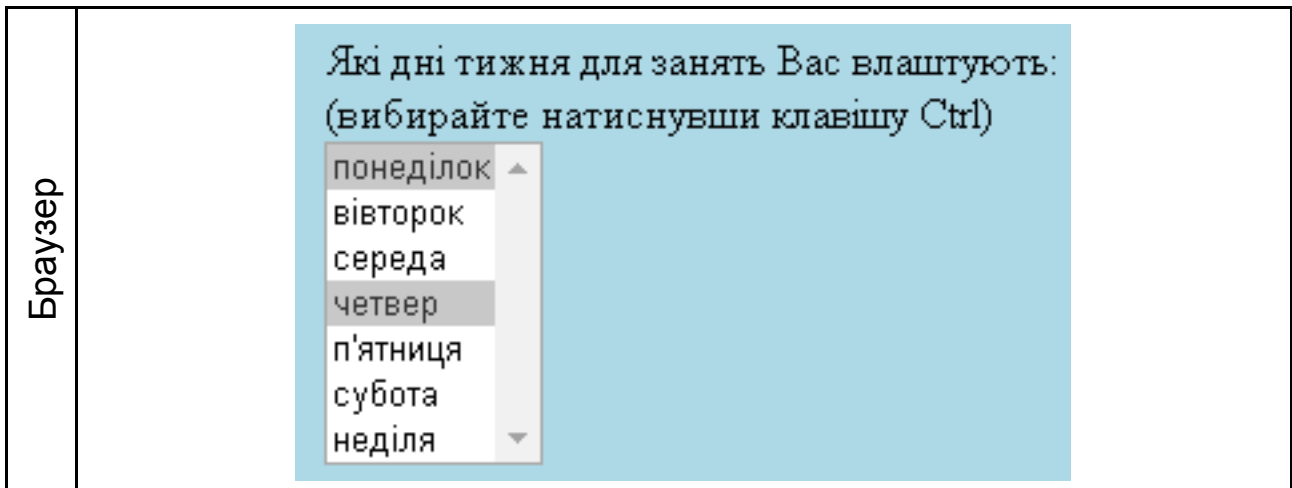


Рис. 5.14.14. Створення списку, що розкривається (вигляд у браузері)

Індивідуальні завдання

Створіть макет web-сторінки відповідно до наведених варіантів.

Порядок виконання:

1. Створіть на сторінці таблицю 6x6.
2. Об'єднайте клітинки таблиці та заповніть їхній вміст згідно з вашим варіантом.
3. Словами "синій", "жовтий", "зелений" і т. д. вказано колір фону клітинки.
4. Фраза "фоновий рисунок" визначає необхідність створення фонового зображення у клітинці.
5. Фрази "гіперпосилання 1", "гіперпосилання 2" та "гіперпосилання 3" визначають необхідність створення гіперпосилань на web-ресурс або HTML-файл.
6. Фрази "Рисунок 1", "Рисунок 2" та "Рисунок 3" передбачають вставку рисунка у клітинку.

Варіант 1

Знання мов <input checked="" type="checkbox"/> українська <input type="checkbox"/> англійська <input type="checkbox"/> німецька <input type="checkbox"/> іспанська			<ul style="list-style-type: none"> • <u>Реклама</u> ○ Медіа ▪ Бізнес 		
Гіперпосилання 1	Гіперпосилання 2	Гіперпосилання 3	Рожевий		
	Синій		Жовтий		
III. Реклама IV. <u>Медіа</u> V. Бізнес		Фоновий рисунок	Рисунок 1	Рисунок 2	Рисунок 3
			Рисунок 1	Я студент ^{ХНЕУ}	Рисунок 1
	© 2020				

Варіант 2

Вкажіть Вашу стать: <input checked="" type="radio"/> чоловіча <input type="radio"/> жіноча		Гіперпосилання 1		Гіперпосилання 2	
		Фоновий рисунок			
A. <u>Реклама</u> B. Медіа C. Бізнес		Рисунок 1	Я студент ^{ХНЕУ}	Рисунок 1	Зелений
	Жовтий				
		Рисунок 1	Рисунок 2	Рисунок 3	<ul style="list-style-type: none"> • Реклама ○ <u>Медіа</u> ▪ Бізнес
© 2020	Червоний		Гіперпосилання 3		

Варіант 3

	Гіперпосилання 1		<ul style="list-style-type: none"> • <u>Реклама</u> ○ Медіа ▪ Бізнес 	Гіперпосилання 3	
Гіперпосилання 2	Червоний			Жовтий	
Рожевий	Я студент ^{ХНЕУ}		Введіть пароль: <input type="text"/> <input type="button" value="Ввести пароль"/>		
VI. Реклама VII. <u>Медіа</u> VIII. Бізнес	Синій		Фоновий рисунок		
		Жовтий			
	© 2020				

Варіант 4

Рисунок 1			ЖОВТИЙ			Рисунок 1			Я студент ^{ХНЕУ}			Гіперпосилання 1		
Рисунок 1			Рисунок 2			Рисунок 3			Фоновий рисунок					
D. Реклама E. Медіа F. Бізнес			Гіперпосилання 2			Багаторядкове текстове поле:						Червоний		
									Гіперпосилання 3					
									<ul style="list-style-type: none"> ● Реклама ○ <i>Медіа</i> ■ Бізнес 					
Зелений						© 2020								

Варіант 5

Пошук по сайту			Знайти			<ul style="list-style-type: none"> ● <u>Реклама</u> ○ Медіа ■ Бізнес 			Синій					
Гіперпосилання 1		Гіперпосилання 2		Гіперпосилання 3		Рожевий								
Червоний						Фоновий рисунок			Жовтий					
Рисунок 1		Рисунок 2		Рисунок 3					© 2020			21. Реклама 22. Медіа 23. Бізнес		
Рисунок 1		Я студент ^{ХНЕУ}		Рисунок 1										

Варіант 6

Текстове поле: Введіть текст			Гіперпосилання 1			Гіперпосилання 2					
Рисунок 1		Рисунок 2		Рисунок 3		Фоновий рисунок					
Жовтий		15. Реклама 16. Медіа 17. Бізнес				Червоний			Рожевий		
Гіперпосилання 3			Я студент ^{ХНЕУ}			© 2020			<ul style="list-style-type: none"> ● Реклама ○ <i>Медіа</i> ■ Бізнес 		

Варіант 7

Гіперпосилання 1			<ul style="list-style-type: none"> ● <u>Реклама</u> ○ Аудит ■ Бізнес 	© 2020	
Гіперпосилання 2		Рисунок 1		Жовтий	
Рожевий	Гіперпосилання 3		Рисунок 2	Рисунок 3	
IX. Реклама X. <u>Медіа</u> XI. Бізнес	Синій		Рисунок 1	Я студент ^{ХНЕУ}	Рисунок 1
		Фоновий рисунок	<div style="border: 1px solid black; padding: 5px;"> Знання мов <input checked="" type="checkbox"/> українська <input type="checkbox"/> англійська <input type="checkbox"/> німецька <input type="checkbox"/> іспанська </div>		

Варіант 8

Гіперпосилання 1		Я студент ^{ХНЕУ}		Гіперпосилання 2
Рисунок 1		Рисунок 3	Фоновий рисунок	
G. <u>Маркетинг</u> H. <u>Менеджмент</u> I. <u>Аудит</u>		<div style="border: 1px solid black; padding: 5px;"> Введіть пароль: <input type="text"/> <input type="button" value="Ввести пароль"/> </div>		Червоний
	Рисунок 2	Рисунок 1	Рисунок 3	© 2020
Червоний	Гіперпосилання 3			<ul style="list-style-type: none"> ● Реклама ○ <u>Медіа</u> ■ Бізнес

Варіант 9

Рожевий		Рисунок 1	<ul style="list-style-type: none"> ● <u>Реклама</u> ○ Медіа ■ Бізнес 	Жовтий	
Гіперпосилання 1	Червоний			Фоновий рисунок	
Гіперпосилання 2	Я студент ^{ХНЕУ}		Гіперпосилання 3	Синій	
XII. Фінанси XIII. <u>Аудит</u> XIV. Маркетинг		Рисунок 2	<div style="border: 1px solid black; padding: 5px;"> понеділок вівторок середа четвер п'ятниця субота неділя </div>		
	© 2020	Зелений			

Варіант 10

Рисунок 1			Гіперпосилання 1	Гіперпосилання 2
Рисунок 1	Рисунок 2	Рисунок 3	Фоновий рисунок	
Рисунок 1	Я студент ХНЕУ	Рисунок 3		
18. <u>Реклама</u> 19. <u>Медіа</u> 20. <u>Бізнес</u>		Текстове поле: <input style="width: 150px;" type="text" value="Введіть текст"/>		
Жовтий			<ul style="list-style-type: none"> ● Реклама ○ <u>Медіа</u> ■ Бізнес 	
Гіперпосилання 3	Червоний		© 2020	

Запитання для самоконтролю

1. Які атрибути тегу <TABLE> ви знаєте?
2. У чому відмінність між тегами <TD> і <TH>?
3. Яким чином можна вирівняти текст у клітинці по горизонталі та вертикалі?
4. Яким чином можна змінити фон таблиці, рядка, клітинки?
5. Яким чином можна об'єднати клітинки в таблиці? Наведіть приклад.
6. Назвіть переваги та недоліки табличного верстання.
7. Для чого потрібна форма на web-сторінці? Назвіть основні елементи форми.
8. Як на формі створити текстове поле?
9. Як на формі створити прапорці?
10. Як на формі створити перемикачі?
11. Як на формі створити кнопки?
12. Які типи кнопок на формі ви знаєте? У чому їхня відмінність?
13. Із якою метою використовується тег <FIELDSET>?
14. Яким чином можна зв'язати певний текст і елементи форми?

Тестові завдання за розділом 5

1. Мова HTML призначена для:

- а) програмування розв'язання математичних задач;
- б) розроблення браузерів;
- в) створення гіпертекстових документів;
- г) перегляду гіпертекстових документів.

2. Для яких цілей може використовуватися тег <a>:

- а) для організації переходу за гіперпосиланням;
- б) для вставки рисунків;
- в) для створення абзацу;
- г) для вставки міток переходу?

3. Щоб закінчити рядок і почати новий рядок, використовують тег:

- а) <HR>;
- б) <A>;
- в)
;
- г) <P>;
- д) <END>.

4. Який із тегів розмістить текст посередині вікна браузера:

- а) <CENTER>;
- б) <MIDDLE>;
- в) <P>;
- г) <TITLE>;
- д) ?

5. Тег <HEAD> використовують для:

- а) введення заголовка до тексту;
- б) організації гіперпосилання;
- в) опису заголовка таблиці;
- г) створення головної частини HTML-документа, у якій браузеру повідомляється налаштування web-сторінки;
- д) виведення заголовка рисунка.

6. Який тег потрібно використовувати, щоб намалювати на екрані горизонтальну лінію:

- а) <H1>;
- б) <H6>;
- в)
;
- г) <A>;
- д) <HR>?

7. За допомогою тегу можна:

- а) оформити виділений фрагмент тексту курсивом;
- б) задати ім'я мітки для гіперпосилання;
- в) вказати адресу файлу, де знаходиться рисунок для вставки;
- г) ввести ім'я HTML-документа.

8. Який із варіантів вставки рисунка є правильним:

- а) ;
- б) <IMC SRG = "file.gif" ALT = "Гарний рисунок">;
- в) ;
- г) <IMC SRC = "file.gif" ALT = "Гарний рисунок">;
- д) ?

9. Для чого можна використовувати тег <SUP>:

- а) для збільшення розміру шрифту;
- б) для позначення найголовнішого заголовка в документі;
- в) для введення підрядкового знаку в математичній формулі вигляду $C_1 = A_1 + B_1$;
- г) для введення надрядкового знаку в математичній формулі вигляду $C^2 = A^2 + B^2$?

10. Тег <TITLE> використовується для:

- а) введення найголовнішого заголовка документа;
- б) введення заголовка таблиці;
- в) створення заголовка документа;
- г) вказівки браузеру ступеня важливості документа.

11. За допомогою тегу `<TR>` можна створити:

- а) підпис під рисунком;
- б) стовпчик таблиці;
- в) горизонтальну лінію на сторінці;
- г) рядок таблиці.

12. За допомогою тегу `<TD>` створюється:

- а) рядок таблиці;
- б) клітинка таблиці;
- в) заголовок таблиці;
- г) таблиця?

13. Який із варіантів об'єднання декількох клітинок таблиці є правильним:

- а) `<td rowspan="3">`;
- б) `<td colspan="4" rowspan="3">`;
- в) `<td rowspan="3" colspan="4" >`;
- г) `<td colspan="4">`;
- д) `<td rovsan="3">?`

14. Для створення форми на web-сторінці використовується тег:

- а) `<PRIVATE SUB>`;
- б) `<CASE>`;
- в) `<FORM>`;
- г) `<END SUB>`.

15. Які елементи управління не можуть розміщуватися на формі web-сторінки:

- а) Label;
- б) Text;
- в) Checkbox;
- г) CommandButton;
- д) Radio?

Розділ 6. Створення сайтів (блогів) з використанням HTML5 та CSS3

Метою розділу є формування у майбутніх фахівців системи компетентностей із використання мови розмітки гіпертексту HTML у процесі створення сайтів.

Професійні компетентності

У результаті вивчення навчальної дисципліни студент має:

знати:

принципи зміни зовнішнього вигляду сайтів;

сучасні технології розроблення сайтів;

вміти:

оформляти сайти у вибраному стилі з використанням CSS;

формулювати вимоги щодо оформлення дизайну сайту;

проектувати дизайн сайтів;

створювати адаптивний дизайн сайту.

Ключові терміни: стилі, селектори, CSS, форматування, класи, псевдокласи, блок, властивості блоків, моделі позиціонування, блокова верстка, теги HTML5, адаптивна верстка, flexbox, медіазапити, Bootstrap.

Теоретичні відомості

Мова HTML визначає, що буде розміщено на web-сторінці. В HTML є досить багато засобів для форматування вигляду, розміру, кольору та інших елементів на web-сторінці. Проте з часом цього стало недостатньо. У зв'язку з цим, у середині 90-х років XX століття активно почався розвиток CSS (Cascading Style Sheets – каскадні таблиці стилів) – формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки.

CSS використовується розробниками web-сторінок для задавання кольорів, шрифтів, стилів, розташування окремих блоків та інших аспектів представлення зовнішнього вигляду цих web-сторінок. Основною метою розроблення CSS було відділення опису логічної структури web-сторінки (яке проводиться за допомогою HTML) від опису зовнішнього вигляду цієї web-сторінки (яке тепер проводиться за допомогою формальної мови CSS). Такий поділ може збільшити доступність документа, надати велику гнучкість і можливість управління його поданням, а також зменшити складність і повторюваність у структурному вмісті.

15. Форматування елементів web-сторінки засобами CSS

Мета: ознайомитися з технологією використання стилів для зміни зовнішнього вигляду та положення елементів web-документа.

Ключові терміни: стилі, селектори, CSS, форматування, класи, псевдокласи.

Теоретичні відомості

Стили CSS – це набір правил для форматування як елементів на web-сторінці, так і самої web-сторінки. Стили CSS можуть бути задані такими методами.

1. Опис стилів знаходиться в окремому файлі, він може бути підключений до документа за допомогою тегу <LINK>, включеного в контейнер <HEAD></HEAD> (рис. 6.15.1).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Зовнішній файл CSS</title>
    <link href="style.css" rel="stylesheet" type="text/css">
  </head>
  <body>
    .....
  </body>
</html>
```

Рис. 6.15.1. Підключення зовнішнього файлу стилів CSS

Такий підхід дозволяє використовувати один файл зі стилями для будь-якої кількості web-сторінок, також його можна застосовувати на інших сайтах; можна змінювати стилі без модифікації web-сторінок. Зміна стилю в одному файлі автоматично застосовується до всіх сторінок.

2. Стили, описані всередині документа, можуть бути включені у контейнер <STYLE></STYLE> і зазвичай розташовуються в заголовку сторінки в контейнері <HEAD></HEAD> (рис. 6.15.2).

```

<!DOCTYPE html>
<html>
  <head>
    <title>Внутрішні стилі CSS</title>
    <style>
      .....
    </style>
  </head>
  <body>
    .....
  </body>
</html>

```

Рис. 6.15.2. Підключення внутрішніх стилів CSS

За своєю гнучкістю та можливостями цей спосіб використання стилів поступається попередньому, але виключає необхідність роботи з декількома документами.

3. Стили, описані в тілі документа за допомогою атрибута STYLE окремого тегу (рис. 6.15.3).

```

<!DOCTYPE>
<html>
  <head>
    <title>Стили, вбудовані в елемент HTML</title>
  </head>
  <body>
    <p style="...">
      .....
    </p>
  </body>
</html>

```

Рис. 6.15.3. Підключення стилів в елемент HTML

Слід відмітити, що у CSS синтаксис задання властивостей (атрибутів) відмінний від HTML. Після назви атрибута ставиться ":" і значення атрибута в лапки не береться. Після значення атрибута обов'язково ставиться ";". Наприклад: "color:red;".

Синтаксис запису коментарів у CSS-файлах також відрізняється від HTML і має такий вигляд: /*Текст коментаря*/.

Розглянемо особливості форматування шрифтів, тексту та фону web-сторінок за допомогою CSS.

Властивості (атрибути) для зміни вигляду шрифту і його розміру за допомогою CSS наведено в табл. 6.15.1.

Таблиця 6.15.1

Зміна вигляду шрифту і його розміру за допомогою CSS

Властивість	Значення	Опис	Приклад
font-family	ім'я шрифту	задає список шрифтів	font-family: Arial;
font-style	normal italic oblique	Нормальний шрифт Курсив Похилий шрифт	font-style: italic;
font-variant	normal small-caps	Нормальний шрифт Малі прописні букви)	font-variant: small-caps;
font-weight	normal lighter bold bolder 100 – 900	Нормальна жирність Світле креслення Напівжирний Жирний 100 – світлий шрифт, 900 – самий жирний	font-weight: bold;
font-size	normal pt px %	нормальний розмір Пункти Пікселі Проценти	font-size: normal; font-size: 12pt; font-size: 12px; font-size: 120%;

Властивості для роботи з текстом за допомогою CSS наведено в табл. 6.15.2.

Таблиця 6.15.2

Властивості CSS для управління видом тексту

Властивість	Значення	Опис	Приклад
1	2	3	4
line-height	normal множник точно %	Інтерліньяж (міжрядковий інтервал)	line-height: normal; line-height: 1.5; line-height: 12px; line-height: 120%;
text-decoration	none underline overline line-through blink	Прибрати всі оформлення Підкреслення Лінія над текстом Перекреслення Миготіння тексту	text-decoration: none;

1	2	3	4
text-transform	none capitalize uppercase lowercase	Прибрати всі ефекти Починати з Прописної Усі ПРОПИСНІ усі малі	text-transform: capitalize;
text-align	left right center justify	По лівому краю По правому краю По центру По ширині	text-align: justify;
text-indent	Значення %	Відступ першого рядка	text-indent: 15px; text-indent: 10%;
color	Колір	Колір тексту	color:green;

Властивості для роботи з фоном за допомогою CSS наведено в табл. 6.15.3.

Таблиця 6.15.3

Властивості CSS для управління фоном web-сторінки

Властивість	Значення	Опис	Приклад
1	2	3	4
background-color	Колір	Визначає колір фону	background-color: pink;
background-image	url none	Визначає рисунок фонового зображення елемента: адреса рисунка; рисунок відсутній	background-image: url(весна.jpg);
background-repeat	repeat-x repeat-y no-repeat	Керує методом повторення фонового зображення: повторення зображення по горизонталі; повторення зображення по вертикалі; зображення не повторюється	background-repeat: no-repeat;
background-attachment	scroll	Керує методом прикріплення зображення до вікна браузера: браузер переміщує фонове зображення разом із текстом;	background-attachment: fixed;

1	2	3	4
	fixed	зображення не рухається стосовно вікна браузера під час прокрутки документа	
background-position	left, center, right; top, center, bottom; 0% 0% 0% 100% 0% 100% 50% 50% 50% 100% 100% 100%	Задає початкове положення фонового зображення: по горизонталі; по вертикалі; лівий верхній кут; правий верхній кут; лівий нижній кут; по центру; по центру внизу; правий нижній кут	background-position: 50% 50%;
background-size	cover contain auto auto % px, cm	Встановлює масштаб фонового зображення: масштабує зображення так, щоб його висота і ширина дорівнювали висоті та ширині блока; масштабує зображення так, щоб рисунок помістився всередину блока; розміри зображення залишаються вихідними; задає розмір фонові картинки в процентах від ширини або висоти блока; розмір у будь-яких доступних для CSS одиницях	background-size: cover;
background		Запис усіх властивостей в один рядок	background: url(весна.jpg); no-repeat; cover;

Підключаючи стилі окремим файлом *style.css* або за допомогою контейнера `<STYLE></STYLE>` всередині web-сторінки, кожне правило CSS має дві основні частини – селектор і блок оголошень.

Селектори визначають набір елементів web-сторінки, до яких застосовуються ті чи інші стилі. Селектор, розташований у лівій частині правила до знака "{", визначає, на які частини документа поширюється правило.

Блок оголошень розташовується у правій частині правила. Він міститься у фігурних дужках, і, у свою чергу, складається з одного або більше оголошень, розділених знаком ";" (рис. 6.15.4).

```
селектор1, селектор2 {  
    властивість: значення;  
    властивість: значення;  
    властивість: значення;  
}
```

Рис. 6.15.4. Синтаксис задавання стилів за допомогою селекторів

Кожне оголошення є поєднанням властивості CSS і значення, розділених знаком ":". Селектори можуть групуватися в одному рядку через кому. У такому випадку властивість застосовується до кожного з них.

Можна виділити такі основні типи селекторів:

1. Універсальний селектор. Використовується для встановлення одного стилю для всіх елементів web-сторінки, що мають цю властивість (колір, шрифт та ін.). Для позначення універсального селектора застосовується символ зірочки (*). Приклад використання універсального селектора наведено на рис. 6.15.5.

HTML	<pre><!doctype html> <html> <head> <title> Селектори </title> <meta charset = "utf-8"> <link href="style.css" rel="stylesheet" type="text/css"> </head> <body> <p>Універсальний селектор</p> </body> </html></pre>
CSS	<pre>*{color:orange;}</pre>

Рис. 6.15.5. Приклад підключення зовнішнього файлу (style.css) та використання універсального селектора

Під час відображення у браузері текст і гіперпосилання будуть оранжевого кольору.

2. Селектор елемента (тегу). Селектори елементів дозволяють форматовувати всі елементи цього типу. Приклад форматування всіх заголовків h1 наведено на рис. 6.15.6.

CSS	HTML
<pre>h1{font-family: Arial; font-style: italic;}</pre>	<pre><h1>Селектор елемента</h1></pre>

Рис. 6.15.6. Приклад селектора елемента

3. Селектор класу. Селектори класу дозволяють задавати стилі для одного і більше елементів з однаковим ім'ям класу, розміщених у різних місцях сторінки або на різних сторінках сайту (рис. 6.15.7).

CSS	HTML
<pre>.A{color: green; font-size: 40px;}</pre>	<pre><p class="A">Селектор класу</p></pre>

Рис. 6.15.7. Приклад селектора класу

4. Селектор ідентифікатора. Селектор ідентифікатора дозволяє форматовувати один конкретний елемент. Значення id має бути унікальним, на одній сторінці може зустрічатися тільки один раз і має містити щонайменше один символ. Значення не має містити пробілів.

Унікальний ідентифікатор елемента може використовуватися для різних цілей, зокрема, як спосіб посилання на конкретні частини документа з використанням ідентифікаторів фрагментів, як спосіб націлювання на елемент під час створення сценаріїв і як спосіб стилізації конкретного елемента з CSS (рис. 6.15.8).

CSS	HTML
<pre>#B{text-decoration: underline; color: red;}</pre>	<pre><p id="B">Унікальний ідентифікатор</p></pre>

Рис. 6.15.8. Приклад унікального ідентифікатора

5. Селектор нащадків. Селектори нащадків застосовують стилі до елементів, розташованих усередині елемента-контейнера. Наприклад, `div p` вибере всі елементи `p`, які є нащадками всіх елементів `div` (рис. 6.15.9).

CSS	HTML
<pre>div p{color:blue; text-align:center;}</pre>	<pre><div> <p>Селектор нащадків</p> </div></pre>

Рис. 6.15.9. Приклад селектора нащадків

6. Селектор псевдокласу. Псевдокласи – це класи, практично не прикріплені до HTML-тегів. Вони дозволяють застосувати CSS-правила до елементів під час здійснення події або підпорядковуються певним правилам. Опис псевдокласів наведено в табл. 6.15.4.

Таблиця 6.15.4

Опис псевдокласів

Псевдоклас	Опис
:link	не відвідане гіперпосилання
:visited	відвідане гіперпосилання
:active	елемент, який був активізований користувачем
:hover	будь-який елемент, на який наводять курсор миші
:focus	інтерактивний елемент, до якого перейшли за допомогою клавіатури або активували за допомогою миші
:valid	поля форми, вміст яких пройшов перевірку в браузері на відповідність зазначеного типу даних
:invalid	поля форми, вміст яких не відповідає вказаним типам даних
:enabled	всі активні поля форм
:disabled	заблоковані поля форм, тобто такі, що знаходяться в неактивному стані
:in-range	поля форми, значення яких знаходяться у заданому діапазоні
:out-of-range	поля форми, значення яких не входять у встановлений діапазон
:lang ()	елементи з текстом на зазначеній мові
:not (селектор)	елементи, які не містять вказаний селектор – клас, ідентифікатор, назву або тип поля форми
:target	елемент із символом #, на який посилаються в документі
:checked	виділені (вибрані користувачем) елементи форми

Приклад використання псевдокласів для форматування гіперпосилань наведено на рис. 6.15.10.

CSS	HTML
<pre>a:link{color:blue;} a:visited{color:green;} a:hover{color:red;} a:active{color:black;}</pre>	<pre>Google
 ХНЕУ ім. С.Кузнеця</pre>

Рис. 6.15.10. Приклад застосування псевдокласів

Стили, вбудовані в елемент HTML, підключаються за допомогою атрибута STYLE і діють тільки на той елемент, для якого вони задані (рис. 6.15.11).

```
<p style="color:pink; text-align:right; font-size:70px;
font-weight:bold;">Стили, вбудовані в елемент HTML</p>
```

Рис. 6.15.11. Стили, вбудовані в елемент HTML

У разі необхідності для виділення частини інформації всередині інших тегів і встановлення для неї свого стилю використовується тег , призначений для визначення малих елементів документа. Наприклад, всередині абзаца (тегу <P>) можна змінити колір і розмір слова, якщо додати початковий і кінцевий тег і визначити для нього стиль тексту (рис. 6.15.12).

```
<p style="color:pink; text-align:right; font-size:70px; font-weight:bold;">
Стили, <span style="color:yellow; font-size:100px;">вбудовані</span>
в елемент HTML</p>
```

Рис. 6.15.12. Приклад використання тегу

Усі описані методи використання CSS можуть застосовуватися як самостійно, так і разом. Застосування CSS до документів HTML засноване на принципах успадкування і каскадності.

Принцип успадкування полягає в тому, що властивості CSS, оголошені для елементів-предків, майже завжди успадковуються елементами-нащадками.

Принцип каскадності застосовується тоді, коли якомусь елементу HTML одночасно поставлено у відповідність більше одного правила CSS, тобто тоді, коли відбувається конфлікт значень цих правил. Щоб вирішити такі конфлікти, вводяться правила пріоритету:

- найнижчий пріоритет має стиль браузера;
- наступним за значущістю є стиль, заданий користувачем браузера в його налаштуваннях;
- найвищий пріоритет має стиль, заданий безпосередньо автором сторінки.

В авторському стилі пріоритети розставляються таким чином: найбільший пріоритет мають стилі, вбудовані в елемент HTML, далі – внутрішні стилі, найменший пріоритет мають зовнішні стилі.

Практичне виконання роботи

Завдання 15.1. Використання зовнішніх файлів стилів CSS для форматування web-сторінки

Створіть web-сторінку та відформатуйте її за допомогою зовнішнього файлу стилів CSS.

Вирішення

1. У редакторі Notepad++ створіть файли *Селектору.html* та *style.css*.
2. Для створення файлу *style.css* у вікні редактора Notepad++ відкрийте новий файл (вкладка "**Файл**"→**Новий**), задайте мову сторінки (вкладка "**Мова**"→**CSS**) і тип кодування (вкладка "**Кодування**"→**UTF-8**) та збережіть файл із відповідним ім'ям та розширенням.
3. Підключіть файл *style.css* до HTML-файлу *Селектору.html*, як показано на рис. 6.15.1.
4. Використовуючи раніше наведені приклади (рис. 6.15.5 – 6.15.10), послідовно (шляхом накопичення) відпрацюйте особливості використання зовнішніх стилів CSS для форматування web-сторінок.

Завдання 15.2. Використання внутрішніх стилів CSS для форматування web-сторінки

Створіть web-сторінку та відформатуйте її за допомогою внутрішніх стилів CSS.

Вирішення

1. У редакторі Notepad++ створіть файл *Селектору1.html*, зробивши копію з файлу *Селектору.html*.
2. Видаліть у файлі *Селектору1.html* тег <LINK> та перевірте відображення файлу *Селектору1.html* у браузері. Проаналізуйте результати.
3. Вставте замість тегу <LINK> контейнер <STYLE></STYLE>, як показано на рис. 6.15.2.
4. Виділіть вміст файлу *style.css*, зробіть копію та вставте в контейнер <STYLE></STYLE>.
5. Перевірте відображення файлу *Селектору1.html* у браузері. Проаналізуйте результати.

Завдання 15.3. Використання стилів CSS, вбудованих в елемент HTML для форматування web-сторінки

Створіть абзац на web-сторінці та відформатуйте його за допомогою вбудованих в елемент HTML стилів CSS.

Вирішення

1. У файлі *Селектору1.html* додайте новий абзац з текстом та відформатуйте його за допомогою вбудованих в елемент HTML стилів CSS, як показано на рис. 6.15.11.
Перевірте відображення файлу *Селектору1.html* у браузері. Проаналізуйте результати.
2. Використайте тег для форматування тексту всередині абзацу, як показано на рис. 6.15.12.
Перевірте відображення файлу *Селектору1.html* у браузері. Проаналізуйте результати.

Індивідуальні завдання

Створіть сайт, що складається з 5 – 6 сторінок.

Порядок виконання та основні вимоги:

1. Тема сайту має відповідати вашій майбутній професійній діяльності.
2. Зверстайте макети сторінок сайту, використовуючи прийоми табличного верстання, та наповніть відповідним вмістом.
3. Форматування web-сторінок здійснюйте тільки за допомогою стилів CSS.
4. На одній із web-сторінок необхідно розмістити форму для оформлення замовлення або відгуку про сайт.
5. Використовуючи лекційний матеріал, розмістіть сайт у мережі "Інтернет" на одному із безкоштовних хостингів.

Запитання для самоконтролю

1. Що таке CSS?
2. Які методи задавання стилів CSS ви знаєте? Назвіть переваги та недоліки.
3. Назвіть основні властивості форматування шрифтів за допомогою CSS.
4. Назвіть основні властивості форматування тексту за допомогою CSS.
5. Назвіть основні властивості форматування тла за допомогою CSS.
6. Що таке "селектор"? Назвіть основні типи селекторів.
7. Для чого використовується тег <LINK>? Наведіть приклад запису на web-сторінці.
8. Для чого використовується тег <STYLE>? Наведіть приклад запису на web-сторінці.
9. Для чого використовується атрибут STYLE? Наведіть приклад запису на web-сторінці.
10. Для чого використовується тег ? Наведіть приклад запису на web-сторінці.
11. У чому полягає принцип успадкування стилів CSS?
12. У чому полягає принцип каскадності стилів CSS?

16. Блокова верстка сайту

Мета: вивчити особливості роботи з блоковими елементами в HTML. Створити сторінку за допомогою блокової верстки.

Ключові терміни: блок, властивості блоків, моделі позиціонування, блокова верстка, теги HTML5.

Теоретичні відомості

Основними недоліками табличного верстання є: громіздкий код, відсутність гнучкості, погана індексація пошуковими системами, тривалий час завантаження сайтів та ін. Таблиці поступово здають свої позиції на користь верстки блоками (шарами).

Блоковим називається елемент, який відображається на сторінці у вигляді прямокутника. Він займає всю доступну ширину браузера, висота елемента визначається його вмістом і він завжди починається з нового рядка. До блокових елементів відносяться теги `<P>`, `<H1>` - `<H6>`, `<DIV>`, `` та ін.

Дозволяється вкладати один блок в інший, а також розміщувати всередині блоків рядкові елементи.

Наприклад: `<h2> Головна</h2>`.

Забороняється додавати всередину рядкових елементів блокові.

Наприклад: `<h2> Головна</h2>`.

Кожен блоковий елемент документа займає прямокутну область, яка розміщена послідовно ще в трьох областях: `padding`, `border`, `margin`.

Основою блоку виступає його контент (це може бути текст, зображення тощо), ширина якого задається властивістю `width`, а висота через `height`; навколо контенту йдуть внутрішні поля (`padding`), вони створюють порожній простір від контенту до внутрішнього краю рамки, потім йдуть власне сама рамка (`border`) і завершують блок зовнішні відступи (`margin`), невидимий порожній простір від зовнішнього краю рамки (рис. 6.16.1).

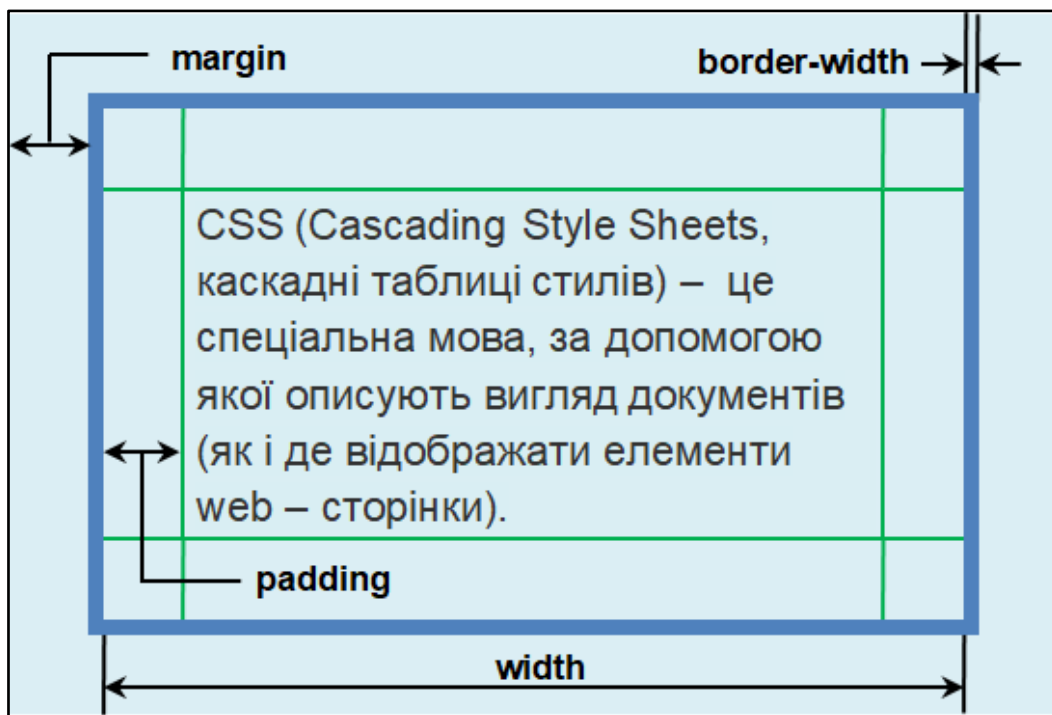


Рис. 6.16.1. Складові блокового елемента

За замовчуванням ширина блоку обчислюється автоматично і займає весь доступний простір. Тут слід обумовити, що під цим мається на увазі. Наприклад, якщо в коді документа присутній один тег `<DIV>`, то він займає всю вільну ширину вікна браузера і ширина блоку буде дорівнювати 100 %. Варто помістити один тег `<DIV>` всередину іншого, як ширина внутрішнього тегу починає обчислюватися щодо батьківського, тобто зовнішнього контейнера.

Аналогічно, браузер за висоту шару набуває значення властивості `height` і додає до нього ще значення `margin`, `padding` і `border`. Якщо висота блоку не встановлена явно, то вона обчислюється автоматично, виходячи з обсягу вмісту.

Фон блоку найпростіше встановлювати через універсальну властивість `background`. Кольором фону заливається область, яка визначається значеннями `width`, `height` і `padding`. Таким чином, `margin` не бере участі у формуванні кольорової області.

Відступи можна задавати окремо, використовуючи властивості: `margin-left`, `margin-right`, `margin-top`, `margin-bottom` або значення `auto` (стандартний відступ). Аналогічно задаються відступи всередині блока: `padding-left`, `padding-right`, `padding-top`, `padding-bottom` або значення `auto` (стандартний відступ).

Слід відмітити, що коли зовнішні вертикальні відступи блокових елементів стикаються, то залишається відступ, значення якого дорівнює значенню більшого з них, водночас значення відступу елемента з меншою межею дорівнює нулю. Таке явище називається "схлопуванням" зовнішніх відступів і характерне тільки для властивостей `margin-top` і `margin-bottom`.

Властивості для форматування рамки блоку наведено в табл. 6.16.1.

Таблиця 6.16.1

Зміна вигляду шрифту і його розміру за допомогою CSS

Властивість	Значення	Опис	Приклад
<code>border-width</code>	<code>thin</code> <code>medium</code> <code>thick</code>	Товщина межі (задається словами або у px): тонка середня товста	<code>border-width: 2px;</code>
<code>border-style</code>	<code>solid</code> <code>dashed</code> <code>dotted</code> <code>double</code> <code>dotted</code> <code>groove</code> <code>inset</code>	Стиль межі: суцільна; пунктиром; крапками; подвійна; за допомогою крапок; паз; втоплена	<code>border-style: dashed;</code>
<code>border-color</code>	колір	Колір рамки	<code>border-color: green;</code>
<code>border</code>		Всі властивості записуються в один рядок	<code>border: medium solid red;</code>

Якщо вказано єдине значення властивостей, то всі чотири сторони рамки будуть однаковими. Є можливість задавати кожній стороні рамки різні значення властивостей.

На рис. 6.16.2 наведено приклад форматування рамки блокового елемента.

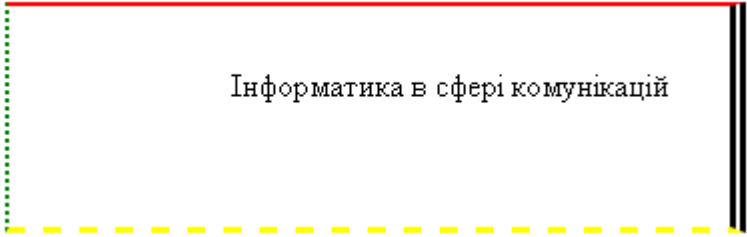
Код HTML (CSS)	<pre> <!DOCTYPE HTML> <html> <head> <title>Блок</title> <meta charset="utf-8"> <style> .p1{ border-top:2px solid red; border-bottom:3px dashed yellow; border-left:2px dotted green; border-right:8px double black; text-align:right; padding:30px; margin:25px; width:300px; height:50px;} </style> </head> <body> <p class="p1">Інформатика в сфері комунікацій</p> </body> </html> </pre>
Браузер	

Рис. 6.16.2. Форматування рамки блокового елемента

Вирівнювання (обтікання) блокових елементів

За замовчуванням елементи на web-сторінці відображаються в тому порядку, в якому вони подані у HTML-документі:

блокові елементи займають всю доступну для них ширину і розміщуються вертикально один під одним;

рядкові елементи розміщуються по горизонталі доти, поки не буде зайнята вся доступна ширина, після чого буде зроблений перенос рядка і все піде з початку.

Такий порядок розташування елементів називається нормальним потоком.

Властивість **float** визначає вирівнювання елемента на сторінці та сторону, з якої його будуть обтікати інші елементи і має такі значення:

left – вирівнює елемент по лівому краю, решта обтікають справа;

right – вирівнює елемент по правому краю, решта обтікають зліва;

none – обтікання не задано (за замовчуванням).

Властивість **clear** встановлює, з якого боку заборонено обтікання елемента іншими елементами, і має такі значення:

left – елемент зміщується вниз щодо змінного елемента зі встановленим лівим обтіканням, але праве обтікання залишається у силі;

right – елемент зміщується вниз щодо змінного зі встановленим правим обтіканням, параметр лівого обтікання залишається у силі;

both – скасовує обтікання з обох боків;

none – обтікання відбувається згідно з заданими параметрами, відповідно до властивості **float** (за замовчуванням).

На рис. 6.16.3 наведено приклад використання властивості **float**.

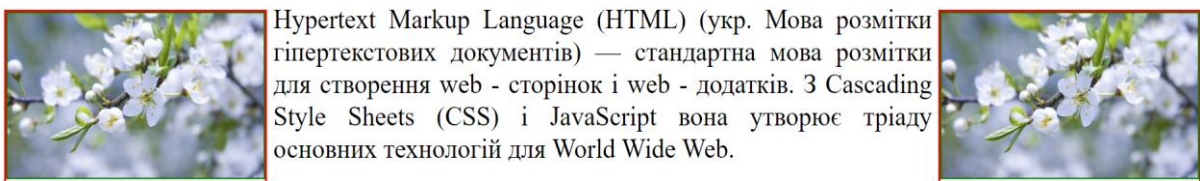
Код HTML (CSS)	<pre> <div style="border: solid red 3px; float:left;"> </div> <div style="border: solid red 3px; float:right;"> </div> <p style="text-align:justify; font-size:40px;">Hypertext Markup Language (HTML) (укр. Мова розмітки гіпертекстових документів) – стандартна мова розмітки для створення web - сторінок і web - додатків. З Cascading Style Sheets (CSS) і JavaScript вона утворює триаду основних технологій для World Wide Web.</p> </pre>
Браузер	

Рис. 6.16.3. Приклад використання властивості **float**

Властивість **overflow**

Властивість **overflow** управляє тим, як поводить себе вміст блочного елемента, якщо його розмір перевищує допустиму довжину/ширину.

Зазвичай блок збільшується у розмірах під час додавання до нього елементів, що містять в собі всіх нащадків. Але що робити, якщо висота/ширина вказані явно? Тоді блок не може збільшитися, і вміст "переповнює" блок. Його відображення в цьому випадку задається властивістю **overflow**.

Можливі значення:

visible – відображається весь вміст елемента, навіть за межами встановленої висоти і ширини (за замовчуванням);

hidden – відображається тільки область усередині елемента, інше буде приховано;

scroll – завжди додаються смуги прокрутки;

auto – смуги прокрутки додаються тільки в разі перевищення розмірів.

На рис. 6.16.4 наведено приклад використання властивості overflow.

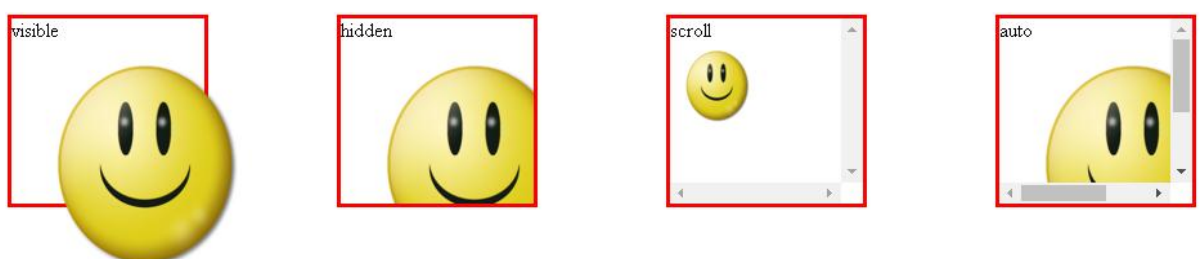
Код HTML (CSS)	<pre><!-- відображається весь вміст елемента, навіть за межами встановленої висоти і ширини --> <div style="border:solid red 3px; width:150px; height:150px; float:left; margin:50px; overflow:visible;">visible </div> <!-- відображається тільки область всередині елемента, інше буде приховано --> <div style="border:solid red 3px; width:150px; height:150px; float:left; margin:50px; overflow:hidden;">hidden </div> <!-- завжди додаються смуги прокрутки --> <div style="border:solid red 3px; width:150px; height:150px; float:left; margin:50px; overflow:scroll;">scroll
 </div> <!-- смуги прокрутки додаються тільки при перевищенні розмірів --> <div style="border:solid red 3px; width:150px; height:150px; float:left; margin:50px; overflow:auto;">auto </div></pre>
Браузер	 The image shows four browser window screenshots illustrating different overflow behaviors. Each window has a red border and a title bar. 1. 'visible': A yellow smiley face is shown, partially cut off by the right edge of the window. 2. 'hidden': A yellow smiley face is shown, completely cut off by the right edge of the window. 3. 'scroll': A yellow smiley face is shown, completely cut off by the right edge of the window, but a vertical scrollbar is visible on the right side. 4. 'auto': A yellow smiley face is shown, completely cut off by the right edge of the window, and a vertical scrollbar is visible on the right side.

Рис. 6.16.4. Приклад використання властивості overflow

Властивість display

За допомогою властивості **display** в CSS можна змусити блоковий елемент виглядати як вбудований (рядковий) та навпаки.

Вбудованими називаються такі елементи web-сторінки, які є безпосередньою частиною іншого елемента, наприклад, текстового абзаца. До вбудованих елементів належать теги ``, `<A>`, `<Q>`, `<CODE>` та ін. В основному вони використовуються для зміни вигляду тексту або його логічного виділення.

Відмінність між блоковими та вбудованими елементами така:

вбудовані елементи можуть містити тільки дані або інші вбудовані елементи, а в блокові допустимо вкладати інші блокові елементи, рядкові елементи, а також дані. Інакше кажучи, вбудовані елементи не можуть зберігати блокові елементи;

блокові елементи завжди починаються з нового рядка, а рядкові в такий спосіб не акцентуються;

блокові елементи займають усю доступну ширину, наприклад, вікна браузера, а ширина вбудованих елементів дорівнює їхньому вмісту плюс значення відступів, полів і меж.

Щоб блоковий елемент поведився як рядковий (під час заповнення всього рядка переноситься на новий рядок), для нього необхідно записати правило:

`display: inline;`

Якщо ж необхідно відобразити рядковий елемент як блок-елемент (щоб до і після елемента відбувався перенос рядка), то правило записується таким чином:

`display: block;`

Основні можливі значення властивості `display` наведено в табл. 6.16.2.

Таблиця 6.16.2

Зміна вигляду шрифту і його розміру за допомогою CSS

Властивість	Значення
1	2
block	Елемент відображається як блоковий. Застосування цього значення для вбудованих елементів, наприклад, тегу , змушує його вести себе подібно до блоків – відбувається перенесення рядків на початку і в кінці вмісту
border-inline	Елемент відображається як вбудований. Використання блокових тегів, таких як <DIV> і <P>, автоматично створює перенос і показує вміст цих тегів із нового рядка. Значення inline скасовує цю особливість, тому вміст блокових елементів починається з того місця, де закінчився попередній елемент
inline-block	Це значення генерує блоковий елемент з поведінкою, як у рядкового. У цьому випадку весь вміст таких блокових елементів буде відображатися, як зазвичай, але водночас блоки будуть вести себе як рядкові елементи, шикуючись в одному рядку один за одним, і переноситися на новий рядок лише за необхідності. "Схлопування" margin у таких випадках перестає діяти
inline-table	Визначає, що елемент є таблицею, як під час використання тегу <TABLE>, але водночас таблиця є вбудованим елементом і відбувається її обтікання іншими елементами, наприклад, текстом

1	2
list-item	Елемент відображається як блоковий і додається маркер списку
none	Тимчасово видаляє елемент із документа. Займане ним місце не резервується і web-сторінка формується так, немов елемента і не було. Змінити значення і зробити знову видимим елемент можна за допомогою скриптів, звертаючись до властивостей через об'єктну модель. У цьому випадку відбувається переформатування даних на сторінці з урахуванням знову доданого елемента

Моделі позиціонування

CSS позиціонування – це управління місцем і способом відображення елемента на web-сторінці, воно здійснюється за допомогою властивості **position**.

Властивість *position* має 4 значення:

static – статичне позиціонування;

relative – відносне позиціонування;

fixed – фіксоване позиціонування;

absolute – абсолютне позиціонування.

Значення властивостей (px, %) top, left, right або bottom задають місце елемента на сторінці.

Статичним (position:static;) називають позиціонування, задане за замовчуванням для всіх елементів на web-сторінці.

Якщо до елемента не застосовувати властивість *position*, то він буде статичним і буде відображатися на web-сторінці відповідно до загального потоку елементів.

Під час застосування CSS властивостей top, left, right або bottom до статично позиціонованих елементів вони будуть ігноруватися.

Елементи з **відносним (position:relative;)** позиціонуванням, як і статичні елементи, залишаються в загальному потоці.

Під час застосування властивостей top, left, right або bottom до відносно позиціонованих елементів вони будуть зміщуватися щодо свого місця розташування, залишаючи порожній простір там, де елемент розташовувався спочатку. Такі елементи не впливають на розташування елементів, що їх оточують, інші елементи залишаються на своїх місцях і можуть бути перекриті позиціонованим елементом.

Елементи з **фіксованим (position:fixed;)** позиціонуванням розташовуються на сторінці відносно вікна браузера.

Такі елементи видаляються із загального потоку. Елементи, які прямують у потоці за фіксованим елементом, будуть ігнорувати його, зміщуючись і займаючи його місце на web-сторінці.

Під час застосування фіксованого позиціонування положення елемента задається значеннями властивостей top, left, right або bottom.

Прокручуючи сторінки, елементи з фіксованим позиціонування створюють ефект нерухомих об'єктів і можуть перекривати інші елементи, приховуючи їх повністю або частково.

Елементи з **абсолютним (position:absolute;)** позиціонуванням повністю видаляються із загального потоку, інші елементи будуть займати простір, що звільнився, повністю ігноруючи абсолютно позиціонований елемент.

Можна позиціонувати елемент у будь-яке потрібне місце web-сторінки, використовуючи властивості top, left, right або bottom.

Коли елементи знаходяться поза загальним потоком сторінки, вони можуть перекривати один одного. Управляти порядком перекриття можна за допомогою CSS властивості **z-index**, чим більше його значення (задається цифрами), тим вище буде елемент по z-осі. У такий спосіб зручно виводити на web-сторінці різноманітні підказки, спливаючі вікна, рекламу та змінні меню.

Нові теги в HTML5

Мета розроблення HTML5 – поліпшення рівня підтримки мультимедіа-технологій, поліпшення семантичного ядра сайтів, розширення і раціоналізації розмітки документів.

Для коректного відображення нових тегів HTML5 їх доцільно спочатку зробити блоковими через властивість (display:block;), тоді вони почнуть коректно виводитися у всіх браузерах. Усі раніше вказані теги є парними.

Тег <HEADER> дозволяє визначити заголовки блок (шапку) сайту. У заголовному блоці зазвичай розташовуються логотип, гасло сайту, а також поле пошуку і посилання на інші розділи сайту.

Тег <FOOTER> дозволяє визначити футер (підвал) для документа або розділу. У футері зазвичай розташовується інформація про авторське

право, автора документа і можуть бути присутніми посилання на інші ресурси.

Тег <NAV> дозволяє створити навігаційний блок. У навігаційному блоці зазвичай містяться посилання на інші сторінки сайту.

Тег <MAIN> містить головну інформацію вашого сайту. Такі повторювані елементи як логотип, вікно пошуку, меню навігації не рекомендується вкладати в <main>. Також не варто поміщати сам елемент <main> усередину елементів <article>, <aside>, <header>, <footer> або <nav>.

Тег <ARTICLE> містить той контент, який може бути видалений без шкоди для всього сайту. Наприклад, короткий опис новин, рекламний банер, стаття, коментарі.

Тег <ASIDE> використовується для виділення елементів, які не є частиною вмісту, але побічно з ним пов'язані. Може містити такий елемент, як <NAV>.

Приклад вмісту, який може бути вміщено в цей тег: цитати, додаткова інформація, словник із термінами, список посилань. Такий блок, якщо він розташовується збоку, називається, здебільшого, "сайдбар" (sidebar) або "бічна панель".

Тег <SECTION> визначає розділ документа, який може містити заголовки, шапку, "підвал" (footer) і текст. Допускається вкладати один тег <SECTION> всередину іншого.

Тег <VIDEO> призначений для розміщення відеоконтенту на сторінці. Для коректного відображення контенту варто прописати додатково атрибути width, height, src, controls. Тег додає, відтворює і управляє настройками аудіозаписів на web-сторінці.

Тег <AUDIO> додає, відтворює і управляє настройками аудіозаписів на web-сторінці.

Тег <CANVAS> дозволяє малювати на web-сторінках довільні фігури за допомогою JavaScript (або інших клієнтських скриптів).

Під час створення елемента <CANVAS> необхідно задати атрибут id (визначає ім'я елемента для доступу до нього зі скриптів) і його розміри за допомогою атрибутів width і height.

Тег <FIGURE> використовується спільно з тегом <figcaption> і дозволяє групувати будь-які елементи, наприклад, зображення і підписи до них.

Тег <FIGCAPTION> має розташовуватися на самому початку або в самому кінці вмісту тегу <figure>.

Тег <HGGROUP> дозволяє згрупувати заголовки. Цей тег зазвичай використовується, коли заголовок має підзаголовки.

Тег <TIME> позначає текст, вміщений у нього, як дату, час або обидва значення одночасно. Може вказуватися безпосередньо всередині контейнера <TIME> або задаватися через атрибут datetime.

Практичне виконання роботи

Завдання 16.1. Форматування блочних елементів на web-сторінці

Створіть web-сторінку з блоковими елементами та відформатуйте їх, використовуючи внутрішні та вбудовані стилі CSS.

Вирішення

1. У редакторі Notepad++ створіть файл *Блок.html*.
2. Використовуючи раніше наведені приклади (рис. 6.16.2 – 6.16.4), послідовно (шляхом накопичення) відпрацюйте особливості форматування блоків на web-сторінці.

Завдання 16.2. Розміщення блоків на web-сторінці

Створіть web-сторінку та відпрацюйте різні способи відображення блокових елементів.

Вирішення

1. У редакторі Notepad++ створіть файл *Моделі позиціонування.html* (рис. 6.16.5 та рис. 6.16.6).

Код HTML	<pre><div style="border: solid red 3px;"> </div> <div style="border: solid black 3px; background-color:lightgreen;"> </div> <p style="text-align:justify; font-size: 30px;">Hypertext Markup Language (HTML) (укр. Мова розмітки гіпертекстових документів)–стандартна мова розмітки для створення web-сторінок і web-додатків. З Cascading Style Sheets (CSS) і JavaScript, вона утворює триаду основних технологій для World Wide Web. Web-браузери отримують HTML-документи з web-сервера або з локальної пам'яті і передають документи в мультимедійні web-сторінки. HTML описує структуру web-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа.</p></pre>
----------	--

Рис. 6.16.5. Файл "Моделі позиціонування.html" (код)

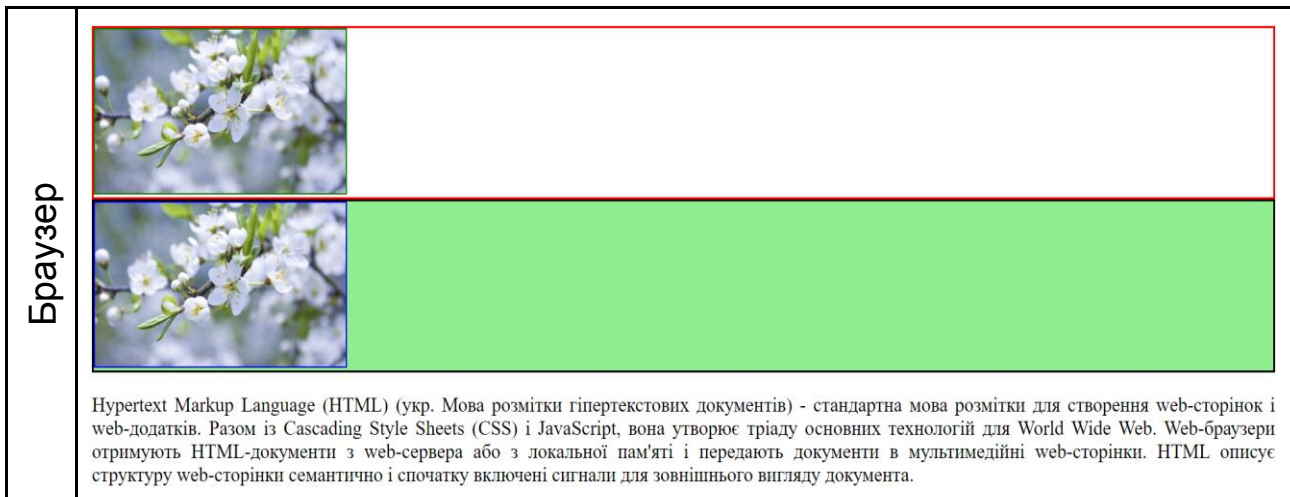


Рис. 6.16.6. Файл "Моделі позиціонування.html" (браузер)

2. Зробіть копію текстового абзаца і вставте на сторінку 7 – 8 разів, щоб з'явилася смуга прокручування.

3. Послідовно, змінюючи властивості блоків (рис. 6.16.7 – 6.16.10), проаналізуйте "поведінку" блокових елементів.

Код HTML	<pre><div style="border: solid red 3px; <u>position:relative; top:80px; left:80px;</u>"> </div> <div style="border: solid black 3px; background-color:lightgreen; <u>position:relative; top:-80px; left:-80px;</u>"> </div></pre>
----------	--

Рис. 6.16.7. Файл "Моделі позиціонування.html (position:relative;)"

Код HTML	<pre><div style="border: solid red 3px; <u>position:relative; top:80px; left:80px;</u>"> </div> <div style="border: solid black 3px; background-color:lightgreen; <u>position:fixed; right:100px; bottom:100px;</u>"> </div></pre>
----------	---

Рис. 6.16.8. Файл "Моделі позиціонування.html (position:fixed;)"

Код HTML	<pre> <div style="border: solid red 3px; position: absolute; top: 80px; left: 80px;"> </div> <div style="border: solid black 3px; background-color: lightgreen; position: relative; top: -80px; left: -80px;"> </div> </pre>
----------	---

Рис. 6.16.9. Файл "Моделі позиціонування.html (position: absolute;)"

Код HTML	<pre> <div style="border: solid red 3px; position: fixed; bottom: 480px; right: 80px; z-index: 10;"> </div> <div style="border: solid black 3px; background-color: lightgreen; position: relative; left: 100px; top: 40px; z-index: 8;"> </div> </pre>
----------	---

Рис. 6.16.10. Файл "Моделі позиціонування.html (z-index;)"

Завдання 16.3. Створення спадаючого меню

Створіть на web-сторінці меню зі спадаючими елементами.

Вирішення

1. У редакторі Notepad++ створіть файли *Меню.html* та *menu.css*.
2. Використовуючи приклади HTML-коду (рис. 6.16.11) та CSS-коду (рис. 6.16.12), створіть спадаюче меню.

Код HTML	<pre> <ul class="menu"> Головна Афіша Розваги Кінотеатр Боулінг Каток Кафе і ресторани Галерея Бронювання Контакти </pre>
----------	---

Рис. 6.16.11. Спадаюче меню (HTML-код)

```

Body{
  background-color: lightgreen; /*Фон для сторінки*/
  margin: 0; } /* Прибираем відступи на веб-сторінці*/
.menu { /*Задаем відступи і розміри меню*/
  background-color: lightgreen; /*фон випадального меню*/
  margin-top: 100px;
  margin-left: 300px;
  min-width: 500px; } /*Мінімальна ширина меню*/
.menu li{
  list-style: none; /*Прибираем маркери в маркованому списку*/
  float: left; } /*Маркований список розміщується в рядок*/
.menu li ul{
  display: none; /*Приховуємо вложений маркерований список*/
  position: absolute; /*вложений <ul> випадає із загального потоку*/
  padding: 0; }
.menu li ul li {
  float: none; }
.menu li a{
  color: #FFF; /*Колір гіперпосилань*/
  text-decoration: none; /*Прибраємо підкреслювання у гіперпосиланнях*/
  text-align: center;
  display: block; /*Гіперпосилання задається як блоковий елемент*/
  padding: 5px;
  width: 100px; }
.menu ul li a{
  color: #000; /*Колір випадальних гіперпосилань*/
  text-decoration: none;
  text-align: center;
  display: block;
  padding: 5px;
  width: 100px; }
.menu li: hover ul{
  display: block; } /*При наведенні миші гіперпосилання стають блоковими*/
.menu li: hover {
  background-color: yellow; /*Зміна фону при наведенні миші*/
  color: red; }
.menu ul li a: hover{
  color: red; } /*Колір випадальних гіперпосилань*/

```

Рис. 6.16.12. Спадаюче меню (CSS-код)

Індивідуальні завдання

Створіть макет web-сторінки, використовуючи прийоми блокового верстання, відповідно до наведених варіантів.

Порядок виконання:

1. У процесі створення макету web-сторінки чітко дотримуйтеся принципу поділу структури (із застосуванням HTML) і форматування (із застосуванням CSS).




2. Створіть на сторінці батьківський блок із класом *wrapper*.

3. Всередині цього блока створіть логічну структуру сторінки, використовуючи контейнери: `<header></header>`, `<footer></footer>`, `<nav></nav>`, `<main></main>`, `<aside></aside>`, `<section></section>` та ін.

4. Використовуйте контейнер `<div></div>` з відповідною назвою класу для розміщення інформації у кожному зі структурних елементів.

5. Рисунки, текст, елементи форми з відповідним форматуванням також мають бути розміщені у середині відповідного блока.




Варіант 1

Назва сайту		Пошук по сайту		Знайти	
Горизонтальне спадаюче меню					
Додаткова інформація		Рисунок за темою сайту		Основний контент	
Основний контент		Основний контент		Основний контент	
©	Адреса	Телефони			




Варіант 2

Логотип		Назва сайту			
Вкажіть Вашу стать: <input checked="" type="radio"/> чоловіча <input type="radio"/> жіноча		Основний контент		Вертикальне спадаюче меню	
		Основний контент			
		Основний контент			
© Ім'я розробника		Адреса		Телефони с логотипами операторів	




Варіант 3

Назва сайту							
Текстове поле: Введіть текст		Анотація сторінки					
Вертикальне спадаюче меню		Фонове зображення за темою сайту		Основний контент		Основний контент	
Рисунок за темою сайту		Телефони с логотипами операторів					
© Адреса							




Варіант 4

Назва сайту			Логотип		
Анотація сторінки					
Рисунок за темою сайту	Основний контент	Основний контент	Багаторядкове текстове поле: <div style="border: 1px solid #ccc; height: 40px; width: 100%;"></div>		
Горизонтальне спадаюче меню					
©	Адреса	Телефони			

Варіант 5

Назва сайту		Горизонтальне спадаюче меню			
Рисунок за темою сайту		Анотація сторінки			
<div style="border: 1px solid #ccc; padding: 5px;"> понеділок вівторок середа четвер п'ятниця субота неділя </div>	Додаткова інформація	Основний контент	Основний контент		
©	Адреса	Телефони			

Варіант 6

Логотип		Назва сайту	
<div style="border: 1px solid #ccc; padding: 5px;"> Знання мов <input checked="" type="checkbox"/> українська <input type="checkbox"/> англійська <input type="checkbox"/> німецька <input type="checkbox"/> іспанська </div>		Основний контент	
		Основний контент	
		Основний контент	
Горизонтальне спадаюче меню			
© Ім'я розробника			


Варіант 7

Назва сайту				
Телефони с логотипами операторів		Вертикальне спадаюче меню		
Анотація сторінки	Фонове зображення за темою сайту	Основний контент		
		Основний контент		
		Основний контент		
Рисунок за темою сайту	Текстове поле: <input type="text" value="Введіть текст"/>		© Адреса	




Варіант 8

Логотип	<input type="text" value="Пошук по сайту"/>	<input type="button" value="Знайти"/>
Назва сайту		
Горизонтальне спадаюче меню		
Вкажіть Вашу стать: <input checked="" type="radio"/> чоловіча <input type="radio"/> жіноча	Основний контент	Фонове зображення за темою сайту
	Основний контент	
	Основний контент	
© Ім'я розробника	Адреса	Телефони с логотипами операторів

Варіант 9

Назва сайту		Логотип	
Анотація сторінки			
Горизонтальне спадаюче меню			
Рисунок за темою сайту	Основний контент	Основний контент	Основний контент
©	Адреса	Телефони	  

Варіант 10

Назва сайту		Горизонтальне спадаюче меню	
Рисунок за темою сайту		Анотація сторінки	
Реклама	Основний контент		
			
			
©	Адреса	Телефони	Знання мов <input checked="" type="checkbox"/> українська <input type="checkbox"/> англійська <input type="checkbox"/> німецька <input type="checkbox"/> іспанська

Запитання для самоконтролю

1. Назвіть особливості блокових елементів.
2. У чому відмінність блокових та вбудованих (рядкових) елементів?
3. Чим відрізняються відступи, задані за допомогою властивостей *padding* та *margin*?
4. Яким чином формуються межі блокових елементів?
5. Що таке "схлопування" зовнішніх відступів блоків?
6. Яким чином вирівнюються на сторінці блокові елементи? Наведіть приклад.
7. Яким чином можна заборонити обтікання блокових елементів? Наведіть приклад.
8. Назвіть особливості використання властивості *overflow*.
9. Яким чином блоковий елемент можна перетворити в рядковий і навпаки? Наведіть приклади.
10. Назвіть особливості статичного позиціонування блокових елементів.
11. Назвіть особливості відносного позиціонування блокових елементів.
12. Назвіть особливості фіксованого позиціонування блокових елементів.
13. Назвіть особливості абсолютного позиціонування блокових елементів.
14. Для чого використовується властивість *z-index*?

17. Адаптивна верстка сайту. Створення сайтів з використанням технології Bootstrap

Мета: вивчити особливості адаптивної верстки сайту. Ознайомитися з технологією Bootstrap.

Ключові терміни: адаптивна верстка, flexbox, медіазапити, Bootstrap.

Теоретичні відомості

Адаптивний web-дизайн – це дизайн web-сторінок, що забезпечує оптимальне відображення та взаємодію сайту з користувачем незалежно від роздільної здатності та формату пристрою, з якого здійснюється перегляд сторінки.

Основні причини виникнення адаптивної верстки сайтів:

- збільшення кількості різних пристроїв, з яких є можливість виходу в мережу "Інтернет";
- популярність мобільних пристроїв із можливістю виходу в мережу "Інтернет";
- рекомендації пошукових систем (відсутність дублюючих сторінок і т. д.).

Переваги адаптивної верстки:

1. Загальний контент для всіх версій сайту.
2. Загальний HTML-код для всіх версій.
3. Простота оновлення сайту.
4. Одна адреса для всіх версій сайту, що набагато полегшує пошук та індексацію інформації.

Недоліки адаптивної верстки:

1. Адаптивний дизайн і верстка коштують дорожче.
2. Файли більше "важать".
3. HTML і CSS містять код усіх версій.
4. Процес створення сайту займає набагато більше часу.

Адаптивний макет сайту – це макет, який може "приспосовуватися" під різні пристрої (ширину робочої області вікна браузера). Тобто на одних пристроях він може мати одну структуру, а на інших – іншу (рис. 6.17.1).

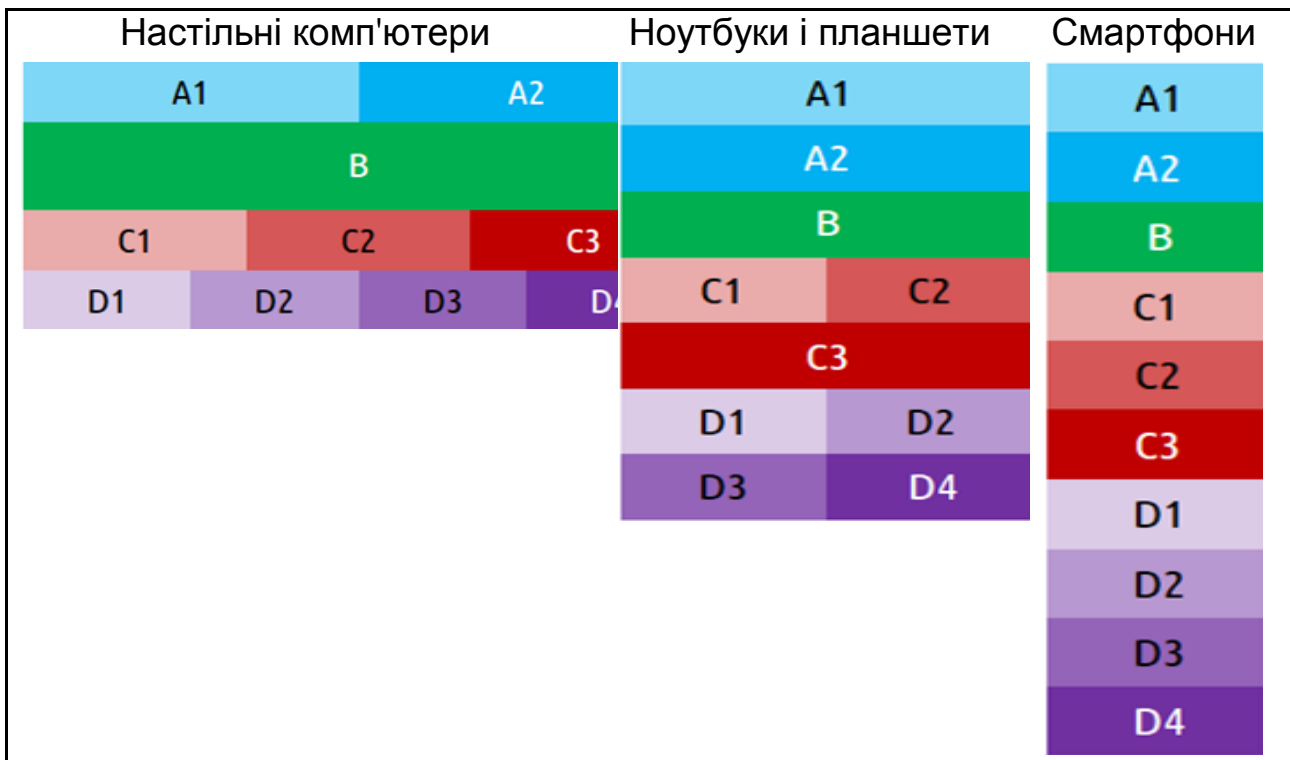


Рис. 6.17.1. Відображення адаптивного сайту на різних пристроях

Розглянемо декілька підходів для створення адаптивного макета сайту:

- flexbox-верстання;
- використання медіазапитів;
- технологія Bootstrap.

Найчастіше використовують комбінацію раніше вказаних підходів. Зупинимося детальніше на кожному із них.

Технологія flexbox

Flexbox – це відносно новий режим розмітки у CSS3. Він дозволяє автоматично організовувати відповідні елементи в контейнері залежно від розміру екрана (або пристрою) та дає змогу використовувати більш ефективний спосіб верстки, вирівнювання й розподілу вільного місця між елементами у контейнері, навіть коли їхній розмір невідомий і/або динамічний.

У flexbox є два основних типи елементів: гнучкий батьківський контейнер (flex-container) і його дочірні елементи – гнучкі елементи (flex-item). Для ініціалізації flex-контейнера досить встановити із використанням CSS батьківському елементу властивість display: flex; або display: inline-flex;.

Різниця між flex і inline-flex полягає лише у принципі взаємодії з елементами, що оточують контейнер, подібно display: block; і display: inline-block; відповідно. Цю властивість потрібно *встановити тільки для батьківського контейнера* і всі його безпосередні дочірні елементи автоматично стануть flex-елементами.

Усередині гнучкого контейнера створюються дві осі (рис. 6.17.2), головна вісь (main-axis) і перпендикулярна або крос-вісь (cross-axis). Переважно гнучкі елементи шикуються саме по головній осі, а потім уже по крос-осі. За замовчуванням головна вісь горизонтальна і має напрямок зліва направо, а крос-вісь вертикальна і спрямована зверху вниз.

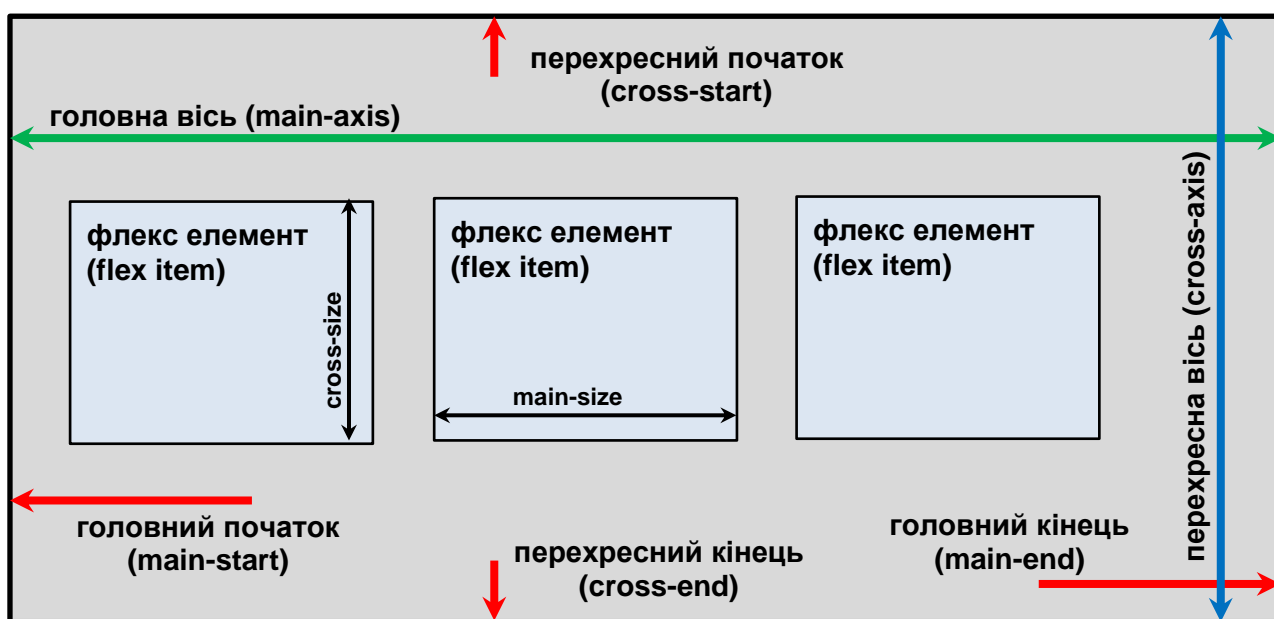


Рис. 6.17.2. Структура флекс-контейнера

Головний початок / головний кінець (main-start / main-end) – флекс-елементи додаються до контейнера, який починається з боку головного початку і закінчується до головного кінця.

Перехресний початок і кінець (cross-start/end) – флекс-лінії заповнюються items (елементами) і поміщаються в контейнер, який починається з боку перехресного початку flex-контейнера та йде до перехресного кінця.

Основна характерна риса флекс-контейнера – це здатність змінювати ширину або висоту своїх дочірніх елементів, щоб якомога краще заповнити доступну область на екранах різної ширини.

Розглянемо можливості flexbox для двох груп:

- flex-контейнера (class="flex-container");
- flex-елементів (class="flex-item").

Властивості flexbox для батьківського контейнера

Властивість **flex-direction** вказує, як flex-елементи викладаються у flex-контейнері. Вони можуть викладатися горизонтально, як рядки (рис. 6.17.3) і вертикально, як колонки (рис. 6.17.4):

row – flex-елементи укладаються у ряд зліва направо (за замовчуванням);

row-reverse – flex-елементи укладаються в ряд справа наліво;



Рис. 6.17.3. Властивість **flex-direction: row**;

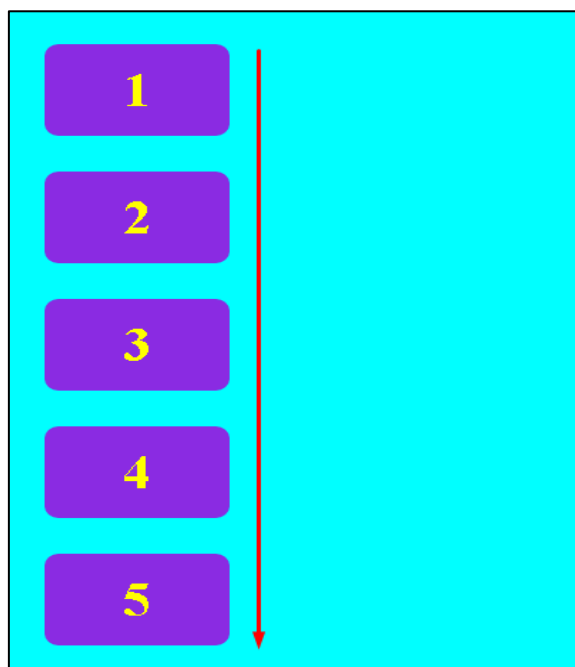


Рис. 6.17.4. Властивість **flex-direction: column**;

column – flex-елементи укладаються у колонку зверху вниз;

column-reverse – flex-елементи укладаються в колонку знизу вверху.

Властивість **justify-content** вирівнює flex-елементи по головній осі поточного рядка flex-контейнера. Вона ефективно розподіляє вільний простір батьківського контейнера.

`flex-start` – flex-елементи вирівнюються по лівій стороні flex-контейнера (за замовчуванням);

`flex-end` – flex-елементи вирівнюються по правій стороні flex-контейнера;

`center` – flex-елементи вирівнюються центру flex-контейнера (рис. 6.17.5);



Рис. 6.17.5. Властивість **justify-content: center**;

`space-between` – flex-елементи відображаються з рівним інтервалом між ними, а перший і останній flex-елементи вирівнюються по краях flex-контейнера (рис. 6.17.6);



Рис. 6.17.6. Властивість **justify-content: space-between**;

`space-around` – flex-елементи рівномірно розподіляються по всьому рядку, водночас порожній простір між першим і після останнього елемента дорівнює половині між сусідніми елементами в контейнері (рис. 6.17.7).



Рис. 6.17.7. Властивість **justify-content: space-around**;

Властивості flexbox для flex-елементів

Властивість **order** визначає порядок, у якому дочірні елементи flex-контейнера розміщуються всередині нього. За замовчуванням, flex-елементи слідують один за одним по мірі додавання їх у flex-контейнер у HTML-коді. Значення за замовчуванням дорівнює 0. Наприклад (рис. 6.17.8):

```
<div class="flex-item" style="order:-1;">4</div>
```



Рис. 6.17.8. Властивість **order**

Flex-елементи можуть бути впорядковані за допомогою властивості **order** без зміни структури HTML-коду.

Властивість **flex-grow** вказує фактор розтягування flex-елемента, який визначає, наскільки буде розтягнутий flex-елемент щодо решти flex-елементів у контейнері під час розподілу вільного позитивного простору.

Значення за замовчуванням дорівнює 0. Наприклад (рис. 6.17.9):

```
<div class="flex-item" style="flex-grow:3;">4</div>
```



Рис. 6.17.9. Властивість **flex-grow**

Якщо всім flex-елементам задано одне і теж значення **flex-grow**, тоді всі елементи у контейнері будуть однакового розміру.

Властивість **flex-shrink** визначає, наскільки буде звужений flex-елемент щодо решти flex-елементів у flex-контейнері під час розподілу негативного вільного простору. Значення за замовчуванням дорівнює 1. Наприклад (рис. 6.17.10):

```
<div class="flex-item" style="flex-shrink:0;">4</div>
```



Рис. 6.17.10. Властивість **flex-shrink**

За замовчуванням flex-елементи можуть бути звужені, але якщо ми встановимо значення `flex-shrink`, що дорівнює 0 (не звужуватися), то елемент збереже первинний розмір.

Медіазапити

Медіазапити – це окремі набори правил у файлі стилів CSS. Кожен із них доповнює набір правил і відповідає за той чи інший спосіб відображення сайту.

Медіазапити містять медіатип (принтери, смартфони, екрани, телевізори, проєктори тощо) та умови, які можуть набувати у свою чергу значень "істина" або "неправда" (`true`, `false`). Залежно від того, чи правильний медіатип і виконується умова, будуть застосовуватися різні стилі CSS. Якщо значення буде `true`, то будуть застосовуватися ті стилі, які прописані в цьому медіазапиті, якщо ж значення буде `false`, то будуть застосовуватися звичайні стилі CSS.

Завдяки таким запитам і створюються різні відображення сайту для мобільних, планшетів і екранів моніторів. Медіазапити підтримуються всіма сучасними браузерами.

Сторінки, адаптовані для перегляду на різних пристроях, мають містити в контейнері `<HEAD>` метатеґ `viewport`. Він повідомляє браузеру, яким чином потрібно контролювати розміри і масштаб сторінки:

```
<meta name="viewport" content="width=device-width,  
initial-scale=1.0">
```

Перший параметр `width` дає команду браузеру адаптувати ширину сторінки до пристрою, на якому вона відкрита. Масштабування ширини сторінки визначається автоматично, якщо вказано значення `"device-width"`.

Параметр *initial-scale* означає коефіцієнт масштабування. Якщо вказано значення "1", розширення CSS і пристрою прирівнюються один до одного незалежно від розміру екрана.

Синтаксис медіазапитів:

1. Формування медіазапиту всередині файлу CSS:

`@media тип_пристрою and|or|not|only (медіаособливості)
{CSS код;}`.

2. Підключення за допомогою атрибута `media` тегу `link`:

`<link rel="stylesheet" media="тип_пристрою and| or|not|only
(медіаособливості)" href="style.css">`.

3. Імпорт файлу CSS:

`@import url(example.css) тип_пристрою and|or|not|only
(медіаособливості).`

Можливі значення типу пристроїв, що вказуються у медіазапиті, наведено в табл. 6.17.1.

Таблиця 6.17.1

Можливі значення типу пристроїв у медіа-запитах

Тип	Опис
all	Використовується для всіх типів пристроїв
aural	Використовується для синтезаторів мови і звуку
braille	Використовується для тактильного зворотного зв'язку пристроїв Брайля
embossed	Використовується для принтерів Брайля
handheld	Використовується для невеликих або портативних пристроїв
print	Використовується для принтерів
projection	Використовується для презентацій
screen	Використовується для екранів комп'ютерів, планшетів, смартфонів і т. д.
speech	Використовується для мовних браузерів
tty	Використовується для носіїв, що мають сітки з фіксованим кроком символів, такі як телетайпи і термінали
tv	Використовується для телеекранів

Можливі значення медіаособливостей пристроїв, що вказуються в медіазапиті, наведено в табл. 6.17.2.

Можливі значення медіаособливостей пристроїв у медіазапитах

Особливість	Опис
device-height	Вказує висоту пристрою (весь екран або сторінку, а не тільки область перегляду). Може мати префікс min або max
device-width	Вказує ширину пристрою (весь екран або сторінку, а не тільки область перегляду). Може мати префікс min або max
orientation	Вказує, в якому режимі перегляду знаходиться дисплей: ландшафтному або портретному
width	Вказує висоту області перегляду, наприклад, вікно браузера. Може мати префікс min або max
height	Вказує ширину області перегляду, наприклад, вікно браузера. Може мати префікс min або max

Можливі значення логічних операторів, що вказуються в медіазапиті, наведено в табл. 6.17.3.

Таблиця 6.17.3

Можливі значення логічних операторів у медіа-запитах

Оператор	Опис
and	Логічне І. За допомогою нього можна задати кілька умов і тільки за умови, що всі вони будуть виконані, тоді стиль буде застосований
or	Логічне АБО. Необхідно, щоб хоча б одна з умов виконувалася. Якщо кома (,) використовується як оператор or, то кожен запит розглядається індивідуально
not	Логічне заперечення НІ. Умова, що знаходиться за not, має виконуватися
only	Приховати правило @media від старих браузерів. Нові браузери просто не помітять цей оператор

Приклади медіазапитів

1. Медіазапит для задавання стилів за умови, що ширина екрана буде 700px і менше:

```
@media screen and (max-width:700px){
    p {color: red;}
}
```


2. Медіазапит для задавання стилів за умови, що ширина екрана буде знаходитися у діапазоні між 500px і 700px включно:

```
@media screen and (min-width:500px) and (max-width:700px) {  
    P {color: green;}  
}
```

Технологія Bootstrap

Bootstrap – це CSS/HTML фреймворк (каркас) для створення сайтів. Інакше кажучи, це набір інструментів для верстки. Охоплює HTML- і CSS-шаблони оформлення для типографіки, web-форм, кнопок, міток, блоків навігації та інших компонентів web-інтерфейсу.

По суті, якщо говорити простою мовою, це файл або декілька файлів із готовим написаним кодом, які підключаються до сайту в контейнері <HEAD>, після чого стає можливим використання можливостей цього фреймворку.

Основні інструменти Bootstrap:

сітки – заздалегідь задані розміри колонок, які можна відразу ж використовувати у CSS-описі документа;

шаблони – фіксований або "гумовий" шаблон документа;

типографіка – опис шрифтів, визначення деяких класів для шрифтів, таких як код, цитати і т. п.;

медіа – надає деяке упорядкування зображень та відео;

форми – класи для оформлення форм і деяких подій, що відбуваються з ними;

навігація – класи оформлення для панелей, вкладок, переходу по сторінках, меню і панелі інструментів;

алерт – оформлення діалогових вікон, підказок і спливаючих вікон.

Сітка Bootstrap – це основна частина фреймворку. Вона призначена для створення адаптивних шаблонів. Сітка Bootstrap ґрунтується на CSS Flexbox і медіазапитах.

Кожен медіазапит у Bootstrap будується на підставі мінімальної ширини viewport браузера. У Bootstrap ключове значення ширини viewport у медіазапиті називається breakpoint (контрольною крапкою, або більш просто класом пристрою відображення). Контрольні крапки мають свої позначення (табл. 6.17.4). Bootstrap має чутливу сітку, яка масштабується у 12 колонок відповідно до розміру пристрою або до розміру браузера.

Назви класів пристрою відображення у Bootstrap

Ширина viewport браузера	Контрольна крапка	Назва класу	Ширина адаптивного блока
Крихітна ($\geq 0px$)	без позначення (xs)	col-?	від col-1 до col-12
Маленька ($\geq 576px$)	sm	col-sm-?	від col-sm-1 до col-sm-12
Маленька ($\geq 768px$)	md	col-md-?	від col-md-1 до col-md-12
Велика ($\geq 992px$)	lg	col-lg-?	від col-lg-1 до col-lg-12
Дуже велика ($\geq 1200px$)	xl	col-xl-?	від col-xl-1 до col-xl-12

Підготовчий етап роботи з Bootstrap

1. Зайдіть на сайт розробника <http://getbootstrap.com>.
2. Скачайте архів Bootstrap, натиснувши кнопку Download.
3. Розархівуйте скачаний архів і перемістіть папки css та js у папку сайту (рис. 6.17.11).



Рис. 6.17.11. Папка сайту

4. У папках css та js залиште файли bootstrap.min.css, bootstrap.min.js, а решту файлів можна видалити. Версії min – це стислі версії, в яких вирізані пробіли і переноси рядків, відповідно, вони займають менше місця, що дозволить зменшити розмір сайту.

5. На сайті getbootstrap.com у закладці Getting started (де ми і скачували архів Bootstrap) виберіть посилання Starter templates. Скопіюйте HTML-код і вставте його в порожній файл index.html, а також в усі наступні проекти, в яких ви будете використовувати Bootstrap.

6. Уточніть шлях до файлів bootstrap.min.css та bootstrap.min.js з урахуванням вашої папки сайту (рис. 6.17.12).

```

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="css/bootstrap.min.css">

  <title>Bootstrap</title>
</head>
<body>
  <h1>Верстка в Bootstrap</h1>

  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
integrity="sha384-J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwjlyYfoRSJoz+n"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
  <script src="js/bootstrap.min.js"></script>

</body>

```

Рис. 6.17.12. Підготовка HTML-файлу для роботи з Bootstrap

Підключення скриптів доцільно задавати у кінці HTML-коду для скорішого завантаження сторінки.

Послідовність створення web-сторінки в Bootstrap

1. Для правильного вирівнювання та відступів рядки мають розміщуватися в межах `.container` (щоб мати фіксовану ширину) або `.container-fluid` (щоб мати повну ширину).

2. Верстання здійснюється рядками, кожен рядок містить 12 стовпців.

3. Батьківському елементу стовпців задається клас `row`, а класи стовпців починаються із префікса `col-`, потім йде розмір екрана (`sm`, `md`, `lg`, `xl`), а закінчуються відносною шириною стовпця. Наприклад: `.col-lg-4`.

4. Якщо в одному рядку вміщено більше ніж 12 стовпців, кожен додатковий стовпець буде перенесено, як єдине ціле, на новий рядок.

5. Класи сітки застосовуються до пристроїв із шириною екрана, що більша або дорівнює контрольній точці, яка встановлена для цього класу; і в той же час, класи сітки, призначені для пристроїв із вузьким екраном, ігноруються. Таким чином, застосування будь-якого класу `.col-md-` до певного елемента буде впливати на його стилі не лише на пристроях середнього розміру, але також і на пристроях із широким екраном, якщо клас `.col-lg-` не представлено.

Використовуючи готові шаблони Bootstrap, можна швидко створити адаптивний сайт. На рис. 6.17.13 наведено приклад створення адаптивного заголовка сайту з логотипом та меню Bootstrap.

```
<body>
  <div id="wrapper"> <!-- Створюємо обгортку для сайту -->
    <header> <!-- Позначаємо заголовок сайту -->
      <div class="center"> <!-- Обгортка по центру -->
        <div class="container-fluid"> <!-- HTML-размітка bootstrap для адаптивного контейнера; -->
          <div class="row"> <!-- Створюємо рядок bootstrap -->
            <div class="col-12 col-lg-3"> <!-- Задаємо адаптивні колонки bootstrap -->
              <strong class="logo blocks_wrap"><a href="#">Bootstrap</a></strong> <!-- Логотип -->
            </div>
            <div class="col-12 col-lg-9"> <!-- Колонки для меню -->
              <nav class="blocks_wrap"> <!-- Створюємо блок навігації -->
                <div class="burger_menu blocks_wrap"></div> <!-- Задаємо класс для адаптивного меню -->
                <ul>
                  <li class="active"><a href="#">Головна</a></li> <!-- Класс для активного меню -->
                  <li><a href="#">Колекції</a></li>
                  <li><a href="#">Галерея</a></li>
                  <li><a href="#">Блог</a></li>
                  <li><a href="#">Контакти</a></li>
                </ul>
              </nav> <!-- Закінчення навігації -->
            </div>
          </div>
        </div>
      </header> <!-- Закінчення header -->
    </div>
  </body>
```

Рис. 6.17.13. Приклад HTML-файлу з використанням Bootstrap

Сітку Bootstrap можна застосовувати окремо (без інших компонентів). Такий варіант застосування фреймворка теж використовується веб-розробниками, тому що не завжди готові макети елементів Bootstrap, такі як кнопки, навігаційні панелі, таблиці та інші компоненти, можна використовувати у рамках того чи іншого проєкту. Найчастіше це стосується проєктів з унікальним дизайном. У цьому випадку необхідні елементи простіше написати з нуля самостійно, замість того, щоб переписувати готові компоненти Bootstrap. У цьому випадку до проєкту достатньо підключити з архіву тільки файл "bootstrap-grid.min.css".

Практичне виконання роботи

Завдання 17.1. Форматування контейнерних блоків на web-сторінці за допомогою технології flexbox

Створіть web-сторінку з блоковими елементами та відформатуйте їх, використовуючи технологію flexbox.

Вирішення

1. У редакторі Notepad++ створіть файл *flexbox.html* відповідно до прикладу, наведеного на рис. 6.17.14.

```
<style>
    .flex-container {
        display: flex;
        background: #00ffff;
        padding: 15px;
        margin: 30px; }
    .flex-item {
        background: #8a2be2;
        font-size: 40px;
        color: #ffff00;
        margin: 15px;
        padding: 15px;
        border-radius: 10px;
        width: 10%;
        text-align: center;
        font-weight: bold; }
</style>
</head>
<body>
    <div class="flex-container">
        <div class="flex-item">1</div>
        <div class="flex-item">2</div>
        <div class="flex-item">3</div>
        <div class="flex-item">4</div>
        <div class="flex-item">5</div>
    </div>
</body>
```

Рис. 6.17.14. Фрагмент файлу flexbox.html

2. Послідовно змінюйте властивості flexbox для батьківського контейнера (.flex-container) та проаналізуйте поведінку дочірніх елементів, надаючи батьківському елементу властивостей та їхніх значень, які вказано на рис. 6.17.3 – 6.17.7:

flex-direction: значення;

justify-content: значення;

Завдання 17.2. Форматування дочірніх елементів на web-сторінці за допомогою технології flexbox

Відформатуйте дочірні блокові елементи, використовуючи технологію flexbox.

Вирішення

1. Використовуючи файл *flexbox.html*, відпрацюйте різні способи відображення flex-елементів, надаючи їм відповідних властивостей (див. рис. 6.17.8 – 6.17.10):

order: значення;

flex-grow: значення;

flex-shrink: значення;

Для надання властивостей дочірнім елементам використовуйте в HTML-кодi атрибут *style* для відповідного дочірнього елемента.

Завдання 17.3. Використання медіазапитів для форматування web-сторінки

Створіть web-сторінку з відповідним текстом та за допомогою медіазапитів задайте форматування за різних розмірів браузера.

Вирішення

1. У редакторі Notepad++ створіть файл *media.html*, у який вставте декілька текстових абзаців.

2. У контейнері <HEAD> підключіть метатег *viewport* та, використовуючи внутрішні стилі, сформууйте медіазапити для форматування web-сторінки за різних розмірів браузера (рис. 6.17.15).

3. Змінюючи розмір браузера, проаналізуйте результати форматування за допомогою медіазапитів.

4. Сформууйте декілька медіазапитів і підключіть їх за допомогою зовнішнього файлу *style.css*.

```
<head>
  <meta charset="UTF-8">
  <title>Медіа-запити</title>
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
  <style>
    @media screen and (max-width:900px)
    {
      p{color: blue;}
      body{background-color: #66CDAA;}
    }

    @media screen and (min-width: 500px) and (max-width: 700px)
    {
      p{color: green;}
      body{background-color: yellow;}
    }
  </style>
</head>
<body>
  <p>Медіа-запити - це окремі набори правил у файлі стилів CSS.
  Кожен з таких доповнює набір правил і відповідає за той чи інший
  спосіб відображення сайту.</p>
</body>
```

Рис. 6.17.15. Приклад використання медіазапитів

Завдання 17.4. Створення web-сторінки за допомогою технології Bootstrap

Створіть web-сторінку, використовуючи технологію Bootstrap.

Вирішення

1. Зайдіть на сайт розробника <http://getbootstrap.com>.
2. Скачайте архів Bootstrap, натиснувши кнопку Download.
3. Розархівуйте скачаний архів і перемістіть папки css та js у папку сайту (див. рис. 6.17.11).
4. У папках css та js залиште файли bootstrap.min.css, bootstrap.min.js, а решту видаліть.
5. На сайті getbootstrap.com у закладці Getting started виберіть посилання Starter templates, скопіюйте HTML-код і вставте його у порожній index.html.
6. Уточніть шлях до файлів bootstrap.min.css та bootstrap.min.js з урахуванням вашої папки сайту (див. рис. 6.17.12).

7. Створіть web-сторінку, використовуючи готові елементи Bootstrap (меню, кнопки та ін.). Використовуючи класи Bootstrap, передбачте адаптивність відображення сторінки за різних розмірів браузера (див. рис. 6.17.13).

Індивідуальні завдання

Створіть сайт, що складається з 5 – 6 сторінок.

Порядок виконання та основні вимоги:

1. Тема сайту має відповідати вашій майбутній професійній діяльності.
2. Зверстайте макети сторінок сайту, використовуючи прийоми блокового й адаптивного верстання та наповніть відповідним вмістом.
3. Здійсніть форматування web-сторінок тільки за допомогою стилів CSS.
4. Одна або декілька web-сторінок мають бути зверстані за допомогою технології Bootstrap.
5. Розмістіть сайт у мережі "Інтернет" на одному із безкоштовних хостингів.

Запитання для самоконтролю

1. Що таке "адаптивний web-дизайн"? Назвіть переваги та недоліки.
2. Які ви знаєте підходи до адаптивної верстки? Назвіть їхні особливості.
3. Що таке "flexbox-верстка"? Поясніть сутність та особливості.
4. Назвіть основні властивості батьківських елементів під час використання flexbox-верстки.
5. Назвіть основні властивості дочірніх елементів під час використання flexbox-верстки.
6. Що таке "медіазапити"? Наведіть варіанти підключення медіазапитів.
7. Для чого потрібний метатег *viewport*?
8. Що таке Bootstrap? Назвіть переваги та недоліки технології Bootstrap.
9. У чому полягає сутність адаптивної сітки Bootstrap?
10. Назвіть основні принципи верстки web-сторінки за допомогою Bootstrap.

Тестові завдання за розділом 6

1. Які види стилів існують у CSS:

- а) вбудовані в елемент HTML;
- б) вбудовані у HTML-документ;
- в) сумісні таблиці стилів;
- г) зовнішні таблиці стилів?

2. Розрізняють такі типи селекторів:

- а) селектори рисунків;
- б) універсальний;
- в) псевдо класів;
- г) id-селектори;
- д) групові селектори.

3. Для форматування тексту використовуються такі властивості:

- а) padding;
- б) title;
- в) font-size;
- г) color.

4. Для форматування фону використовуються такі властивості:

- а) background-color;
- б) background-table;
- в) background-image;
- г) background-body.

5. Для вирівнювання блока на сторінці використовується властивість:

- а) overflow;
- б) clear;
- в) style;
- г) float.

6. Розрізняють такі моделі позиціонування:

- а) static;
- б) relative;

- в) absolute;
- г) overflow.

7. Для управління шарами блоків використовується властивість:

- а) z-index;
- б) position;
- в) style;
- г) span.

8. До блочних елементів відносять:

- а) <div>;
- б) ;
- в) <p>;
- г) <a>.

9. Для створення меж блока використовують такі властивості:

- а) width;
- б) height;
- в) border;
- г) padding;
- д) margin.

10. Для створення відступів між блоками використовують властивість:

- а) margin;
- б) padding;
- в) clear;
- г) margin-left.

11. За допомогою якої властивості можна блоковий елемент перетворити у рядковий і навпаки:

- а) overflow;
- б) display;
- в) border;
- г) flex;
- д) float?

12. За допомогою якого тегу можна відформатувати текст усе-редині рядкового елемента:

- а) <display>;
- б) <forma>;
- в) ;
- г) <format>?

13. Для форматування рамки блока використовують такі властивості:

- а) background-color;
- б) border;
- в) border-style;
- г) margin.

14. Тег <nav> використовують для створення:

- а) заголовка сайту;
- б) підвалу сайту;
- в) навігаційної частини сайту;
- г) головної частини сайту.

15. Універсальний селектор розповсюджується:

- а) на всі елементи web-сторінки;
- б) на елементи, які мають цю властивість;
- в) тільки на блокові елементи;
- г) тільки на рисунки.

Розділ 7. Інформаційні системи створення, редагування та управління контентом сайту

Метою розділу є формування у майбутніх фахівців системи компетентностей із використання систем керування контентом для створення сайтів.

Професійні компетентності

У результаті вивчення навчальної дисципліни студент має:

знати:

поняття системи управління контентом;

основні методи просування сайтів в інтернеті, а також методи просування товарів та послуг;

основні різновиди реклами в інтернеті та особливості їхнього практичного використання;

вміти:

створювати власний сайт на платформі WordPress, розміщувати його в інтернеті та керувати ним.

Ключові терміни: CMS, WordPress, блог, управління контентом, семантичне ядро, статистика, ділення, авторизація, адміністрування, індексація, Google Search Console.

Теоретичні відомості

На сьогодні існують два типи сайтів. Перший тип – статичні сайти – зазвичай створюються в ручному режимі з використанням HTML, CSS та інших допоміжних технологій, як було розглянуто в розділах 5 та 6. До переваг розроблення таких сайтів можна віднести дешеве розроблення власними силами, дешеву технічну підтримку, незалежність сторінок одна від одної; меншу кількість необхідних технічних та серверних ресурсів. Головним недоліком є незручність оновлення таких сайтів, додавання нових сторінок або видалення старих зі збереженням коректних гіперпосилань, також необхідні знання HTML, CSS.

Динамічні сайти зазвичай створюються спеціальними мовами розроблення, наприклад, PHP, та розміщуються на сервері, який генерує

HTML-сторінку. До переваг таких сайтів можна віднести зручність керування; розподілення функцій різних користувачів на ролі; швидке та зручне оновлення матеріалів; швидка окупність розроблення; великий термін життя. Але цей тип є більш дорогим у розробленні та технічній підтримці порівняно зі статичним та більш вимогливим до ресурсів та програмного забезпечення сервера.

WordPress (WordPress.org) становить систему управління контентом (CMS – Content Management System), яка використовується для створення web-сайтів. Посилання на головний сайт – <https://wordpress.org/>, посилання на українську версію – <https://uk.wordpress.org/>. Для того щоб використовувати CMS WordPress, його потрібно скачати з офіційного сайту, завантажити на хостинг, налаштувати з'єднання з базою даних та інсталиувати на сервер.

CMS – це програмне забезпечення для створення web-сайтів чи інших інформаційних ресурсів в інтернеті. Фактично, це сайт у вигляді заздалегідь підготовлених сторінок, який користувач може налаштувати для себе.

Робота із WordPress.org потребує наявності деяких технічних знань щодо розроблення сайтів (HTML, PHP та ін.), розміщення їх на сервері (обрання хостингу, налаштування бази даних, доменне ім'я) та обслуговування (збереження даних, оновлення сайту, виправлення помилок, безпека тощо). Перевагою використання WordPress.org є, фактично, відсутність обмежень на використання будь-яких плагінів, редагування файлів, адміністрування і т. д.

WordPress.com – це хостинг-майданчик або сервіс, що дозволяє створити власний сайт без завантаження та встановлення його власноруч. Потрібно лише зареєструватися, після чого ви отримуєте інструменти для наповнення сайту контентом, обрання інтерфейсу (теми) та адміністрування. Обслуговування сайту, безпека, створення резервних копій – усі ці функції належать до відповідальності власників хостингу. WordPress.com є комерційним продуктом, тож тільки певний набір базових функцій є доступним безкоштовно. Для безкоштовних сайтів WordPress.com існують обмеження на розмір дискового простору, наявність реклами та обмеження на використання власної реклами, стандартне доменне ім'я. Детальний опис планів та цін, що пропонує WordPress.com, доступний за посиланням <https://wordpress.com/#plans>.

Багато додаткової та корисної інформації про роботу на WordPress.com можна знайти за посиланням <https://en.support.wordpress.com/>.

18. Система управління контентом сайту WordPress

Мета: познайомитися з CMS WordPress, познайомитись із web-сайтом WordPress.com, зареєструватися на web-сайті WordPress.com. Ознайомитися із процесом створення сайту на платформі WordPress.com, ознайомитися з елементами інтерфейсу для керування сайтом, опанувати додавання матеріалів на сайт. Ознайомитися з методами персоналізації та управління сайту. Ознайомитися з особливостями створення текстового контенту для мережі "Інтернет".

Ключові терміни: CMS, WordPress, блог, управління контентом, семантичне ядро.

Теоретичні відомості

Головна сторінка сайту WordPress.com знаходиться за посиланням <http://wordpress.com/> (рис. 7.18.1). Якщо прокрутити головну сторінку до кінця, в нижній частині можна знайти селектор мови, який зображено на рис. 7.18.2, однак українська відсутня в цьому переліку.

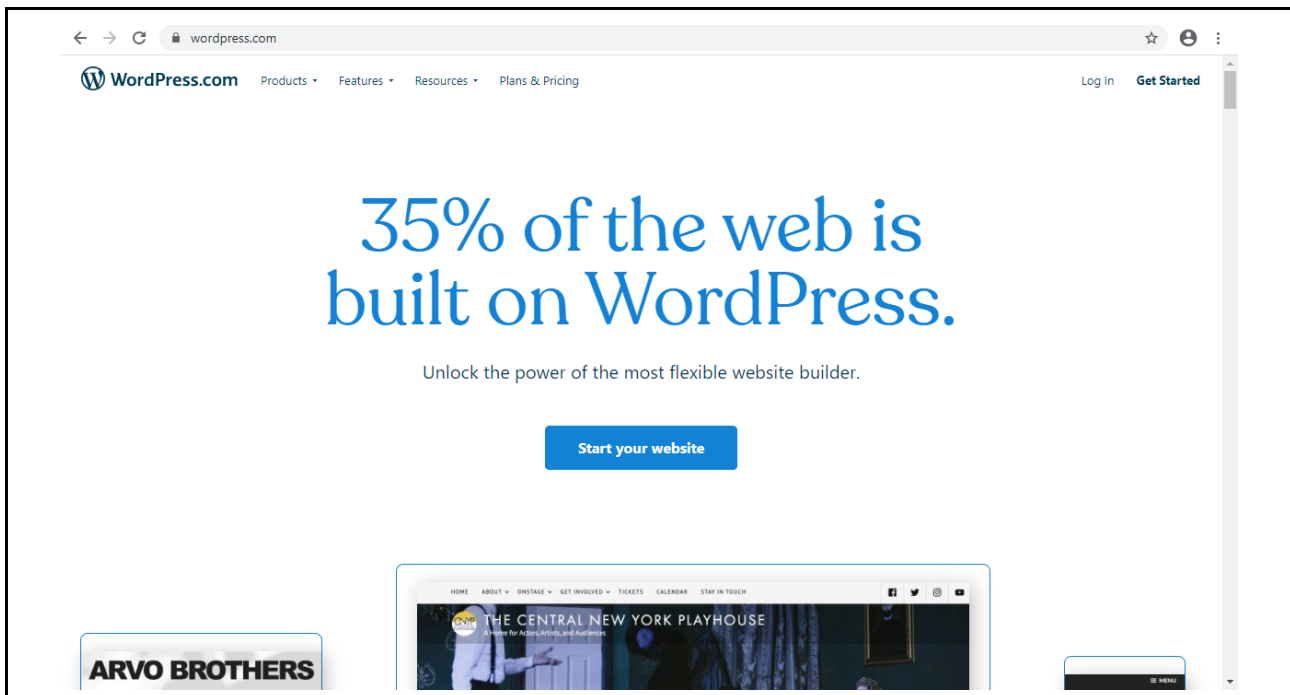


Рис. 7.18.1. Головна сторінка сайту <http://wordpress.com/>



Рис. 7.18.2. Селектор обрання мови сайту <http://wordpress.com/>

Для роботи на сайті WordPress.com та створення власного сайту потрібно зареєструватися за посиланням <https://wordpress.com/start/account/user> (рис. 7.18.3). Потрібно вказати свою електронну пошту (до якої ви маєте доступ), вигадати та коректно ввести ім'я користувача і пароль. Після цього потрібно натиснути кнопку Continue, якщо все виконано правильно, завантажиться сторінка, зображена на рис. 7.18.4. Вона говорить про те, що для продовження реєстрації потрібно підтвердити її, використовуючи вказану електронну пошту.

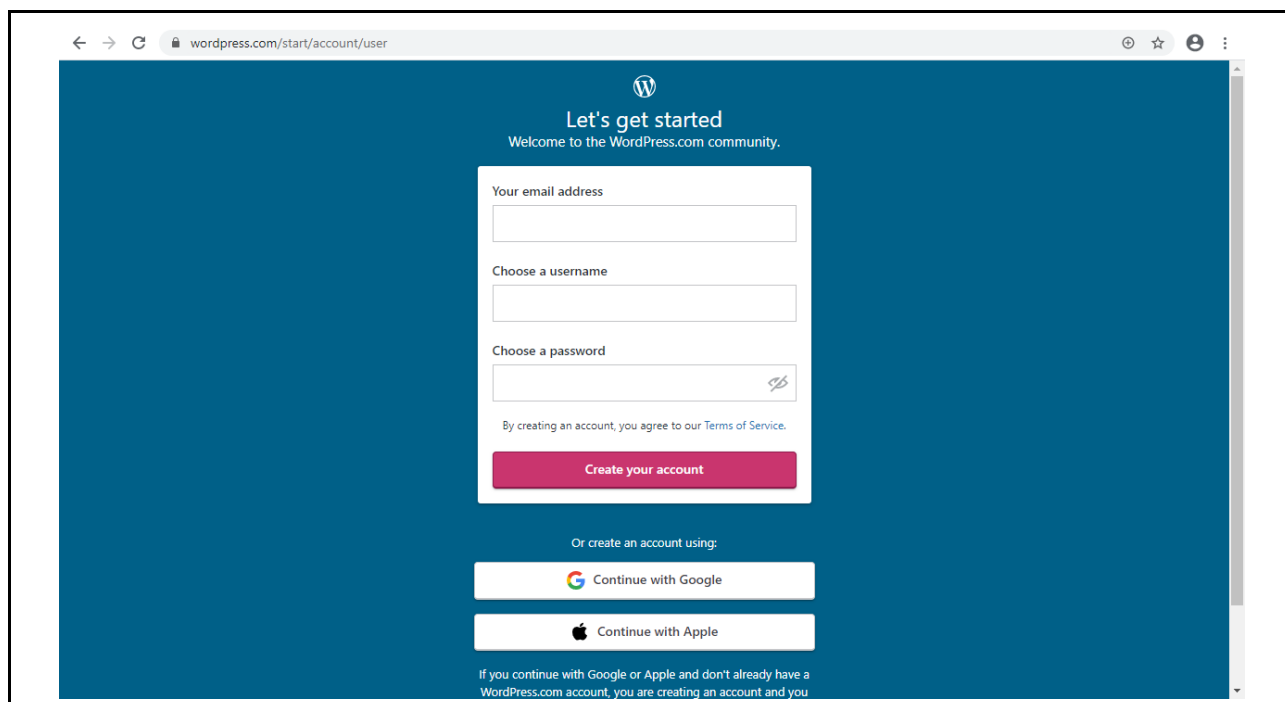


Рис. 7.18.3. Перший крок реєстрації на сайті

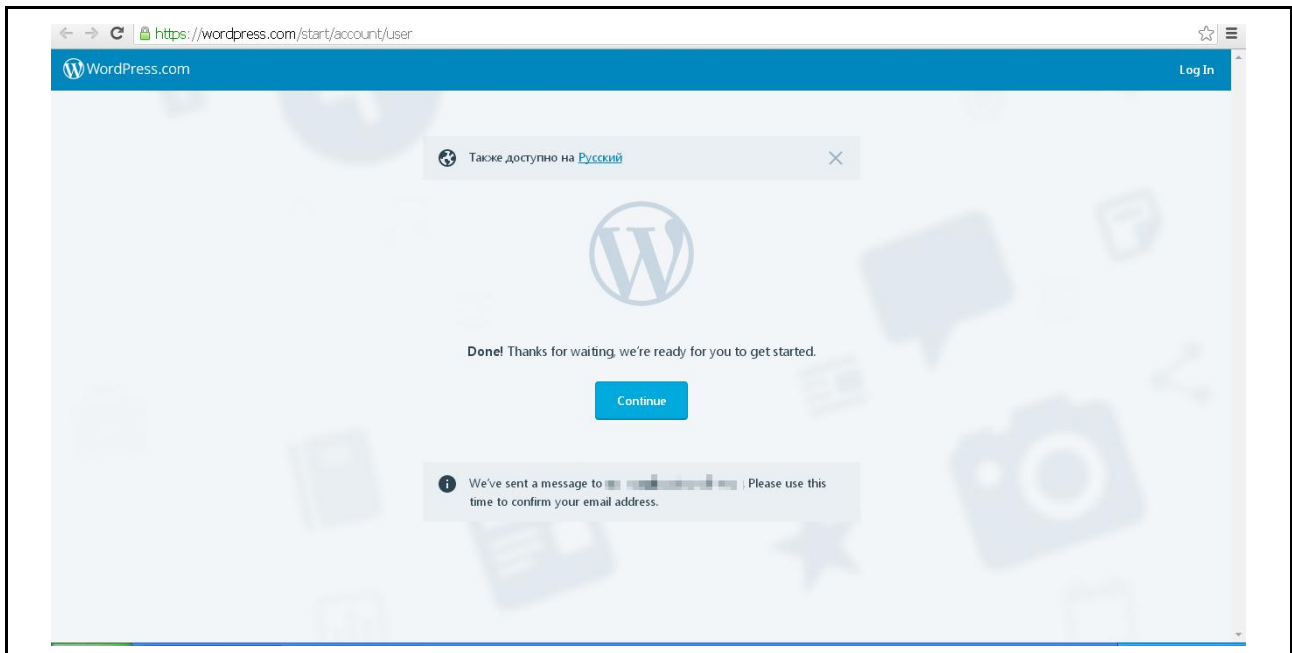


Рис. 7.18.4. Повідомлення про відправлення листа за вказаною раніше адресою електронної пошти

Треба відкрити свою електронну поштову скриньку, вказану на попередньому кроці під час реєстрації, та підтвердити її адресу, клацнувши кнопку Confirm now (рис. 7.18.5).

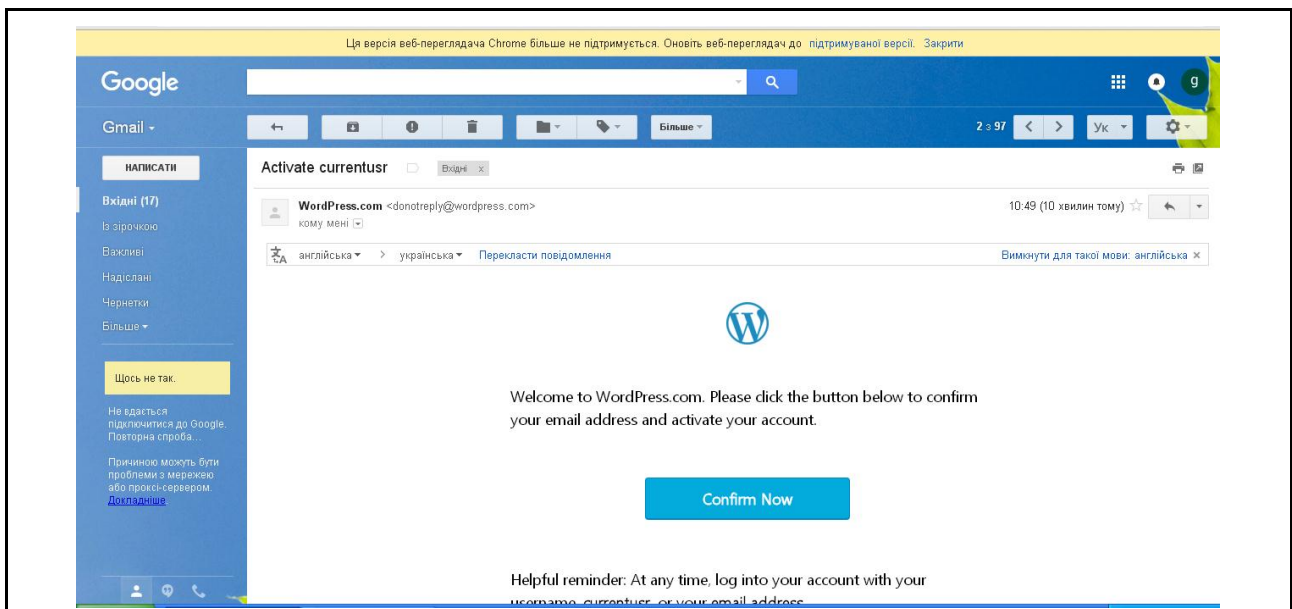


Рис. 7.18.5. Вікно Gmail та отриманий реєстраційний лист

Після активації побачимо сторінку, наведену на рис. 7.18.6.

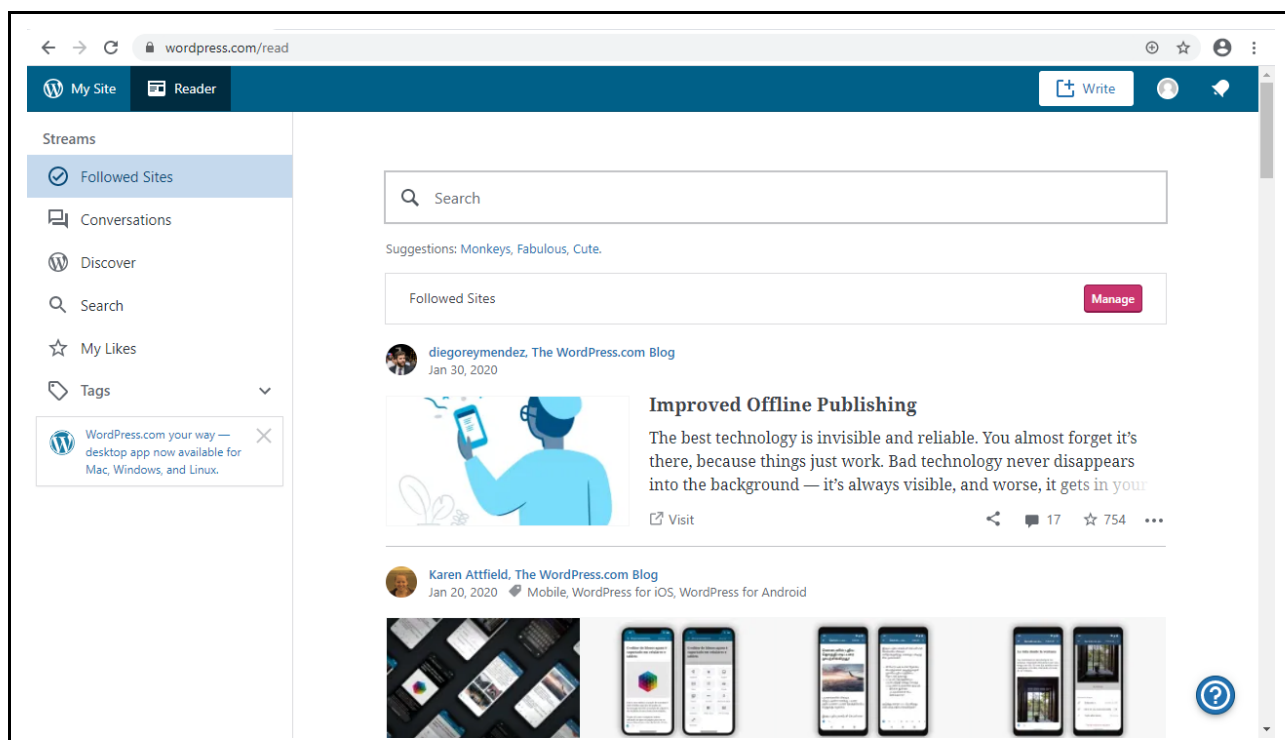



Рис. 7.18.6. Стартова сторінка зареєстрованого користувача

Розглянемо налаштування профілю користувача. Для цього необхідно виконати клік мишею на зображенні  у правій верхній частині екрана.

Автоматично завантажиться сторінка із загальним описом профілю користувача My Profile (рис. 7.18.7) та меню в лівій частині екрана. Звертаємо увагу на місце розташування кнопки Log out, яка дозволяє виконати вихід з акаунта.

На поточній сторінці My Profile (рис. 7.18.7) можна вказати чи змінити ім'я та прізвище, зображення користувача (аватар), публічне ім'я, яке буде відображатися на сайті та ввести короткий опис про автора. Також у нижній частині сторінки можна додати перелік web-сайтів, асоційованих із профілем.

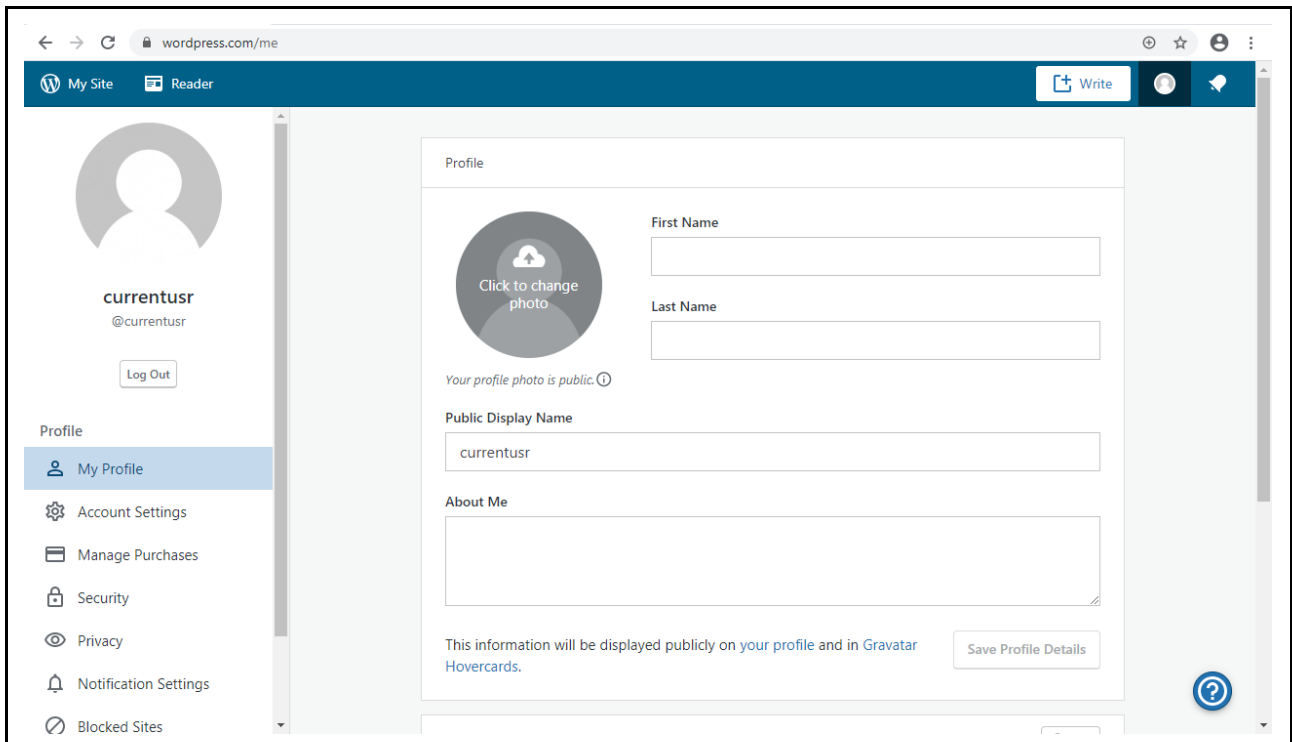


Рис. 7.18.7. Налаштування профілю користувача

Посилання Account settings у лівій частині містить такі параметри акаунта, як адреса електронної пошти, ім'я користувача, web-сторінка. Так само на цій сторінці можна змінити мову інтерфейсу в секції Interface Language. Виконайте клік на написі CHANGE (рис. 7.18.8), перейдіть на вкладку Eastern Europe та оберіть українську мову (рис. 7.18.9). Підтвердіть свій вибір натисканням Select Language та Save account settings у попередньому вікні в нижній частині (рис. 7.18.8). Після цього сторінка має автоматично перезавантажитися і тепер усі елементи інтерфейсу будуть підписані українською мовою.

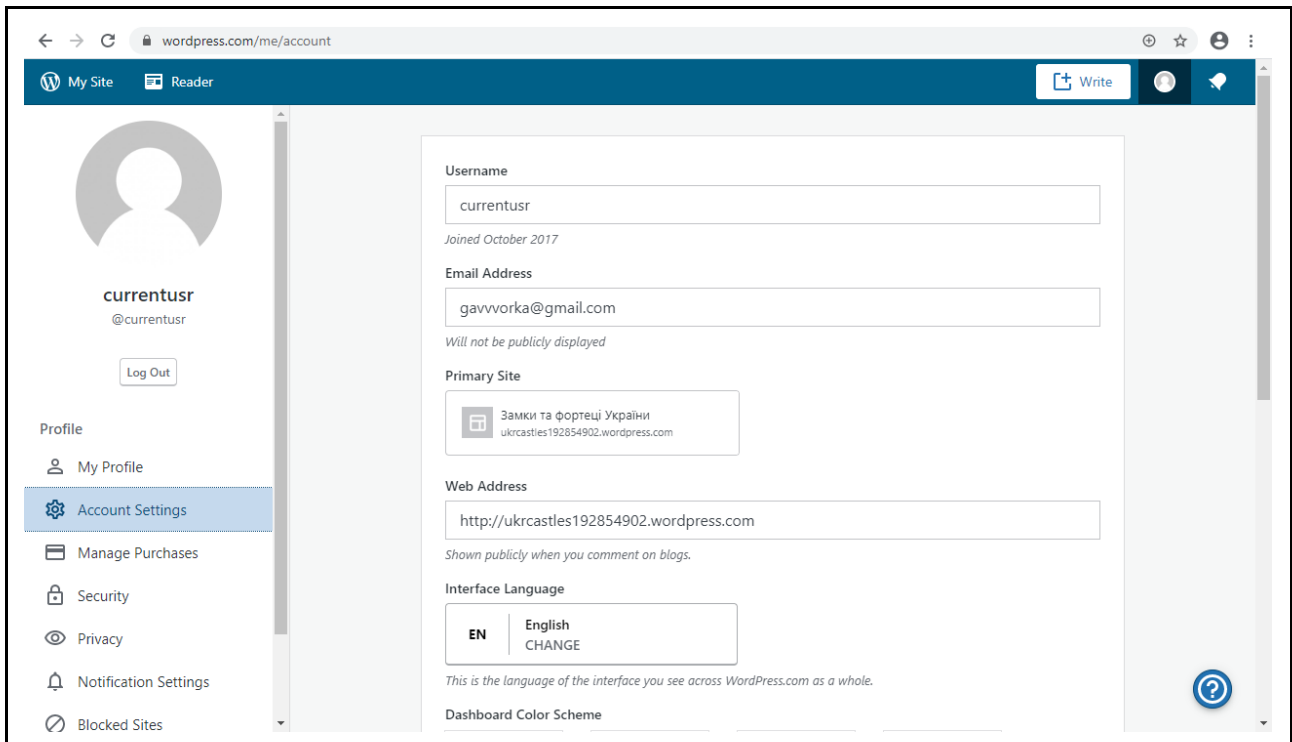


Рис. 7.18.8. Вікно налаштування мови сайту

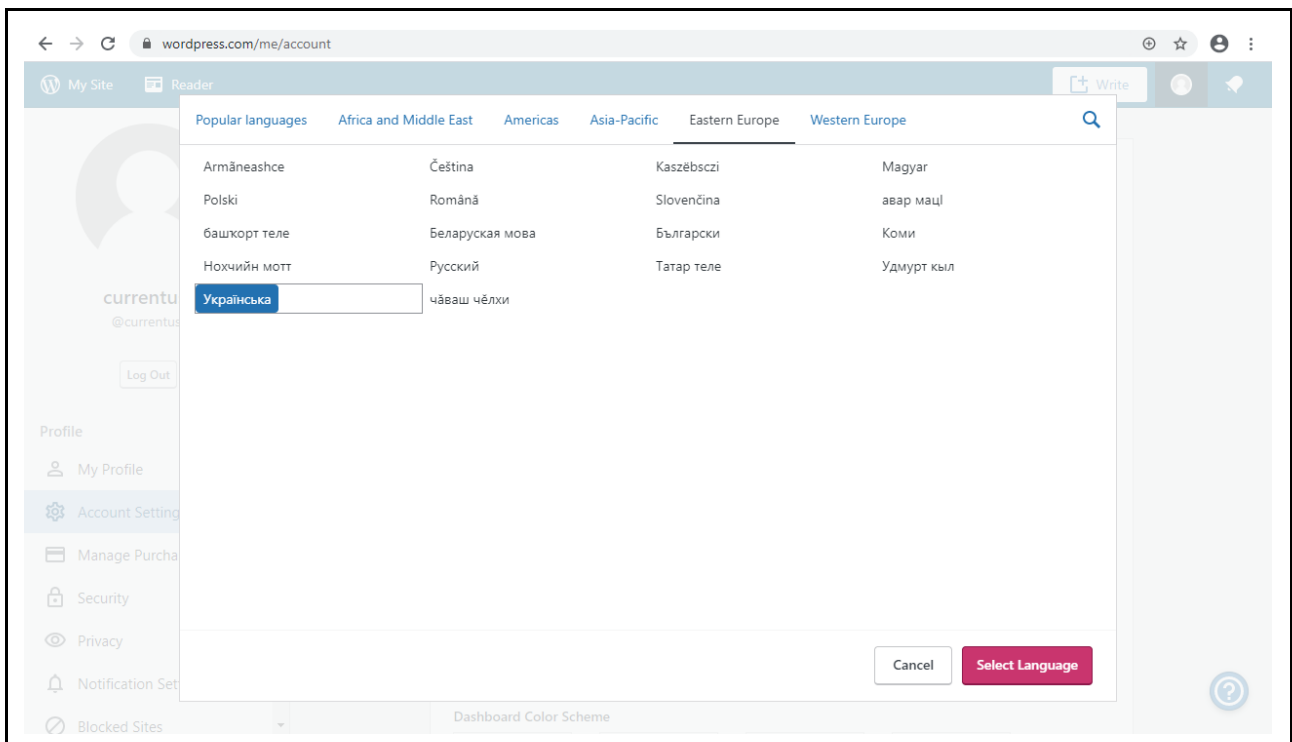


Рис. 7.18.9. Вікно із переліком доступних мов перекладу сайту

Наступний пункт меню ліворуч, який ми розглядатимемо, – "Безпека" (рис. 7.18.10). На цій сторінці можна змінити пароль, налаштувати перехід на сторінку із соціальних мереж, закладка "Двокрокова автентифікація"

дозволяє приєднати мобільний телефонний номер до акаунта і підтверджувати вхід через нього додатково із використанням логіна та пароля, останній пункт "Відновлення облікового запису" дає можливість вказати запасну адресу електронної пошти та телефон для відновлення доступу до втраченого акаунта.

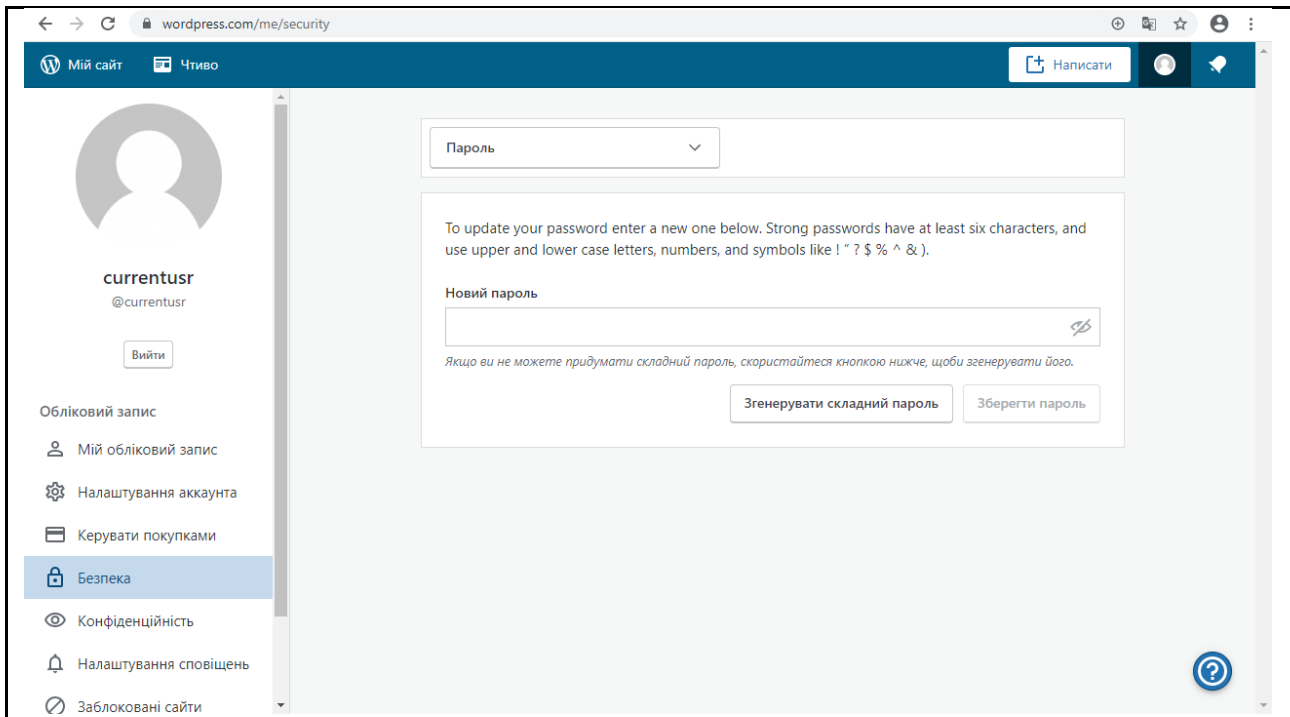


Рис. 7.18.10. Вікно налаштування параметрів безпеки

Між визначеннями "двокрокова автентифікація" та "двофакторна автентифікація" є різниця. Двофакторна автентифікація використовує будь-які 2 різних фактори автентифікації з трьох можливих: те, що вам відомо (наприклад, пароль); те, що ви маєте (наприклад, спеціальний пристрій для генерації синхронних паролів із сервером); те, ким ви є (біометричні способи, наприклад, відбиток пальця). Двокрокова автентифікація переважно складається з використання єдиного фактора "те, що ви маєте" двічі, а не двох різних факторів.

Пункт меню "Конфіденційність" надає можливість налаштувати передачу даних до команди WordPress, які пов'язані із досвідом використання вами послуг та функцій сайту і покращення його роботи.

Останній пункт меню ліворуч – "Налаштування сповіщень" (рис. 7.18.11). Тут можна налаштувати, в яких ситуаціях потрібно відправляти повідомлення про події на власному сайті чи на сайті WordPress.com на електронну пошту чи на сайт.

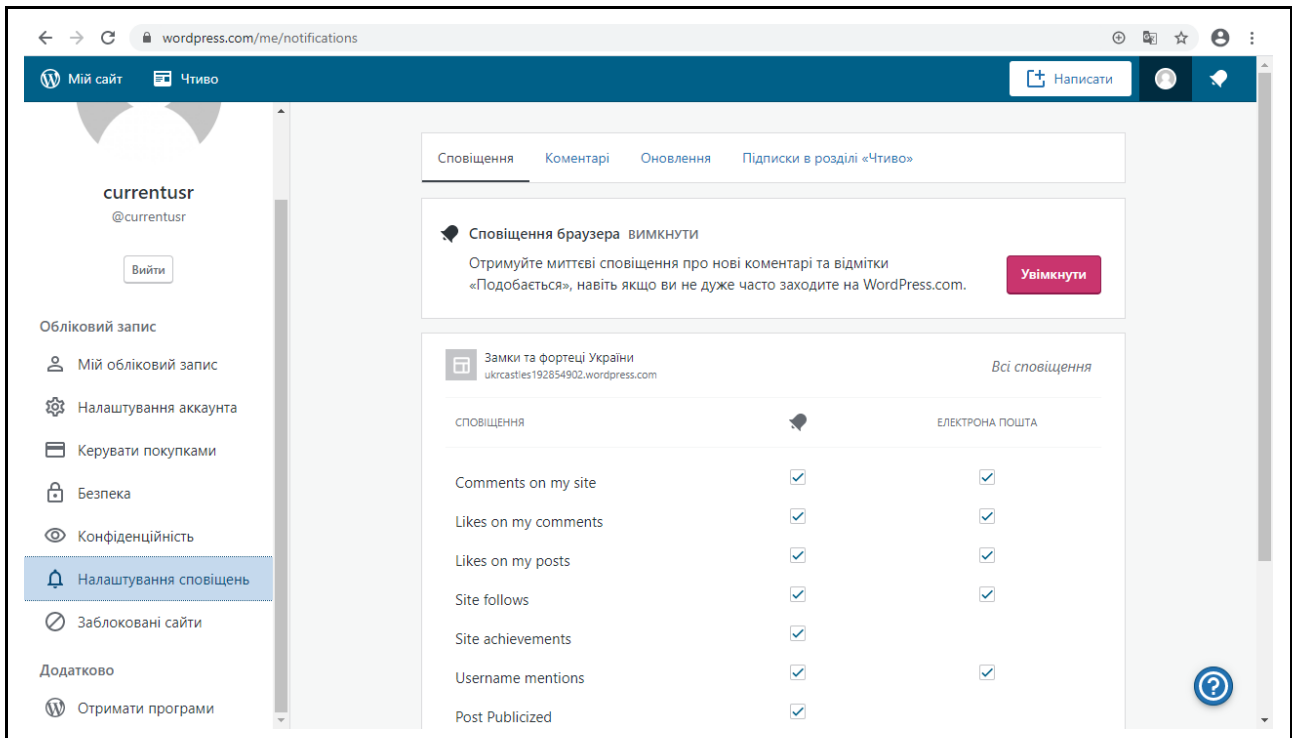


Рис. 7.18.11. Вікно налаштувань сповіщень сайту

Виконаємо клік на посилання "Чтиво" в лівій верхній частині екрана (рис. 7.18.12). На цій сторінці представлено повідомлення WordPress, які можуть бути цікавими для вас. Кнопка "Керувати" в центральній частині сторінки дозволяє побачити сайти, на повідомлення яких ви вже підписані (рис. 7.18.13).

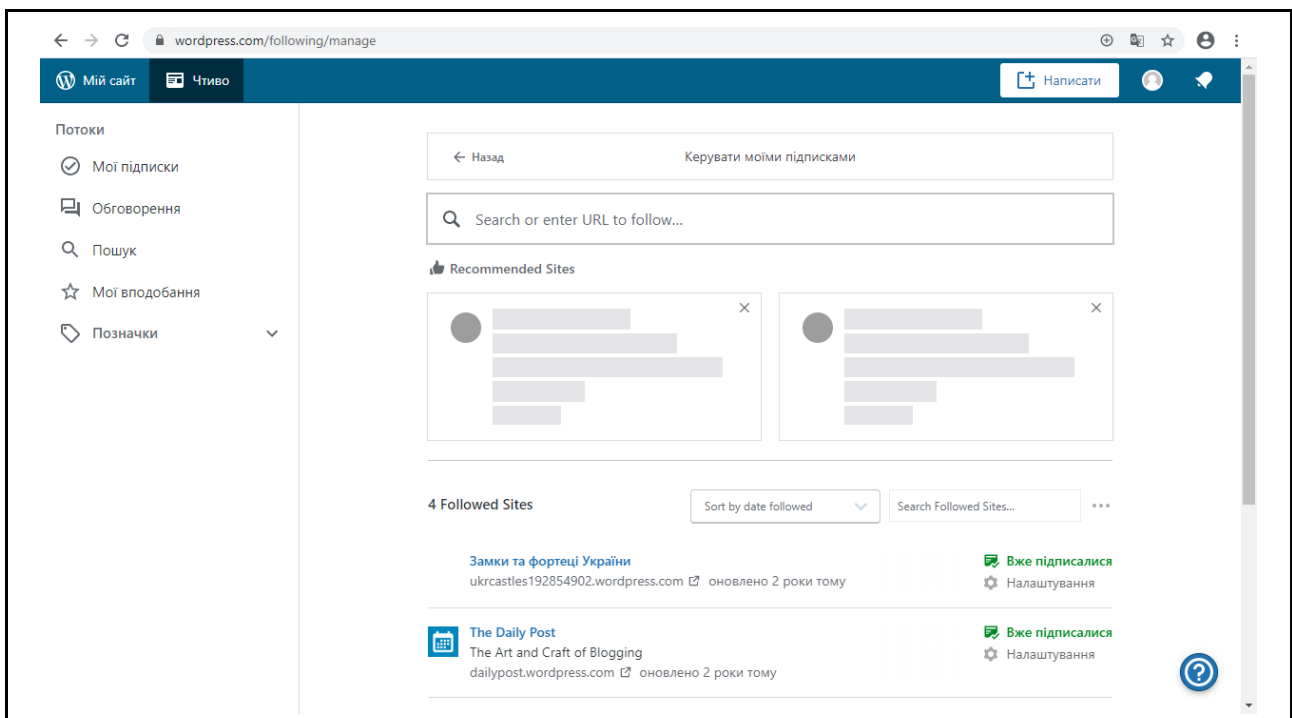


Рис. 7.18.12. Перелік сайтів, на які підписаний користувач

Біля кожного сайту, на який ви підписані, стоїть відповідна позначка "Вже підписалися", якщо клікнути на неї – від цього сайту можна відписатися. Трохи нижче розташовано посилання "Налаштування", яке дозволяє вказати, чи бажаєте ви отримувати повідомлення під час коментування та появи нових постів на цих сайтах.

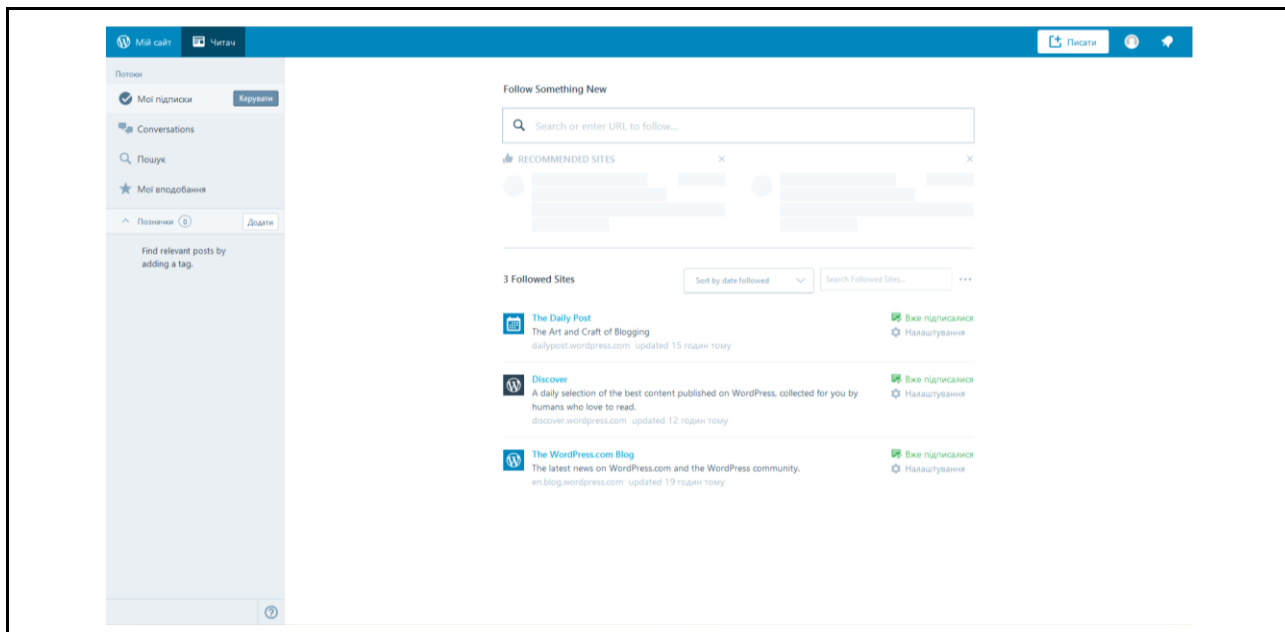


Рис. 7.18.13. Керування підписками на сайті

Створення нового сайту

Виконаємо авторизацію на сайті WordPress.com, використавши логін та пароль, створені в результаті попередньо, після чого має з'явитися сторінка за замовчуванням, зображена на рис. 7.18.14.

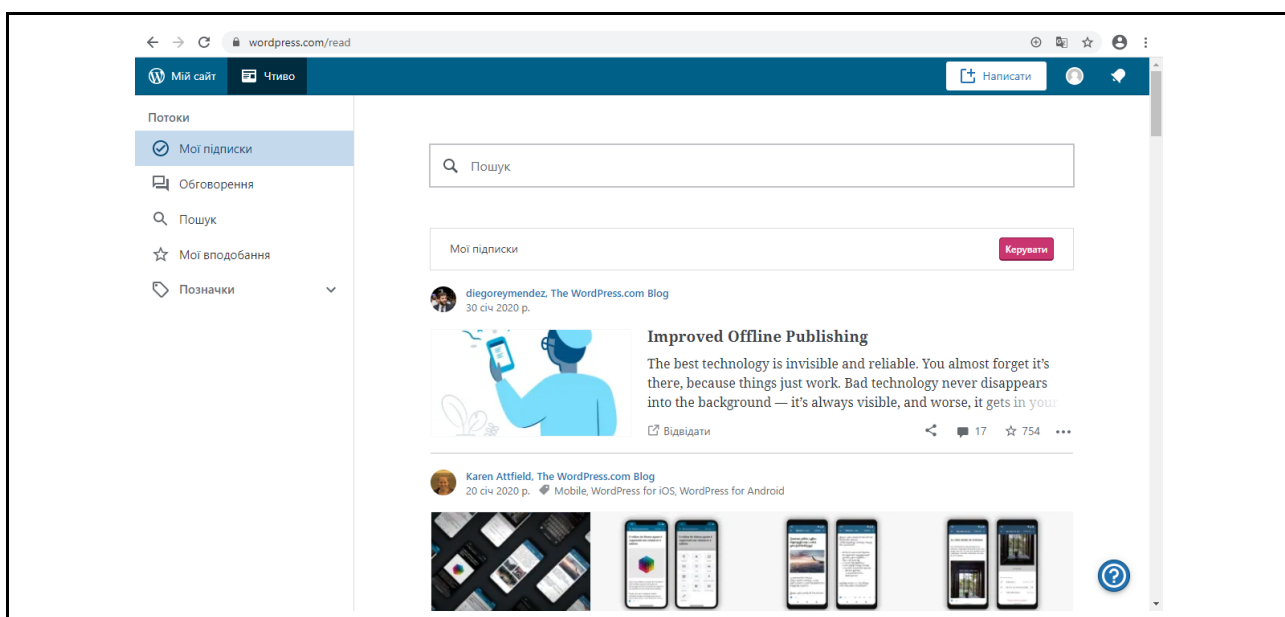


Рис. 7.18.14. Сторінка авторизованого користувача за замовчуванням

Перейдемо до створення власного сайту, для цього натиснемо на посилання "Мій сайт" у лівому верхньому куті сторінки. Якщо сайт уже було створено до цього, ви відразу побачите сторінку зі статистикою відвідування сайту, подібну до тієї, що показана на рис. 7.18.15. Посилання Add new site в нижній частині меню ліворуч дозволяє додати новий сайт.

У тому разі, якщо сайт ще не створено, ви побачите повідомлення про відсутність сайтів і кнопку "Створити сайт".

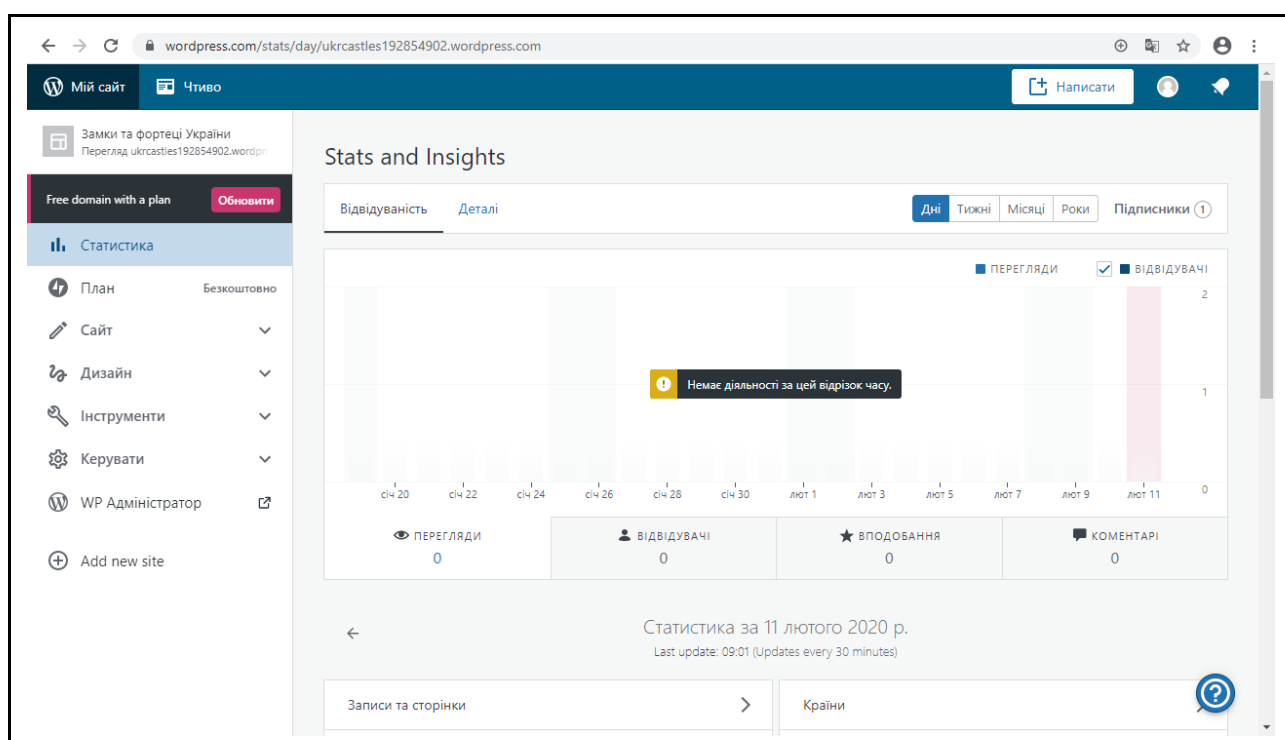


Рис. 7.18.15. Сторінка, що відображається, якщо сайт було створено

На вибір пропонується чотири типи сайтів та короткий опис для кожного (рис. 7.18.16):

- блог, призначений для розповсюдження ідей та історій серед тих людей, хто підписаний на вас;
- сайт, призначений для компаній чи людей, які рекламують власний бізнес чи товар;
- портфолію, яке призначене для візуалізації проєктів чи досягнень;
- магазин, призначений для продажу товарів чи послуг, фраза purchase required означає, що цей тип сайту доступний лише за платною підпискою.

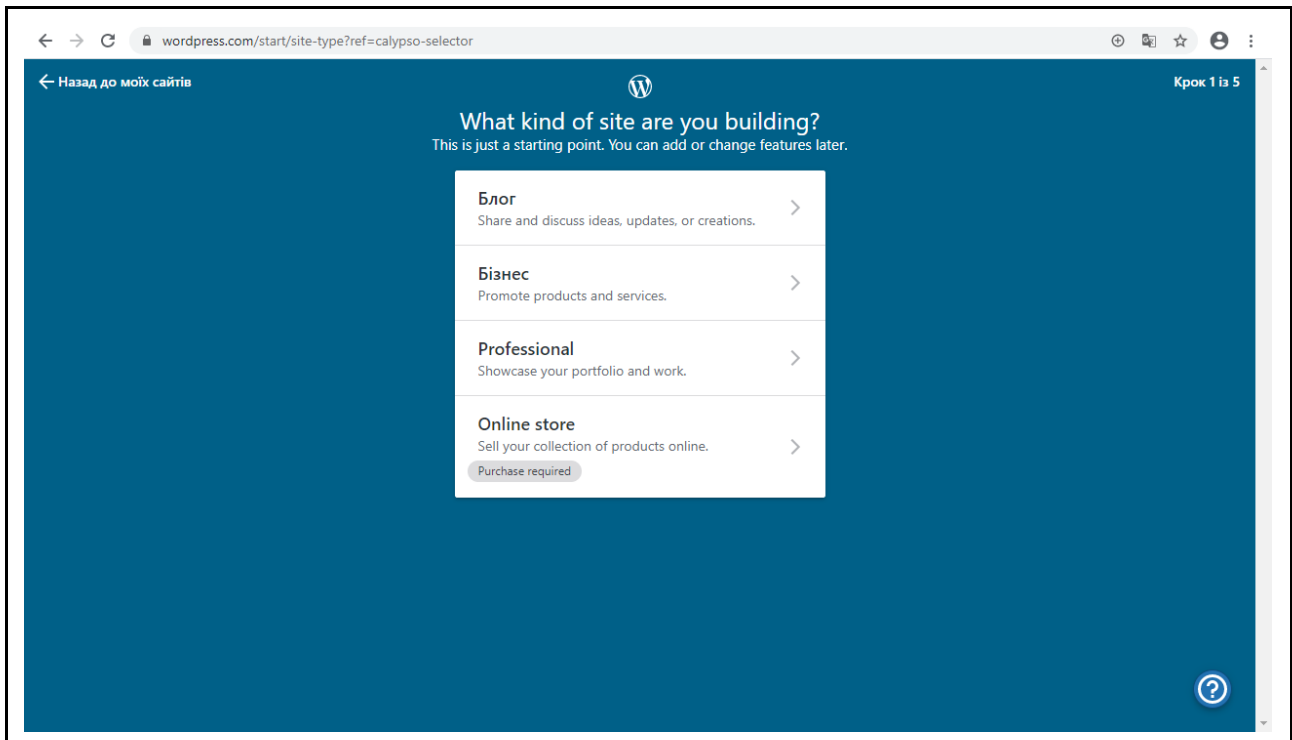


Рис. 7.18.16. Візуалізація сайтів різного типу

Створимо для прикладу сайт найпростішого типу – блог. На першому кроці налаштувань необхідно обрати тему сайту, після чого WordPress відразу запропонує шаблон за замовчуванням (рис. 7.18.17).

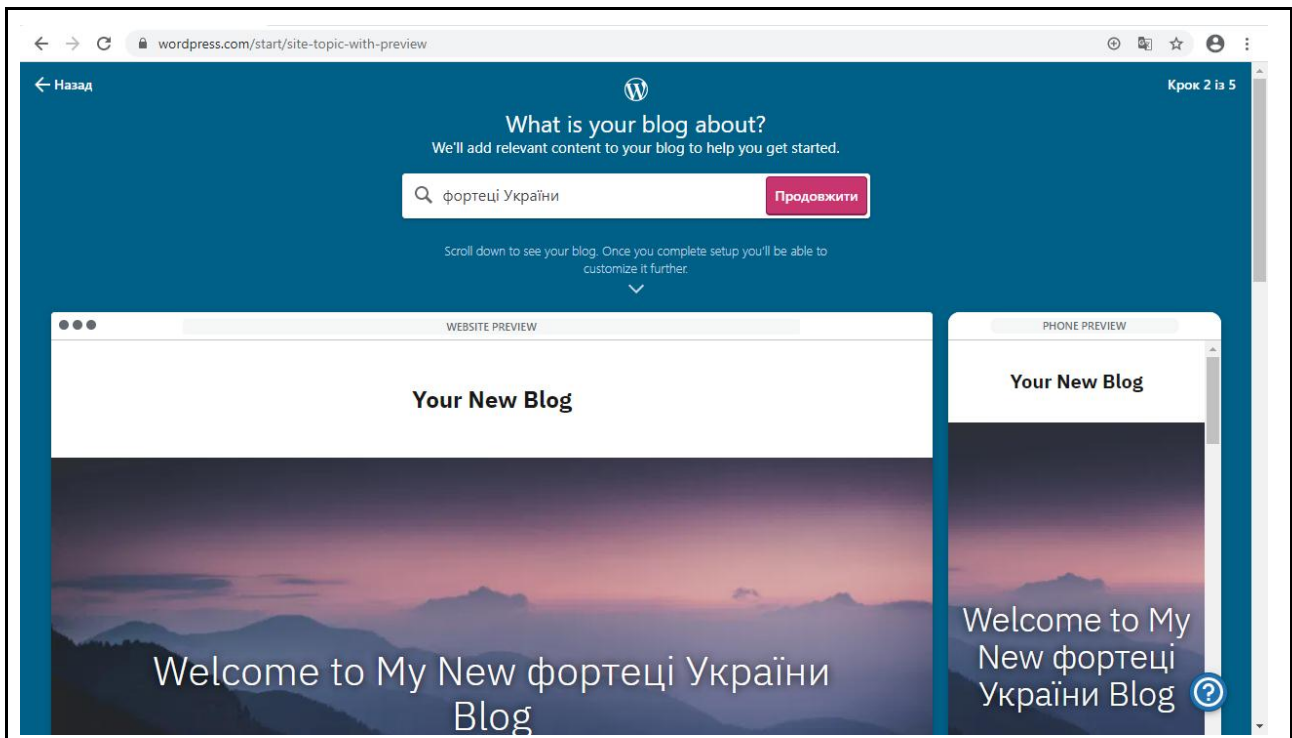


Рис. 7.18.17. Шаблон сайту за замовчуванням

На наступному кроці потрібно ввести чи обрати доменне ім'я (рис. 7.18.18). Наш демонстраційний сайт буде присвячено замкам та фортецям України, тож нехай доменне ім'я буде ukrcastles. Якщо ми введемо цю назву, побачимо варіанти, які пропонує WordPress.com (див. рис. 7.18.18). Наприклад, перше доменне ім'я ukrcastles.travel.blog може бути обраним безкоштовно. Водночас, більш вигідні доменні імена, такі як, наприклад, ukrcastles.com, ukrcastles.blog, ukrcastles.org, ukr-castles.com є доступними лише через Premium-підписку. Оберемо перший, безкоштовний, варіант, натиснувши біля нього посилання "Обрати".

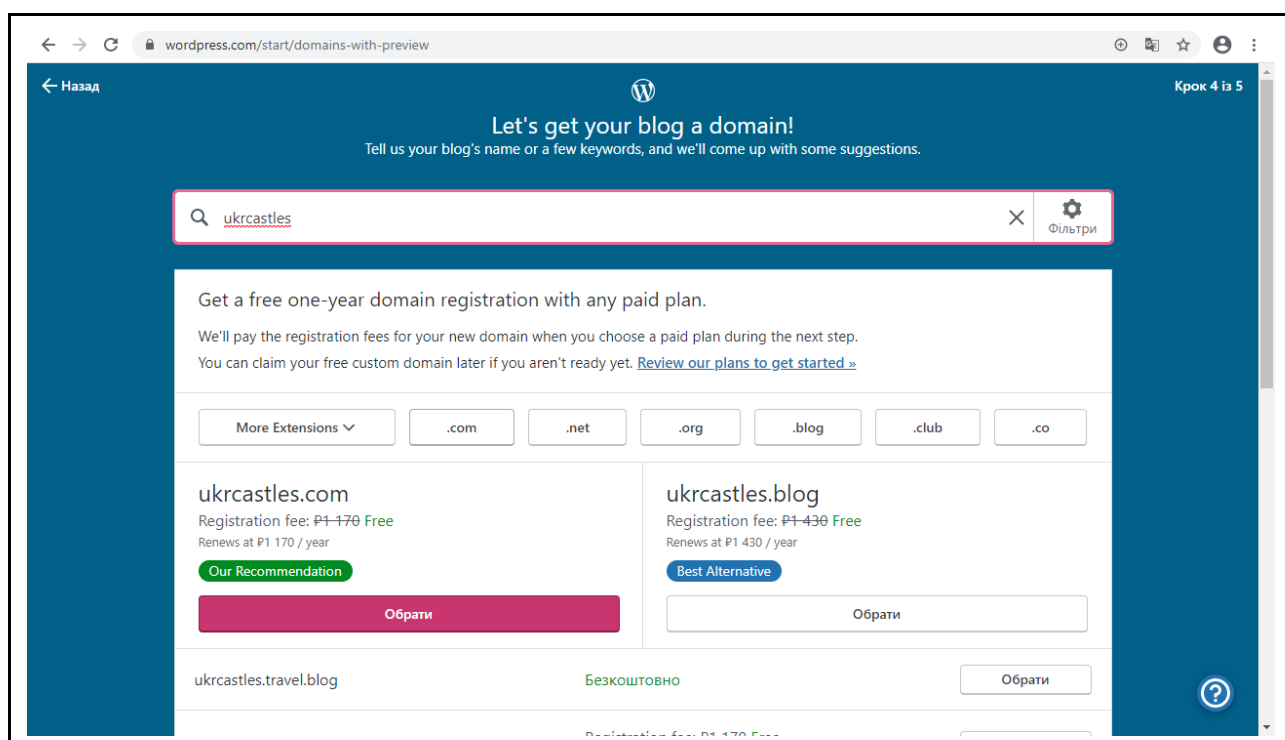


Рис. 7.18.18. Варіанти доступних доменних імен

На наступній сторінці у верхній частині потрібно обрати безкоштовний тип плану. Цей крок є останнім, наступною сторінкою відображається меню для редагування сайту та вітання (рис. 7.18.19).

Звертаємо увагу, що адреса новоствореного сайту розпочинається з протоколу https, що є вимогою для створення захищених сайтів із шифруванням даних.

Позначка "Приватно" у меню ліворуч біля назви сайту повідомляє, що сайт зараз доступний у приватному режимі. Праворуч є меню Launch my site, яке дозволяє це змінити. Натисніть цю кнопку і виконайте послідовність із трьох кроків, які є подібними до обрання доменного імені.

Після цього позначка "Приватно" має зникнути. Зверніть увагу, що необов'язково робити сайт публічним відразу, це налаштування може бути змінено в будь-який момент через пункт меню "Керувати" / "Налаштування".

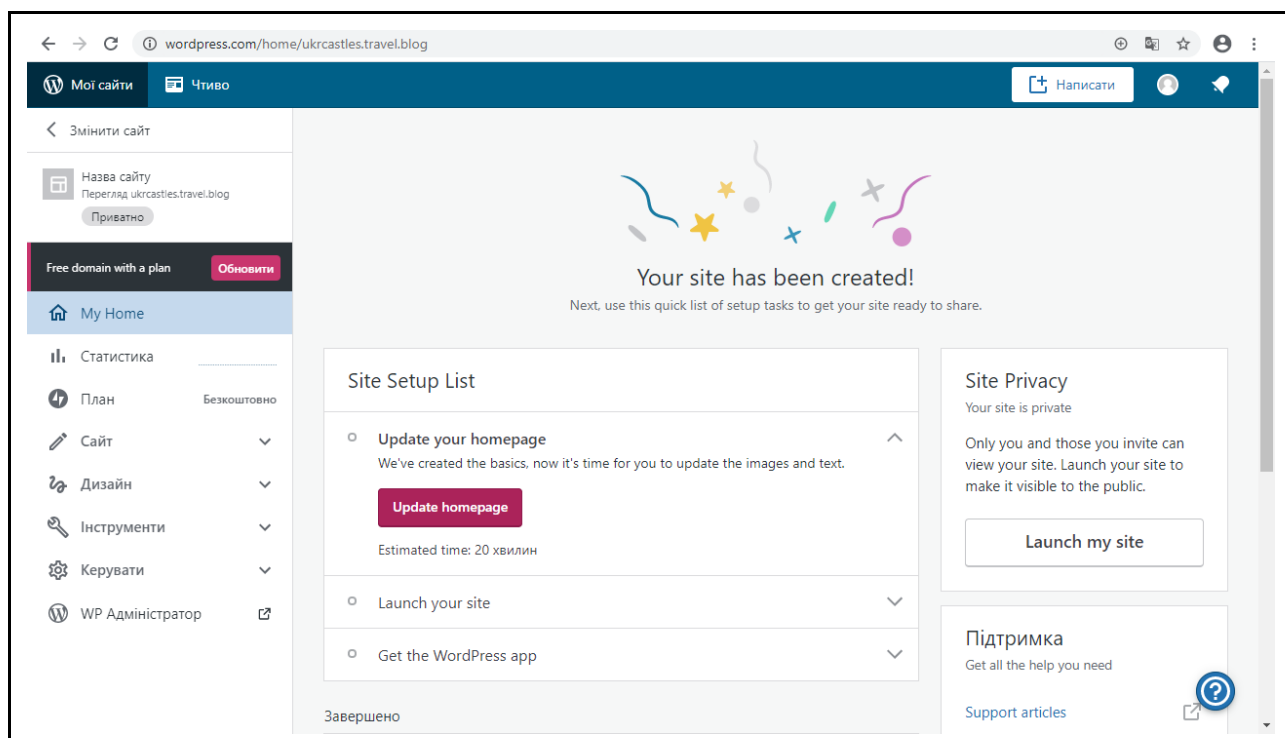


Рис. 7.18.19. Сторінка, що повідомляє про успішне створення сайту

Натиснемо на кнопку із назвою сайту у верхній частині меню ліворуч, щоб переглянути сайт. У результаті побачимо сторінку, наведену на рис. 7.18.20. У лівій верхній частині екрана є список, що випадає, в якому можна побачити відображення сайту з різних пристроїв, наприклад, планшетів чи телефонів. Кнопка "Відвідати сайт" відкриває сайт для перегляду незалежно від майстерні WordPress.

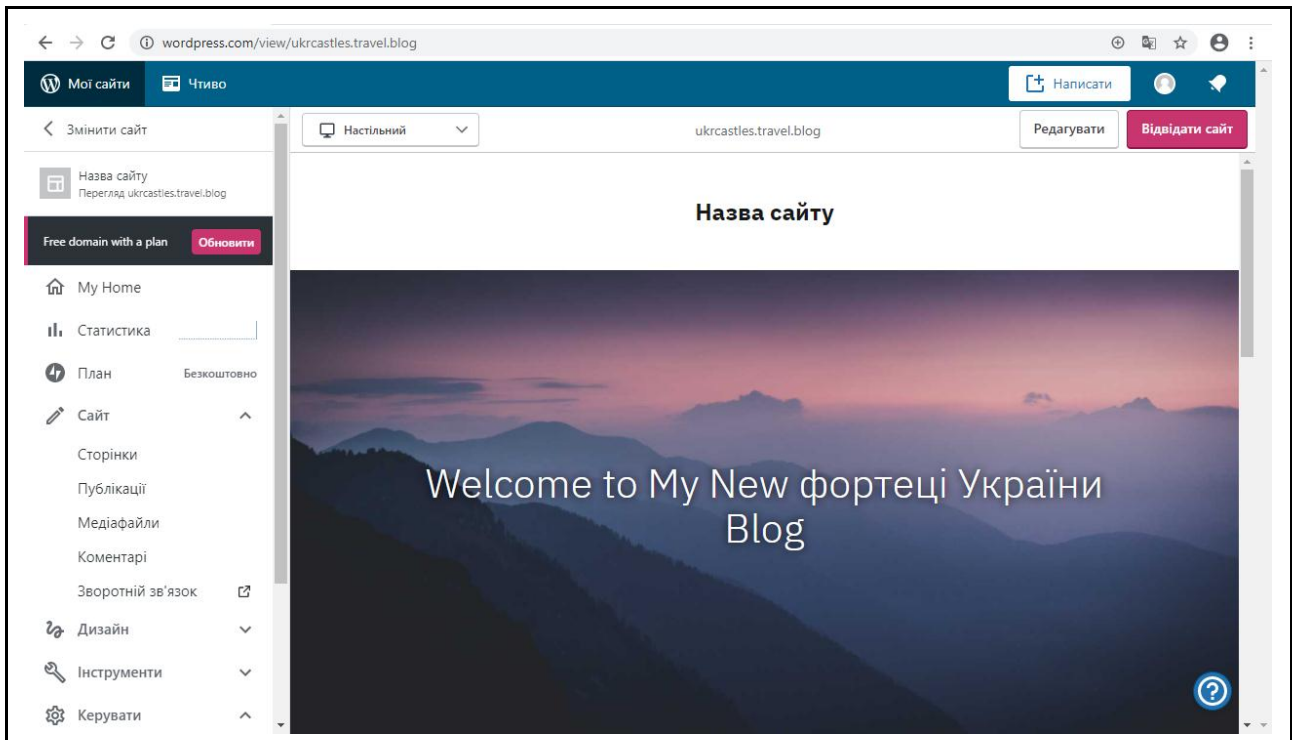


Рис. 7.18.20. Головна сторінка із налаштуваннями сайту

Давайте змінимо деякі основні налаштування сайту. Для цього перейдемо за посиланням "Керувати" в лівому стовпчику стрічки та оберемо "Налаштування". Сторінку, що завантажиться, можна побачити на рис. 7.18.21.

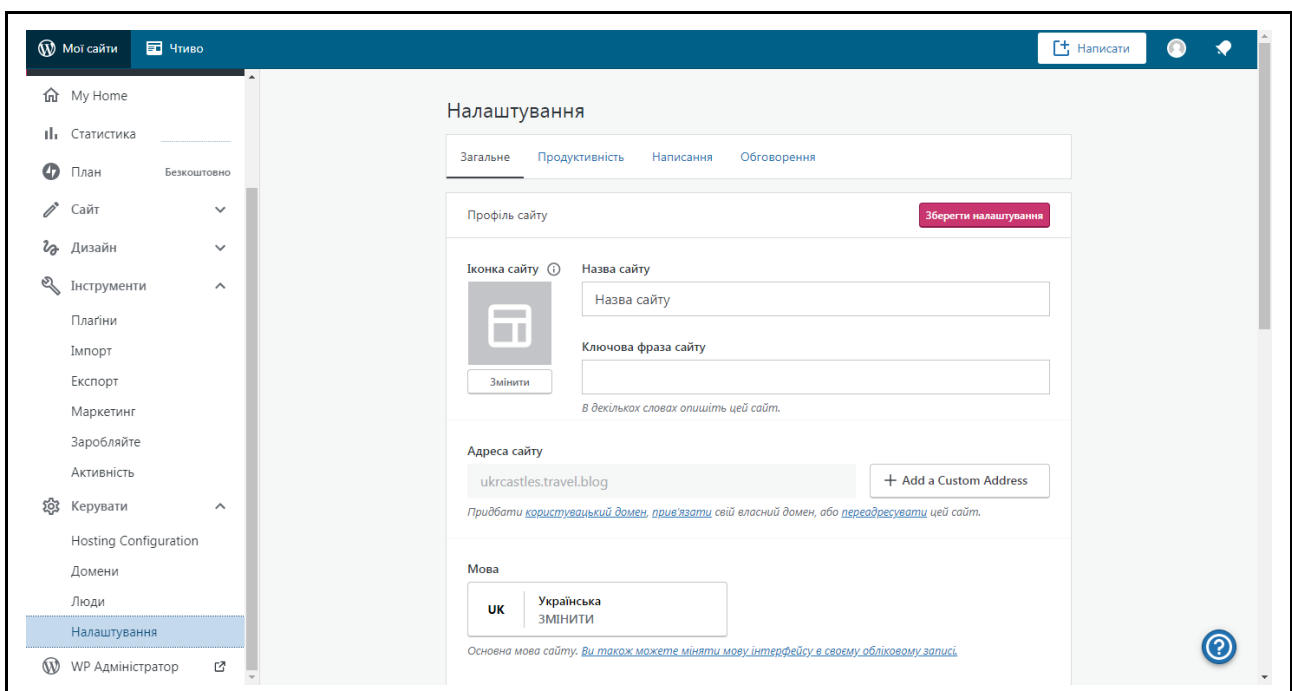


Рис. 7.18.21. Сторінка загальних налаштувань сайту

Сторінка містить 4 вкладки: "Загальне", "Продуктивність", "Написання" та "Обговорення". Розглянемо загальні опції.

У полі "Назва сайту" введемо коректне значення, наприклад, "Замки та фортеці України". Також тут можна змінити мову інтерфейсу WordPress, часовий пояс сайту, встановити, чи є сайт зараз публічно доступним, чи може він бути проіндексований пошуковими системами (наприклад, з'явитися в пошукових результатах Google). Також на цій вкладці містяться важливі елементи обслуговування сайту: очищення чи видалення сайту. Після внесення змін для їх збереження необхідно натиснути кнопку "Зберегти налаштування" у верхній частині екрана.

Тепер ми знову можемо переглянути сайт та переконатися, що назва головної сторінки змінилася.

Вкладка "Продуктивність" дозволяє встановити опцію більш швидкого завантаження сайту для компактних пристроїв на кшталт смартфонів чи планшетів, а також містить посилання на відповідну сторінку із налаштуваннями інтерфейсу для мобільних пристроїв.

Вкладка "Написання" містить параметри створення контенту: тут можна створити нову категорію та переглянути ті, що вже існують. Напис Show publish confirmation підказує, де можна знайти опцію, яка запобігає випадковій публікації статті (її можна знайти під час створення нової статті, клацнувши у правому верхньому куті на позначку з додатковими налаштуваннями із трьома крапками, знайти там пункт "Параметри" та переконатися, що перемикач Pre-publish checks увімкнений). На цій сторінці також можна змінити формат відображення дати, вказати кількість постів, рекомендацій та портфолію, що будуть відображатися, а також опції прокручування.

Вкладка "Обговорення" дозволяє встановити опції контролю за поведінкою користувачів: дозволяти коментувати статті, вимагати заповнення e-mail користувача, який залишає коментар, заборонити коментарі незареєстрованим користувачам, отримувати нотифікаційні листи (односторонні інформування про якісь зміни на сайті, наприклад, надходження нових коментарів чи публікації нових дописів), коли хтось додає коментар, увімкнути опцію модерації коментарів (необхідна для убезпечення від спам-повідомлень), налаштувати "чорний" список слів і таке інше.

Зміна контенту

Розглянемо можливість змін безпосередньо контенту сайту, а саме дописів у блозі. Наразі сайт містить лише єдиний запис за замовчуванням, тож давайте його змінимо. Для цього ми перейдемо в секцію "Сайт" у меню ліворуч та натиснемо посилання "Публікації" (рис. 7.18.22). Якщо виконати клік на позначці трьох крапок правіше від зображення блогу, можна побачити меню із доступними діями із цим записом, наприклад, редагування, перегляд запису чи статистики, видалення у кошику, переліку коментарів тощо.

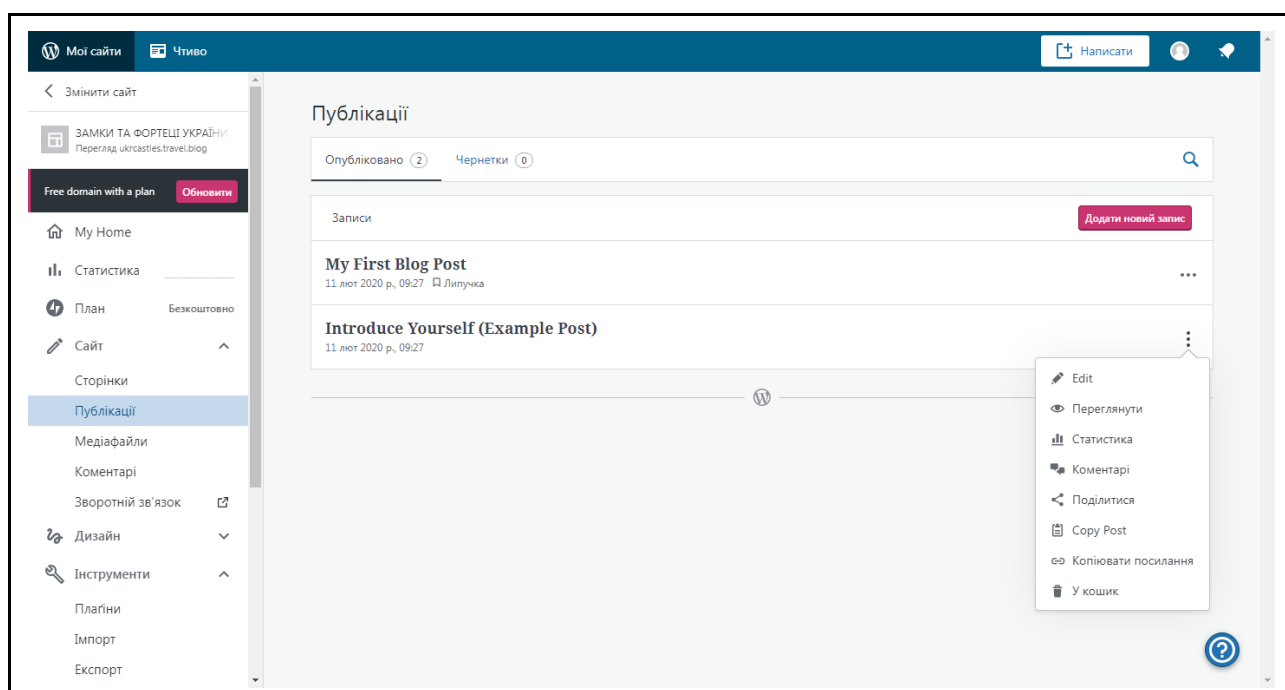



Рис. 7.18.22. Список записів блогу та операції із постом

Натиснемо на кнопку Edit (або просто клікнемо мишею) на перший запис блогу My First Blog Post та внесемо інформацію про першу фортецю – Хотинську.

Введення тексту відбувається блок за блоком. Позначка  дозволяє додати блок необхідного типу, позначка із трьома вертикальними крапками всередині блока дозволяє видалити його, перемістити, скопіювати, тощо.

Закладка "Блок" у меню праворуч відображає інформацію про активний у поточний момент блок та дозволяє налаштувати його параметри: кольори, буквицю, мобільний вигляд.

Закладка "Документ" у меню праворуч містить загальні параметри публікації, дату публікації, можливість відображати публікацію завжди у верхній частині сайту, розмітити публікацію в якусь категорію чи створити нову, дозволити коментарі, встановити головне зображення публікації.

Для того щоб змінити зображення, потрібно клікнути мишею на назві "Встановити зображення запису", на сторінці, що завантажилася, натиснути Add new, завантажити нове зображення та натиснути кнопку "Вставити". Це зображення можна в будь-який момент змінити або видалити.

Кнопка "Передперегляд" згори праворуч дозволяє попередньо переглянути сторінку перед публікацією.

Кнопка "Оновити" (або "Опублікувати" для нової публікації) завершує створення публікації.

На рис. 7.18.23 наведено оновлений варіант посту про Хотинську фортецю.

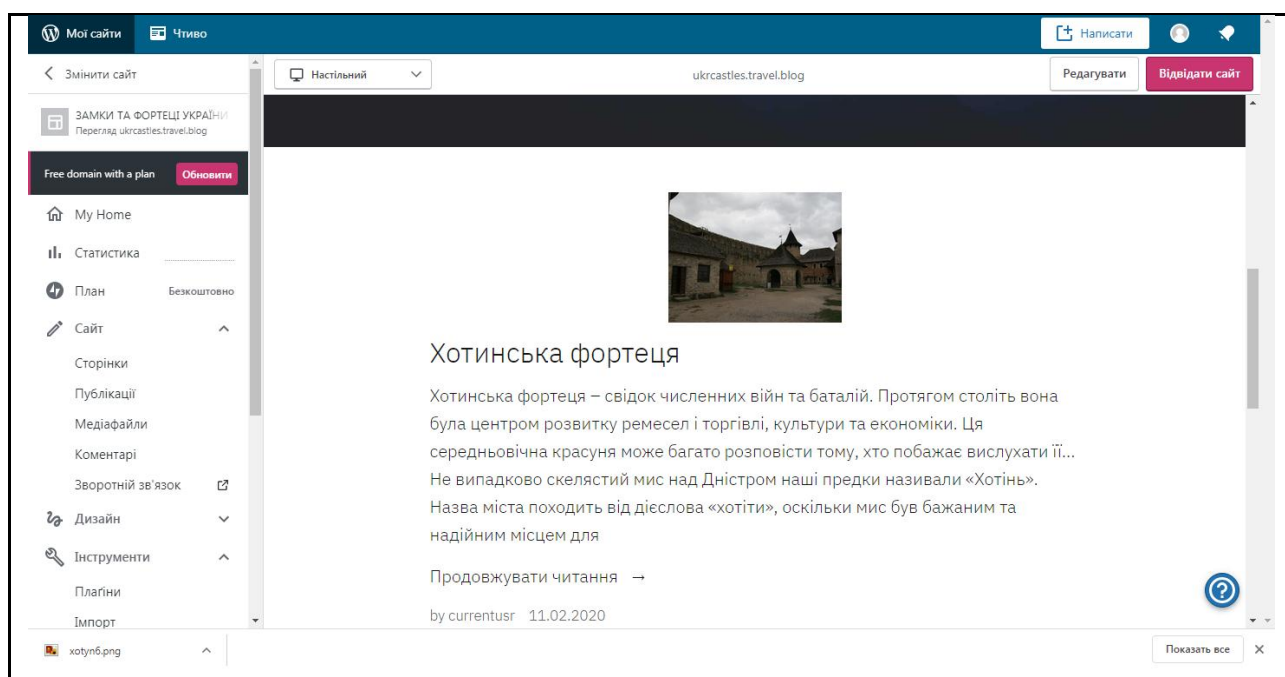


Рис. 7.18.23. Запис блогу про Хотинську фортецю

Створімо тепер новий пост про замок Любарта. Натиснемо кнопку "Додати новий запис" біля посилання "Записи" (див. рис. 7.18.22) та додамо текстовий опис. На рис. 7.18.24 зображено приблизний результат.

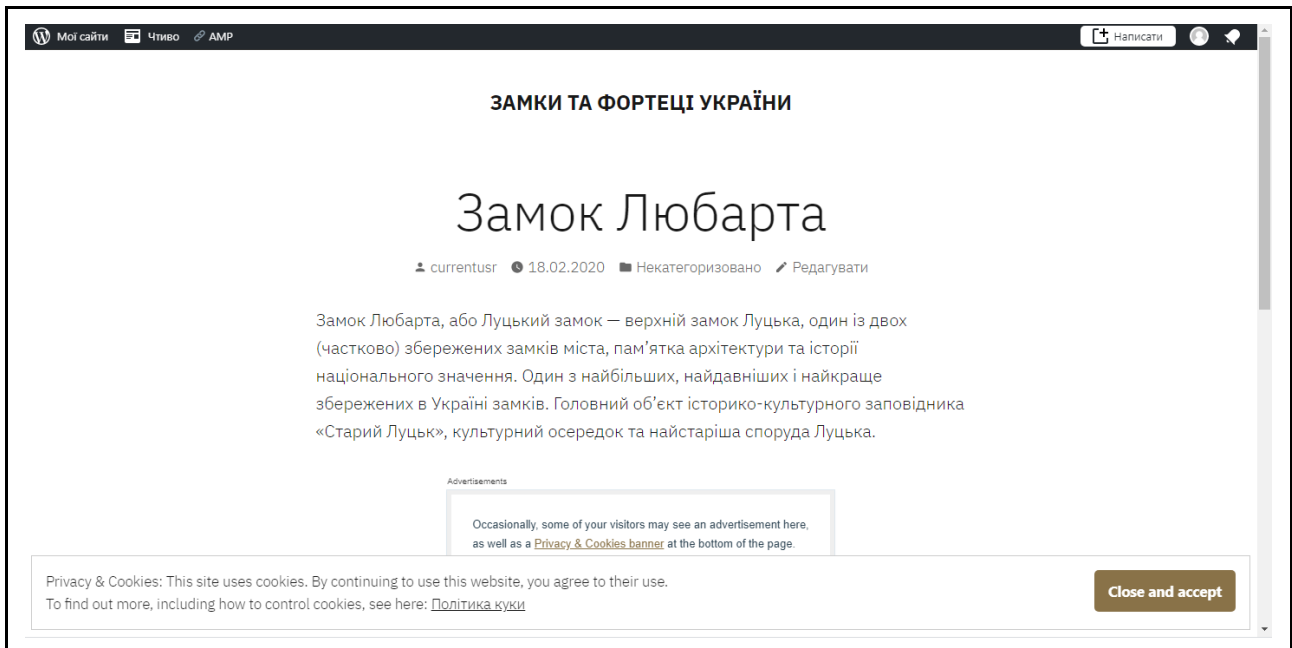


Рис. 7.18.24. Запис блогу про замок Любарта

Розглянемо інтерфейс додавання нового посту та створення нової категорії. Натисніть кнопку "Додати новий запис", що знаходиться в правій частині сторінки "Публікації" основного меню. Має відкритися нове вікно, зображене на рис. 7.18.25. Введемо назву "Кам'янець-Подільська фортеця" замість фрази "Додати заголовок" та вставимо підготовлений текст у блок нижче.

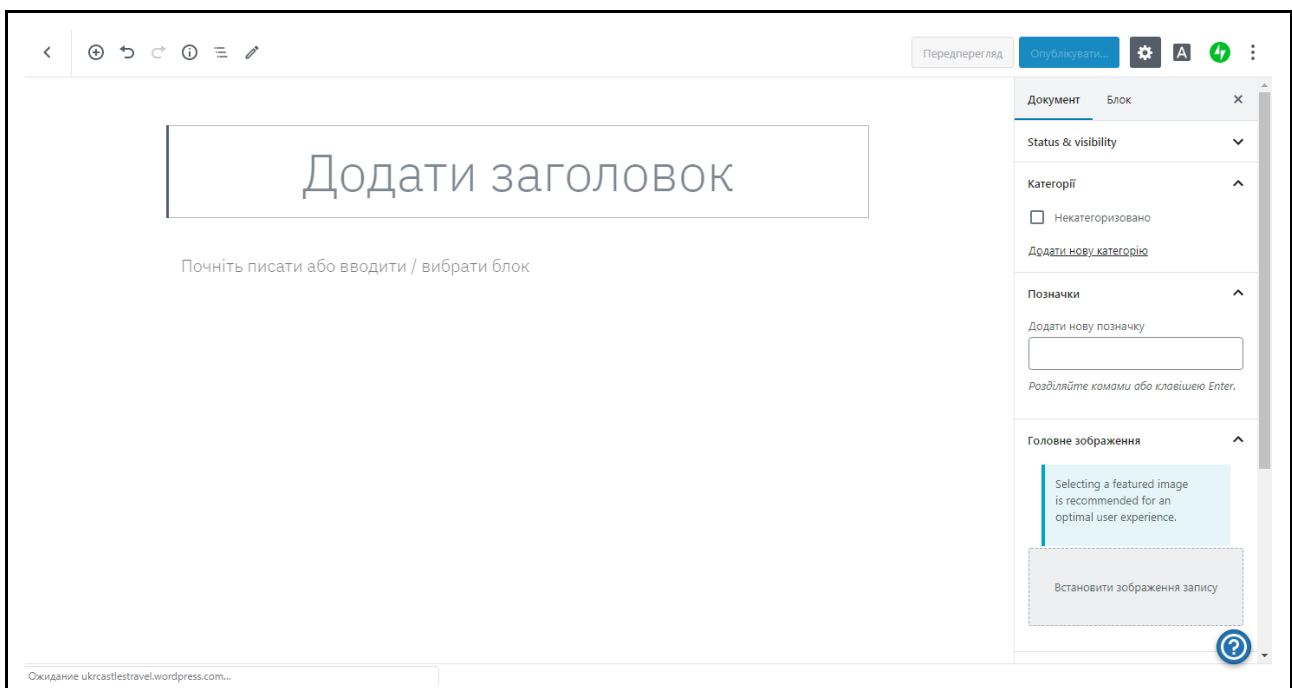


Рис. 7.18.25. Порожнє вікно для додавання нового запису

Для форматування цього тексту можна використати панель інструментів, що розташована над ним у лівому кутку (рис. 7.18.26), а також панель ліворуч для переміщення блока сторінкою та закладку "Блок" у меню праворуч.

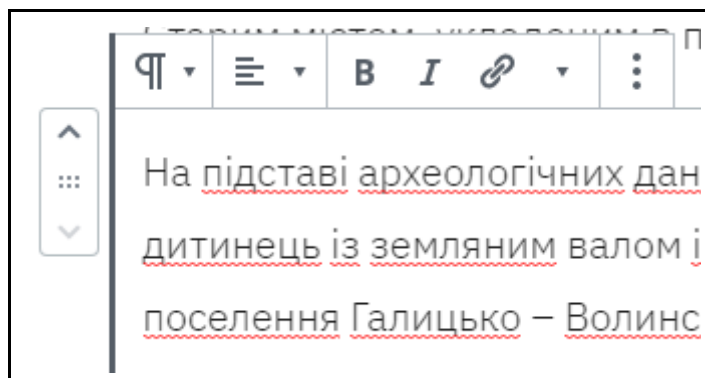




Рис. 7.18.26. Панель інструментів форматування блока

Ця панель містить кнопки для форматування тексту, розміру шрифту та його типу, встановлення цитат та гіперпосилань, позначення тексту кольором, вирівнювання, додавання зображення та гіперпосилань тощо.

Якщо натиснути спеціальну позначку у вигляді трьох крапок  та обрати "Редагувати як HTML", це дозволить змінювати вміст та зовнішній вигляд блока безпосередньо із використанням мови розмітки HTML.

Натискання піктограми  (вона з'являється в центральній частині екрана як реакція на переміщення покажчика миші за межами всіх блоків) дає можливість створити новий блок (рис. 7.18.27). Це можуть бути загальні блоки контенту (наприклад, параграф, зображення, заголовок, галерея, список, цитата, файл, відео), елементи форматування (наприклад, код, таблиця чи просто фрагмент коду HTML), елементи розмітки (наприклад, розрив сторінки, кнопки, згортання тексту "Читати далі"), віджети (календар, останні коментарі, хмара міток і т. п.), публікація медіа із соціальних мереж.

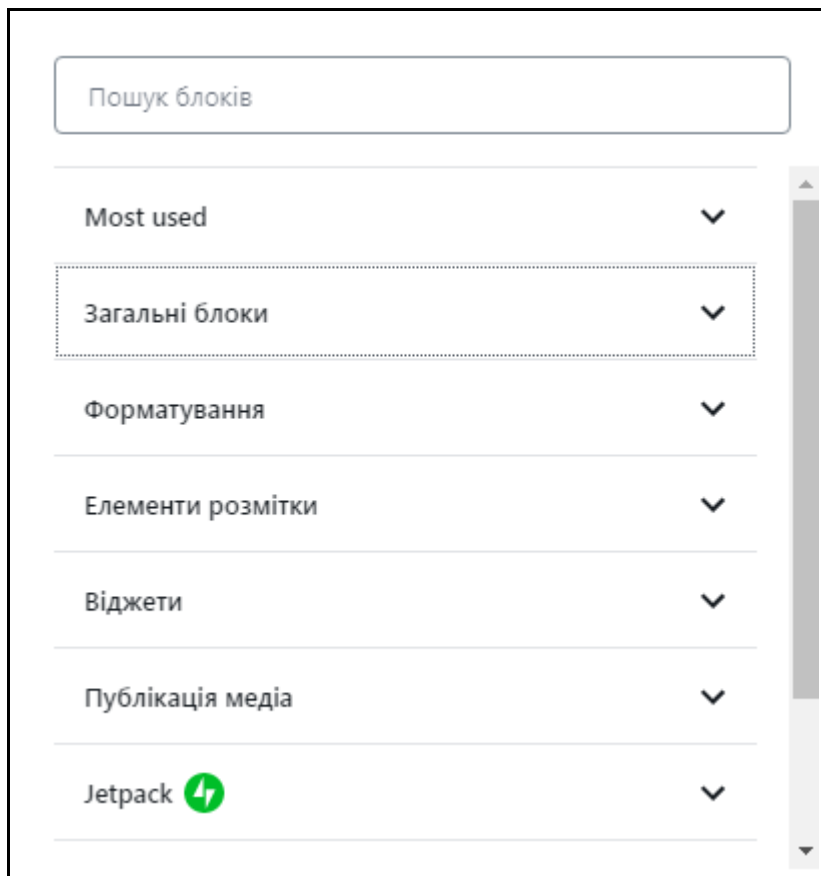


Рис. 7.18.27. Додавання нового блока

Повернемося до загальних налаштувань публікації праворуч та знайдемо там пункт "Категорії". Знімемо позначку "Некатегоризовано", натиснемо трохи нижче на посилання "Додати категорію", введемо її назву, наприклад, "Хмельницька область", оскільки ця фортеця знаходиться там. За необхідності можна обрати батьківську категорію. Підтверджуємо свій вибір натисканням "Додати нову категорію" та встановлюємо галочку напроти неї, відправивши таким чином нову публікацію в цю категорію. Публікуємо чи оновлюємо поточну публікацію відповідною кнопкою у верхній частині меню праворуч. Як можна побачити, назва категорії з'явилася біля назви публікації (рис. 7.18.28).

Кам'янець-Подільська фортеця

currentusr 21.02.2020 Хмельницька область Редагувати

Кам'янець-Подільська фортеця (Antimurale Christianitas – форпост Християнства) була споруджена для захисту Турецького мосту, що є сполучним перешийком між Старим містом, укладеним в петлю річки, і “великою землею”.

На підставі археологічних даних на території сучасної фортеці в IX – XII ст. існував дитинець із земляним валом і дерев'яною стіною, що захищає давньослов'янське поселення Галицько – Волинського князівства.

Рис. 7.18.28. Назва категорії біля публікації

Пункт меню "Медіафайли" в секції "Сайт" у лівій вертикальній стрічці меню містить інформацію про всі зображення, що використані на сайті (рис. 7.18.29).

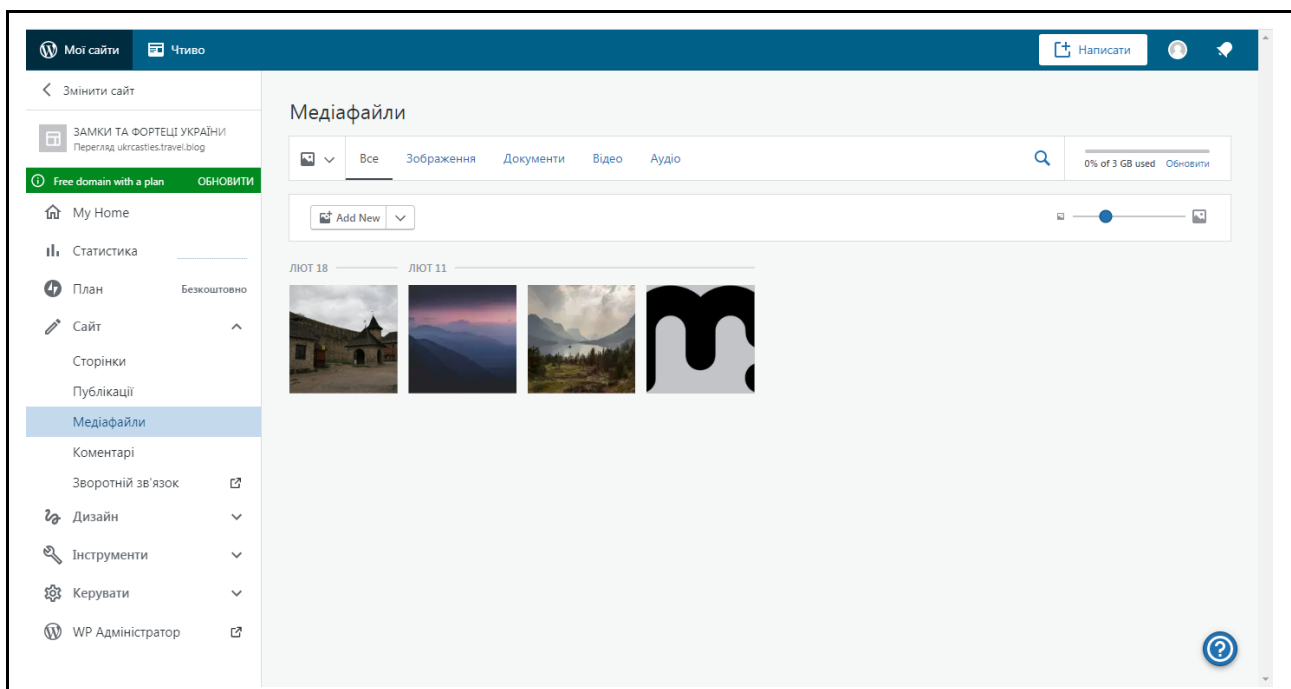


Рис. 7.18.29. Сторінка із зображеннями сайту

Розділ "Сторінки" містить інформацію про всі сторінки сайту, а не лише про пости блогу (рис. 7.18.30). Наприклад, тут можна налаштувати сторінки за замовчуванням Home та Blog Feed, створені автоматично.

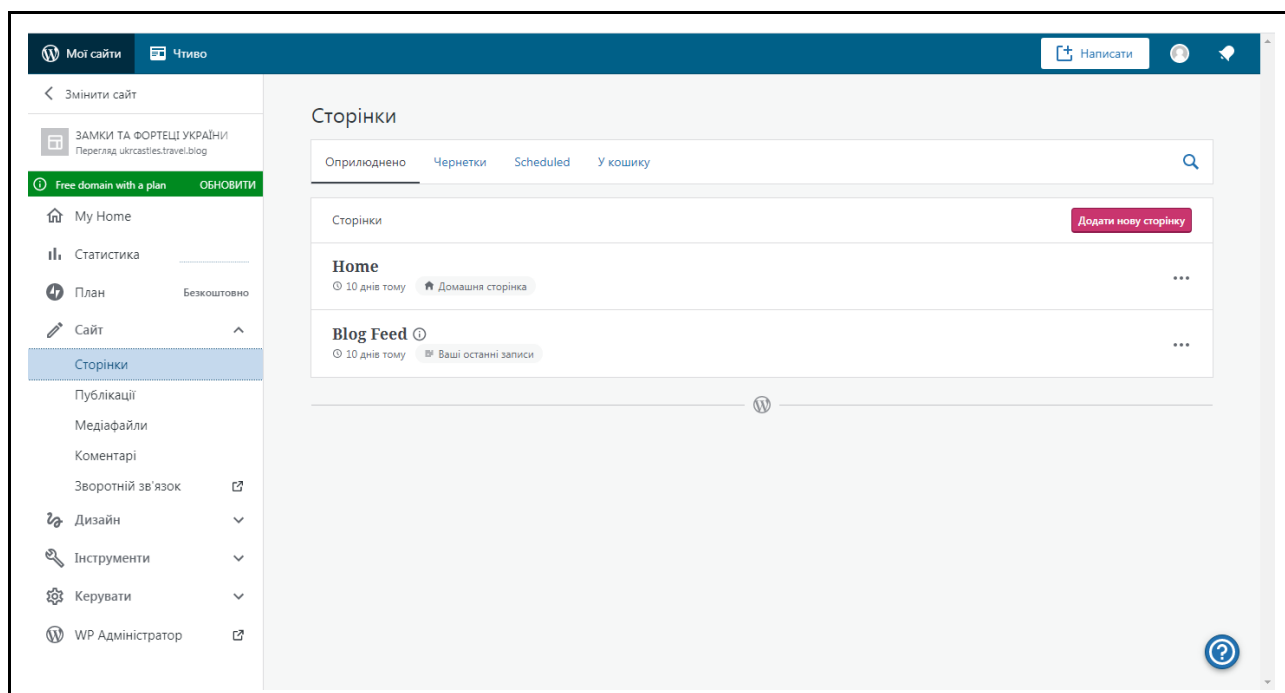


Рис. 7.18.30. Сторінка з усіма сторінками сайту

Розділ "Коментарі" містить перелік закладок для роботи з усіма коментарями, а також із коментарями, що очікують на підтвердження, видаленими, схваленими та коментарями, що вважаються спамом.

Зміна дизайну

Виконаємо вхід на сайт WordPress.com, використавши власний логін і пароль. Перейдемо послідовно за посиланнями "Мій сайт", "Дизайн" та "Теми", де побачимо доступний перелік тем (рис. 7.18.31).

У верхній частині сайту відображається поточна тема за замовчуванням ("Maywood") та посилання "Налаштувати" й "Інформація" біля неї. До них ми повернемося трохи пізніше.

Спробуємо обрати іншу тему, але спочатку переключимо режим відображення тем з "Всі" на "Безкоштовно", щоб побачити лише теми, які доступні для завантаження та встановлення без оплати (кнопка для фільтрації тем відображається під початковим списком тем на рис. 7.18.31).

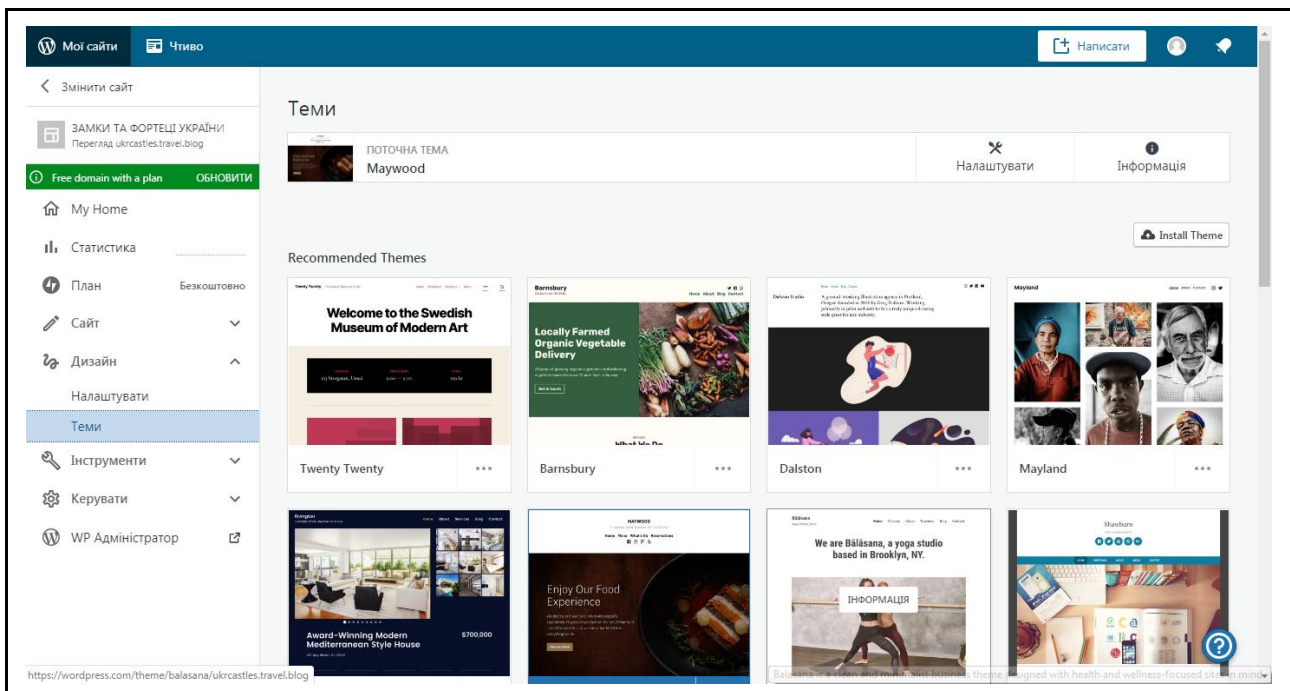


Рис. 7.18.31. Обрання дизайну сайту

Оберемо для прикладу тему Daga та виконаємо клік на ній. У результаті завантажиться сторінка, зображена на рис. 7.18.32. У лівій колонці на закладці Overview цієї сторінки наведено перелік особливостей обраної теми, а також корисний опис та інструкції для активації тієї чи іншої функціональної можливості теми. Рекомендуємо вам детально переглянути цей опис, щоб зрозуміти, які саме зміни в меню будуть внесені під час використання цієї теми. Наприклад, обрана тема Daga містить 2 доступні шаблони для сторінок сайту, які будуть додані в меню налаштувань (це меню доступне за посиланням "Сторінки сайту" для кожної зі сторінок сайту).

Закладка Support містить посилання на форум спільноти цієї теми, де можна задати запитання.

У правій колонці наведено графічне зображення теми.

Біля цих закладок міститься корисне посилання Open Live Demo, якщо перейти по ньому, можна побачити повноцінну робочу версію сайту із використанням цієї теми. Ця опція є дуже корисною, оскільки дозволяє побачити всі потенційні недоліки та переваги обраного дизайну.

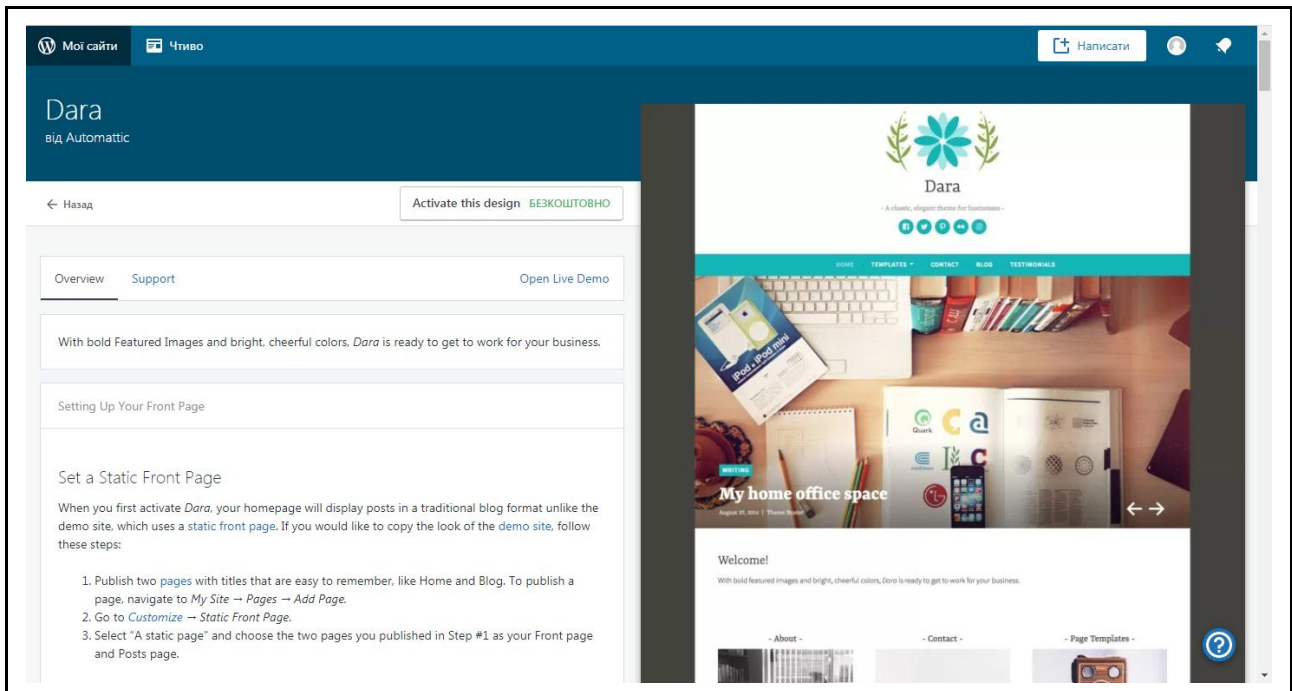


Рис. 7.18.32. Попередній огляд шаблону теми

Припустимо, що ця тема нас влаштовує, тож ми натиснемо на посилання **Activate this design БЕЗКОШТОВНО**, яке розташоване трохи вище знайомого нам уже посилання **Open Live Demo**. У результаті ми побачимо повідомлення, зображене на рис. 7.18.33. Натиснемо на **Learn about this theme**, потім натиснемо "Назад" до повернення на головну сторінку роботи із сайтом та перейдемо за посиланням "Перегляд" ліворуч угорі, щоб переконатися, що тема змінилася. Сайт із новою темою наведено на рис. 7.18.34.

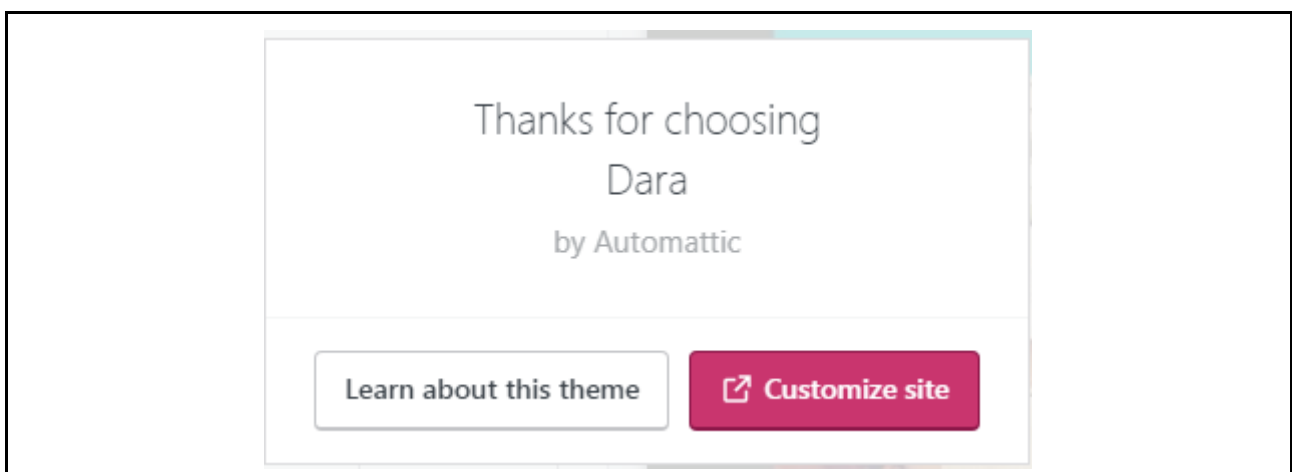


Рис. 7.18.33. Повідомлення після зміни теми

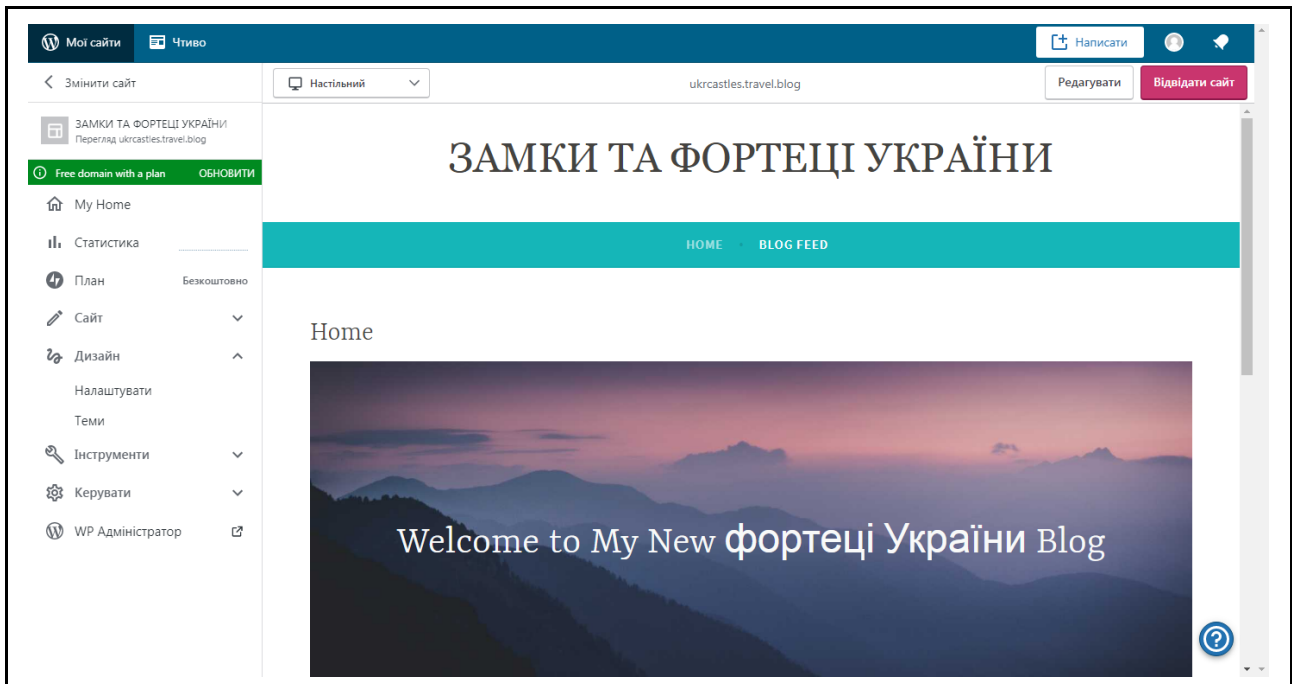


Рис. 7.18.34. Оновлений вигляд сайту після зміни теми

Повернемося до налаштування теми. Для цього необхідно натиснути "Налаштувати" в лівій частині меню в секції "Дизайн" (див. рис. 7.18.31). У результаті має завантажитися сторінка, наведена на рис. 7.18.35.

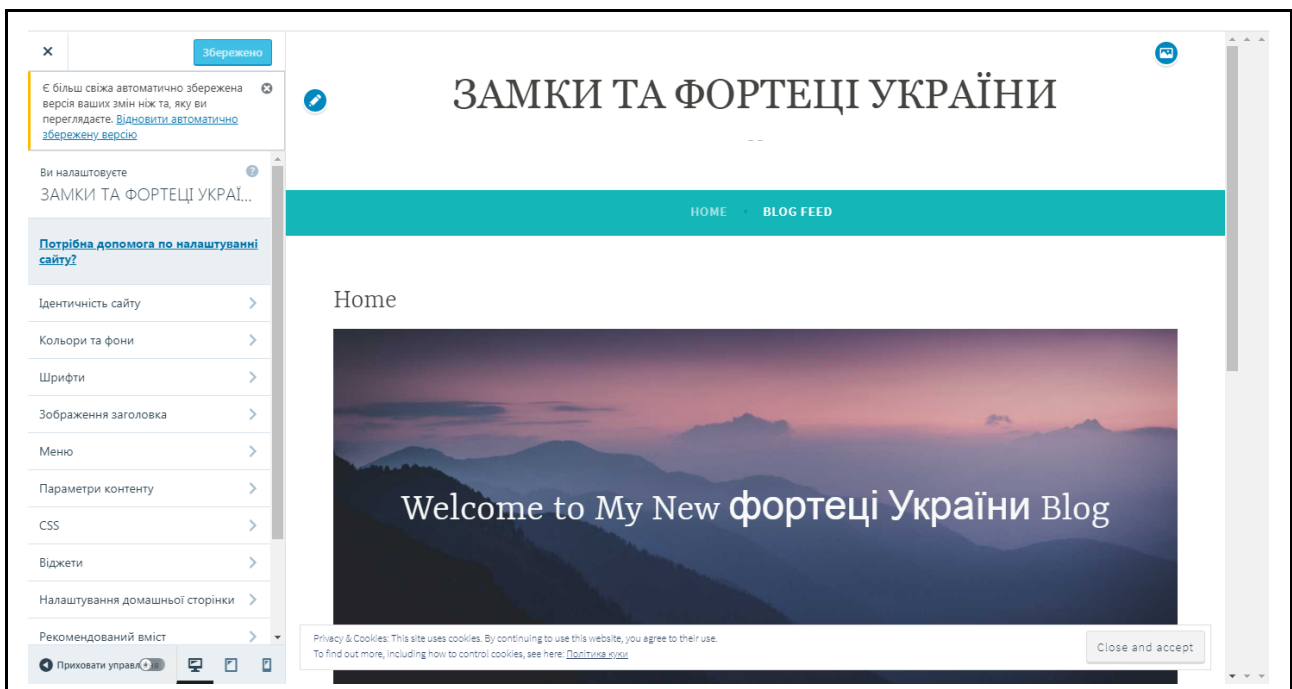


Рис. 7.18.35. Вікно налаштування теми сайту

Розглянемо ці налаштування більш детально. Їх можна змінювати двома засобами:

- за допомогою переліку всіх опцій, розташованого в лівій частині екрана у вигляді вертикального меню;
- за допомогою піктограм на елементах дизайну на сторінці праворуч, натискання на них призводить до автоматичного розкриття відповідного меню в лівій частині екрана.

У лівому нижньому куті наведено іконки для перевірки, як виглядає сайт у режимі планшета, стаціонарного комп'ютера (за замовчуванням) та телефону.

Доступні для налаштування далі наведені пункти меню (рис. 7.18.36 та 7.18.37).

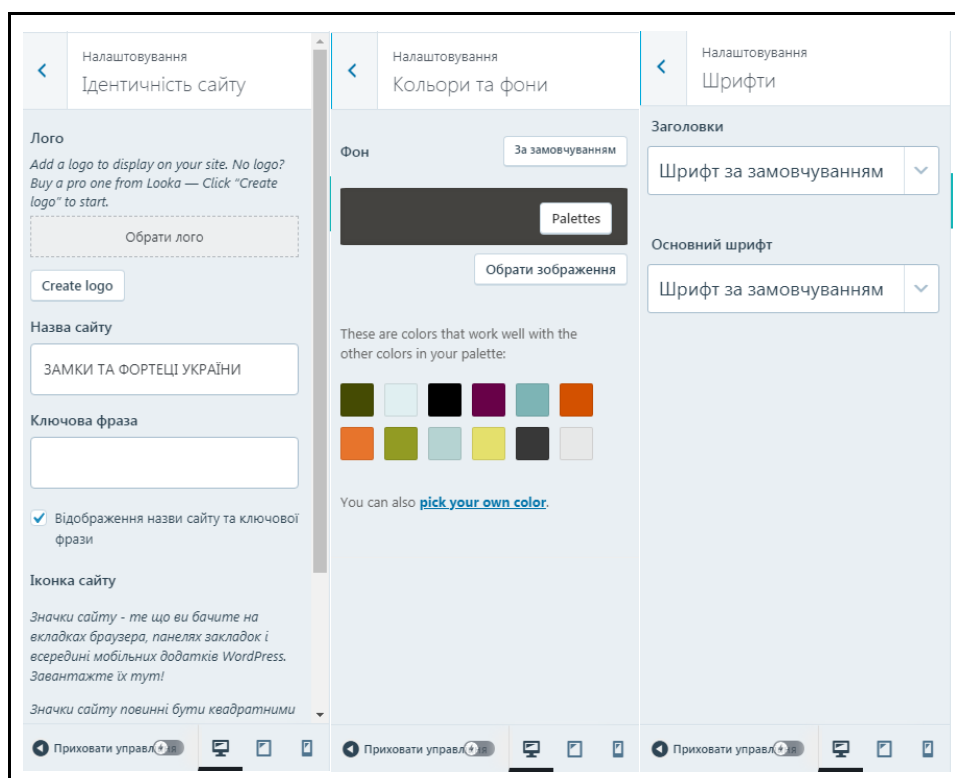
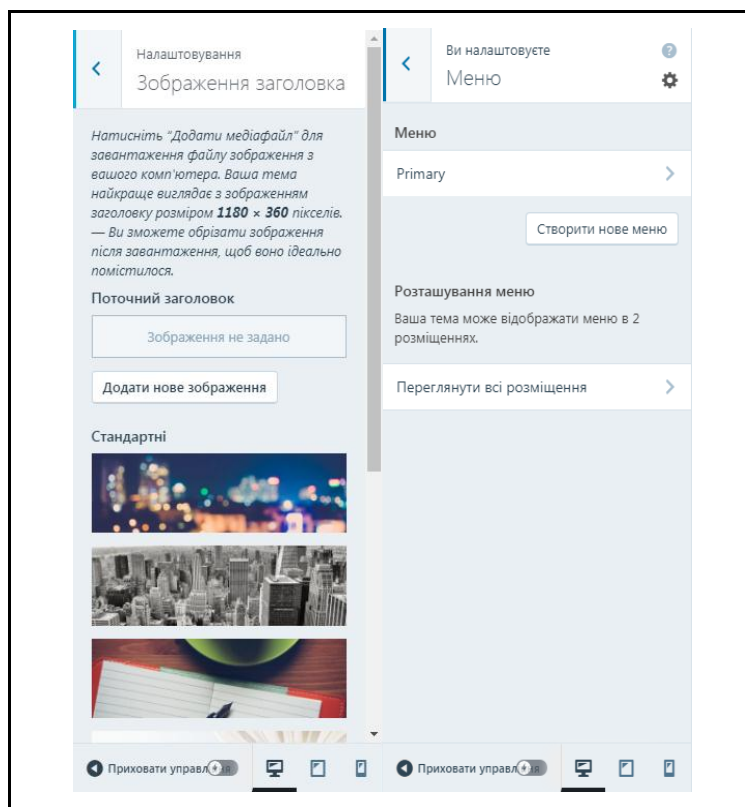


Рис. 7.18.36. Налаштування спеціальних параметрів теми



Закінчення рис. 7.18.36

Ідентичність сайту. Дозволяє встановити загальні опції для всього шаблону сайту: завантажити логотип, вказати назву сайту та ключову фразу (або приховати їх). Усі зміни доступні в інтерактивному режимі, тобто під час зміни значення в лівій частині екрана відображення сайту у правій частині екрана змінюється автоматично і негайно. Також у цій секції налаштувань можна встановити іконку для сайту (зазвичай її можна побачити у закладках браузера), а значення Footer Credit дозволяє обрати рекламне текстове повідомлення WordPress.org у нижній частині екрана, яке можна приховати лише після придбання іншого (не безкоштовного) плану.

Кольори та фони. У цьому меню можна змінити кольорову схему сайту. Якщо обрати будь-який колір за допомогою посилання Pick your own color, а потім клацнути трохи вище на кнопку Palettes, система автоматично запропонує перелік палітр на основі обраного кольору, які мають у своєму складі ефективну комбінацію кольорів.

Шрифти. Цей пункт меню дозволяє змінити шрифти, їхній тип (курсив, напівжирний) та розмір – для заголовків та для основного тексту. Пам'ятайте, що обрання вдалого та зручного шрифту – це ваша відповідальність та одне з необхідних умов популярності сайту.

Зображення заголовка. Тут можна налаштувати заголовок сайту у вигляді зображення. Загалом доступно три опції: без заголовка, заголовка у вигляді статичного зображення та заголовка у вигляді одного з випадкових зображень.

Меню. Містить інструменти для зміни меню залежно від теми. У цьому випадку (див. рис. 7.18.36) маємо лише один тип меню – Primary, який становить головне меню сайту.

За допомогою кнопки "Створити нове меню" можна додати нове меню, видалити пункти меню, що існують, перейменувати пункти меню (наприклад, змінити назву меню за замовчуванням Home чи Blog Feed, які є частинами меню Primary).

Параметри контенту. Опції цього меню дозволяють встановити параметри відображення контенту. Наприклад, Blog Display дозволяє обрати, чи буде запис блогу відображатися у розгорнутому вигляді, чи в скороченому (крім того, налаштувати скорочене відображення посту можна також під час створення тексту посту, додавши у відповідному місці позначку "Читати далі"). Опція "Біографія автора" / Display on single posts дозволяє приховати чи показати деталі про автора під кожним постом. Блок налаштувань Post Details дозволяє показати чи приховати дату посту, категорію, ключові слова та автора під кожним постом. Блок "Головні зображення" призначений для налаштування зображень у шапці сторінки кожного посту.

Віджети. Віджет становить міні-застосунок, який призначений для швидкого доступу до контенту або сервісу. Віджети можна додати у ті місця сторінки, які визначені темою. У нашому прикладі (див. рис. 7.18.37) це бокова панель та 3 області (Footer 1, Footer 2, Footer 3) у нижній частині сторінки. Після обрання місця розташування та натискання кнопки "Додати віджет" з'являється перелік доступних віджетів, наприклад, Facebook Page Plugin (дозволяє переадресувати відвідувачів сайту на вашу фейсбук-сторінку), Instagram (дозволяє відображати фотографії вашого профілю з Instagram), "Пошук", "Перекладач", "Статистика блогу" та інші.

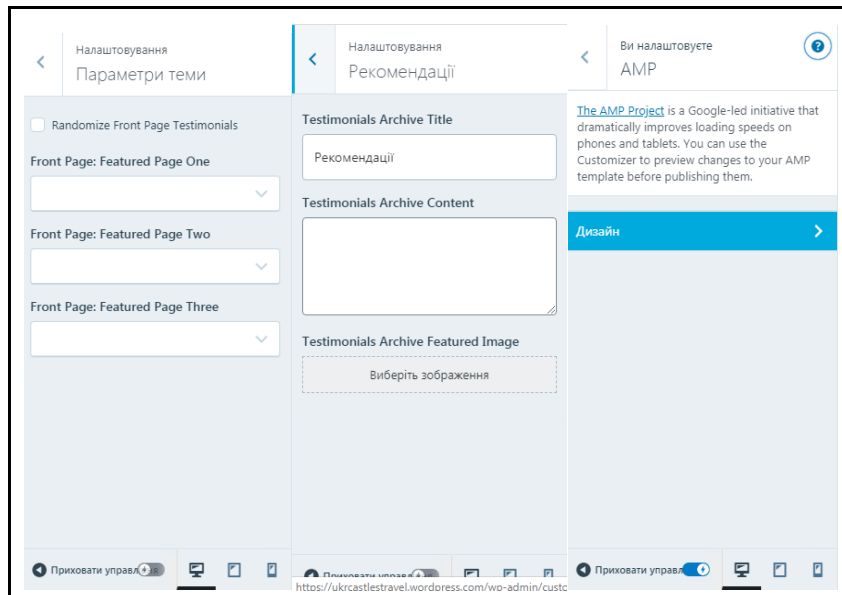
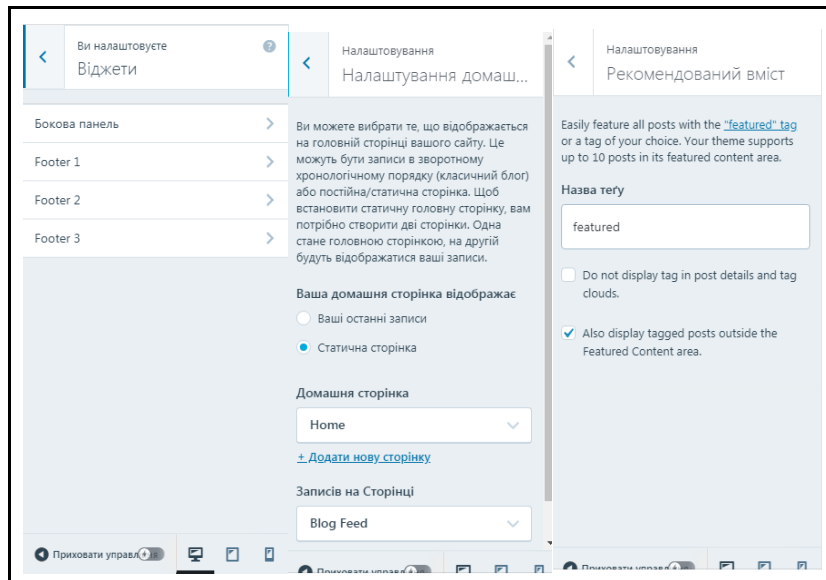


Рис. 7.18.37. Налаштування віджетів та інших параметрів теми

Налаштування домашньої сторінки. Домашня сторінка може бути статичною (потребує створення окремих двох сторінок: одна буде як домашня, інша буде містити всі записи блогу) ти містити перелік усіх записів блогу (активовано за замовчуванням).

Рекомендований вміст. Дозволяє маркувати публікації за допомогою спеціальних теґів.

Параметри теми. Дозволяє встановити 3 головні сторінки та перемішувати їх.

Рекомендації. Дозволяє керувати відгуками на сайт.

AMP. Дозволяє покращити швидкість завантаження сторінок на мобільних пристроях.


У лівому верхньому куті знаходиться кнопка "Оприлюднити", яка дозволяє негайно оновити сайт з урахуванням оновлених опцій.

Практичне виконання роботи

Завдання 18.1. Налаштування профілю користувача WordPress

Внесіть відповідні ім'я, прізвище та встановіть аватар.

Вирішення

Авторизуйтеся на сайті <http://wordpress.com/> шляхом введення логіна та пароля. Після цього на сторінці, що завантажилася (див. рис. 7.18.6), натисніть на позначку  у правому верхньому куті. Безпосередньо тут ви можете ввести ім'я, прізвище та кліком мишею на зображення порожньої фотографії змінити її.

Завдання 18.2. Ознайомтеся із переліком сайтів, на які ви підписані за замовчуванням, та підпишіться на 3 інші

Вирішення

Авторизуйтеся на сайті <http://wordpress.com/> шляхом введення логіна та пароля. Після цього на сторінці, що завантажилася (див. рис. 7.18.6), натисніть на Reader / "Чтиво" в лівому верхньому куті. У центральній частині знайдіть Followed sites / "Мої підписки", а правіше в цьому ж рядку червону кнопку Manage / "Керувати". В центральній частині відобразиться рядок пошуку, в якому можна знайти сайти на будь-яку тематику та підписатися на них, а нижче – перелік сайтів, на які ви вже підписані. Біля кожного з них праворуч є кнопки налаштування сповіщень та підписки чи відписки.

Завдання 18.3. Створення нового сайту

Вирішення

Перейдіть за посиланням "Мої сайти" / "Змінити сайт" (див. рис. 7.18.19) та натисніть Add new site. Це призведе до завантаження майстра створення сайту, який складається з кроків, аналогічних тим, що зображені на рис. 7.18.16 – 7.18.19.

Завдання 18.4. Додавання нового посту на сайт

Вирішення

Перейдіть за пунктом меню "Публікації" (див. рис. 7.18.22) та натисніть на кнопку "Створити новий запис" у центральній частині вікна. Далі створіть новий пост та опублікуйте його, більш детально цей процес зображено на рис. 7.18.22 – 7.18.27.

Завдання 18.5. Перевірка щільності ключових слів

Вирішення

Знайдіть пошуком сервіс семантичного аналізу тексту онлайн, наприклад, <https://istio.com/> чи <https://advego.com/text/seo/>. Всі подібні сервіси мають у центральній частині сторінки місце, куди потрібно скопіювати текст для аналізу та єдину кнопку для запуску процесу аналізу. Після цього ви отримаєте статистику щільності слів, показники нудотності тексту та інші параметри, що дозволяють скоригувати текст та використати слова у ньому в правильному напрямі.

Завдання 18.6. Зміна дизайну сайту

Вирішення

Натисніть пункт меню "Теми" у стовпчику ліворуч (див. рис. 7.18.31). Після цього в центральній частині сторінки завантажаться доступні теми, з яких можна обрати будь-яку та активувати.

Налаштування вже обраної теми доступне за пунктом меню "Налаштувати" в секції "Дизайн" у стовпці ліворуч. Завантажиться сторінка, подібна до рис. 7.18.35, із меню ліворуч, що дозволяє змінити різні елементи дизайну сайту.

Індивідуальні завдання

За результатами виконання роботи має бути оформлений невеликий звіт в електронному вигляді. Він має складатися з титульної сторінки, скриншотів, які підтверджують виконання всіх завдань та коротких пояснень до цих скриншотів.

1. Зареєструйтеся на web-сайті <https://wordpress.com/>.
2. Змініть мову інтерфейсу.
3. Налаштуйте профіль користувача: введіть ім'я, прізвище, встановіть зображення.
4. Ознайомтеся із переліком сайтів, на які ви підписані за замовчуванням, та підпишіться на 3 інші, тематика яких пов'язана з тематикою сайту, який ви збираєтеся створювати в майбутньому.
5. Створіть на web-сайті <https://wordpress.com/> новий сайт типу "блог", оберіть доменне ім'я, що відповідає контенту сайту.
6. Встановіть коректну назву сайту та основні налаштування.
7. Додайте п'ять постів (публікацій) блогу, розташованих у трьох різних категоріях на сайті, кожен із яких має містити головне зображення та зображення у тексті.
8. Текст одного з постів (не менше 2 000 знаків із пробілами) має відповідати критеріям оформлення для подальшої оптимізації у пошукових системах – рівномірно містити ключові слова сторінки із частотою 3 – 5 %, зображення (зі встановленим параметром ALT), виділені слова і т. д. Необхідно описати ці критерії та підтвердити знімками екрана їхнє виконання.
9. Самостійно розгляньте можливості вкладки Scheduled (див. рис. 7.18.30), продемонструйте результат використання її можливостей.
10. Наповніть релевантною інформацією сторінки за замовчуванням Home і Blog Feed (якщо в обраному шаблоні є інші сторінки за замовчуванням – вони мають бути заповнені).
11. Відвідайте сайт студента, який сидить біля вас, та залиште коментарі на його матеріалах.
12. На сайті <https://wordpress.com/> для власного сайту оберіть безкоштовний шаблон відповідно до вимог ефективної організації структури сайту.
13. Змініть кольори, шрифти та головне зображення обраного шаблону. Зробіть відповідні скриншоти, що підтверджують зміну інтерфейсу.
14. Змініть і налаштуйте один віджет та зробіть опис цього процесу із відповідними знімками екрана.
15. Ознайомтеся із переліком контрольних запитань (та матеріалами відповідної лекції) та переконайтеся, що ви розумієте та знаєте відповіді на них.

Запитання для самоконтролю

1. Що таке CMS?
2. У чому полягає різниця між WordPress.org та WordPress.com?
3. Якими є області застосування WordPress.org?
4. Якими є області застосування WordPress.com?
5. Що таке "двокрокова автентифікація"? Наведіть приклади. Чим вона відрізняється від двофакторної автентифікації?
6. Що таке "нотифікація"?
7. Які типи сайтів можна створити за допомогою WordPress.com?
8. Навіщо потрібний режим модерації коментарів?
9. Поясніть, навіщо змінювати тему сайту, встановлену за замовчуванням.
10. Дайте визначення віджета.
11. Які існують вимоги до вибору кольорів та шрифтів сайту?
12. У якій частині сайту потрібно розміщувати головний логотип? Чому?

19. Просування сайтів у мережі "Інтернет"

Мета: познайомитися із засобами керування сайтом CMS WordPress, механізмами збирання статистики, ділення, керування авторами контенту, індексації сайту в пошуковій системі Google.

Ключові терміни: статистика, ділення, авторизація, адміністрування, індексація, Google Search Console.

Теоретичні відомості

Виконаємо вхід на сайт WordPress.com, використавши власний логін і пароль. Перейдемо послідовно за посиланнями "Мій сайт" та "Статистика", де побачимо статистику відвідування сайту (рис. 7.19.1).

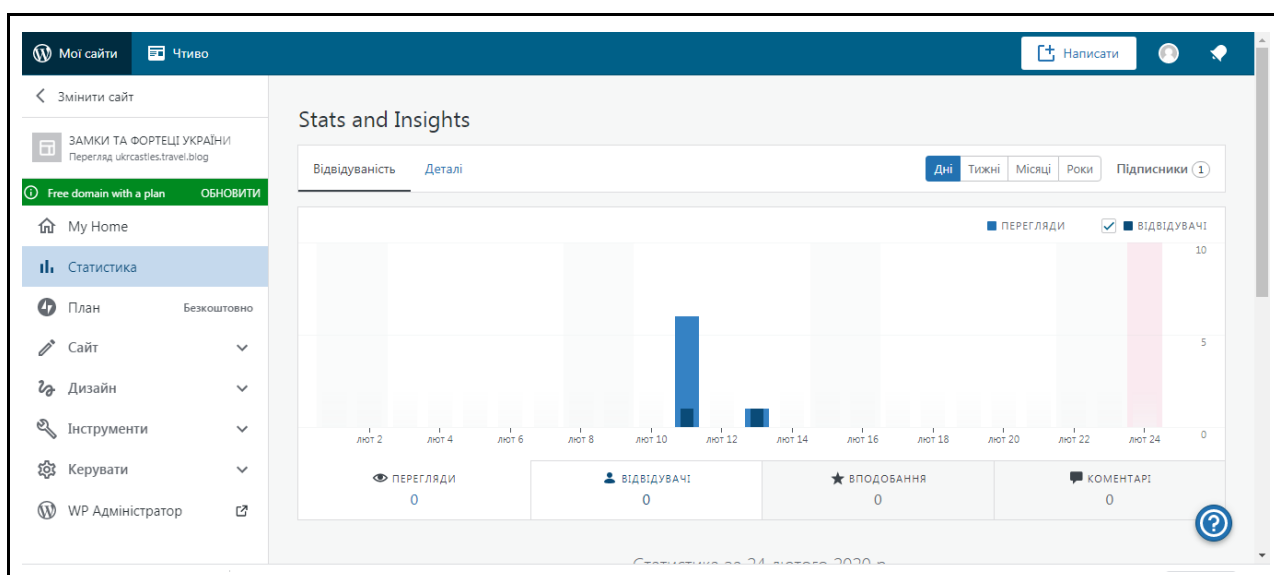


Рис. 7.19.1. Статистика сайту

На вкладці "Відвідування" можна побачити кількість переглядів для кожної сторінки сайту, статистику пошуку та перегляду відео на сайті, джерела переходів (потрапляння) користувачів на сайт, популярність авторів, країни користувачів сайту та навіть кліки користувачів на сайті.

На вкладці "Деталі" відображається більша кількість загальної інформації, така як: кількість постів та переглядів за днями, тижнями та місяцями; підсумкові дані щодо останнього опублікованого посту; загальні підсумки за всі дні існування сайту; інформацію про всіх підписників

та їхній перелік; усі коментарі, згруповані за авторами; всі перегляди матеріалів, згруповані за категоріями; найпопулярніший день та година перегляду сайту; статистика сайту за календарний рік: кількість постів, коментарів та вподобань, слів, середня кількість коментарів, вподобань та слів для посту.

Також на рис. 7.19.1 у правому верхньому куті можна помітити посилання "Підписники" та число у колі біля нього. Якщо перейти за цим посиланням, можна побачити та видалити підписників.

Ділення

Перейдіть до пункту меню "Інструменти" / "Маркетинг" у лівому меню (див. рис. 7.19.1). Сторінка, що має з'явитися, наведена на рис. 7.19.2. Закладка "З'єднання" на цій сторінці дозволяє під'єднати соціальні мережі та сервіси до блогу, щоб публікувати пости автоматично всюди. Закладка "Кнопки "Поділитися" дозволяє розташувати на сторінках із повідомленнями спеціальні кнопки соціальних мереж, щоб читачі могли розповсюдити ваші пости в цих мережах.

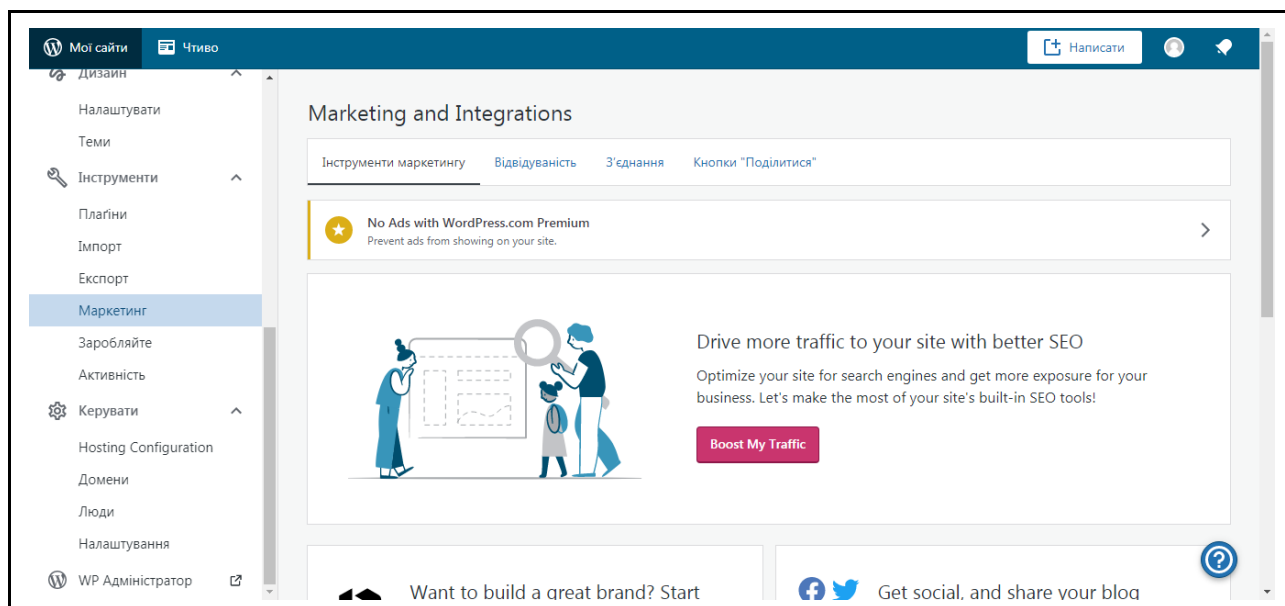


Рис. 7.19.2. Налаштування маркетингових сервісів

Розглянемо приклад з'єднання з сервісом Twitter. Для цього виконаємо клік на кнопці "З'єднати" навпроти нього на складці "З'єднання". Має з'явитися повідомлення, що наведено на рис. 7.19.3. Звертаємо вашу увагу, що подібні повідомлення про взаємодію різних сервісів завжди мають бути уважно прочитані вами, оскільки вони містять перелік даних

та дій, до яких отримує доступ сторонній застосунок (у нашому випадку це WordPress). Отже, ми передаємо сайту WordPress.com права доступу до облікового запису Twitter, і тепер через програму чи сайт WordPress можна побачити пости у Twitter, публікувати нові, оновлювати профіль. Але частина даних Twitter (особисті повідомлення, пароль) залишається недоступною для WordPress.

Потрібно також окремо зазначити про відповідальність публікації постів, що з'являються одночасно в декількох зв'язаних сервісах. Кожна така публікація має бути підготовленою якісно, якщо, наприклад, в пості допущено деякі помилки та виникає необхідність його оновлення у WordPress, не у всіх під'єднаних сервісах це оновлення буде доступним.

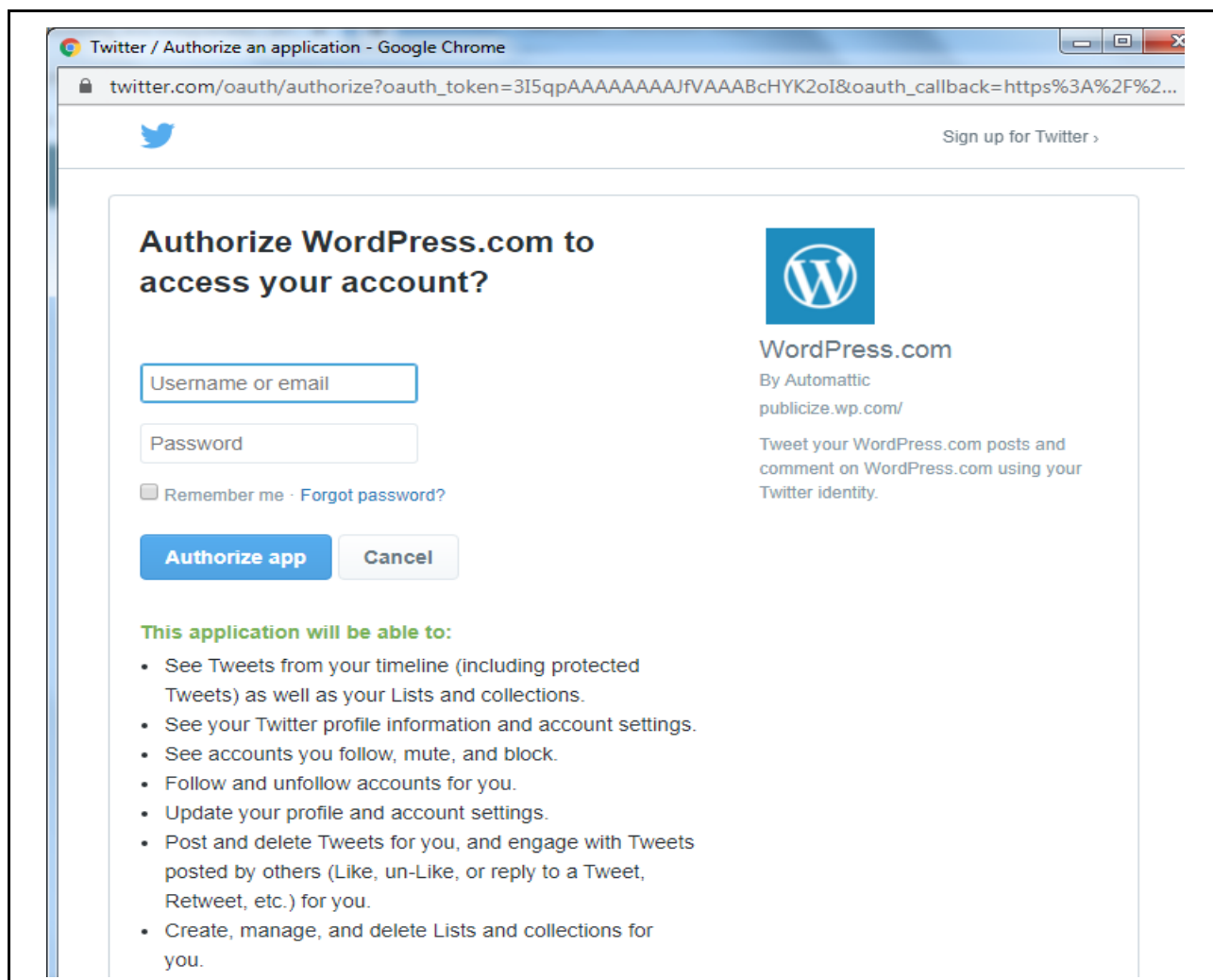


Рис. 7.19.3. Запит на з'єднання WordPress та Twitter

Для продовження ми маємо ввести логін та пароль від акаунта Twitter, натиснути кнопку Authorize app та повернутися до WordPress (рис. 7.19.4),

де потрібно завершити процедуру з'єднання профілів, натиснувши "З'єднати". Має з'явитися повідомлення про успішне з'єднання профілів.

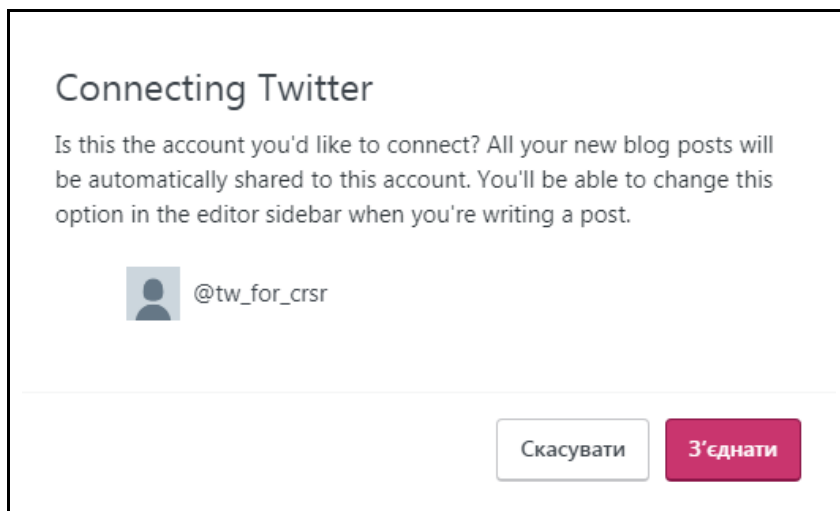


Рис. 7.19.4. Вікно підтвердження з'єднання WordPress із Twitter

Перевіримо, як працює таке ділення. Додамо новий пост про Замок Паланок та опублікуємо його (рис. 7.19.5). У правій частині екрана під час натискання кнопки "Опублікувати" в секції Share this post вже можна побачити під'єднаний акаунт Twitter та за необхідності вимкнути публікацію в ньому.

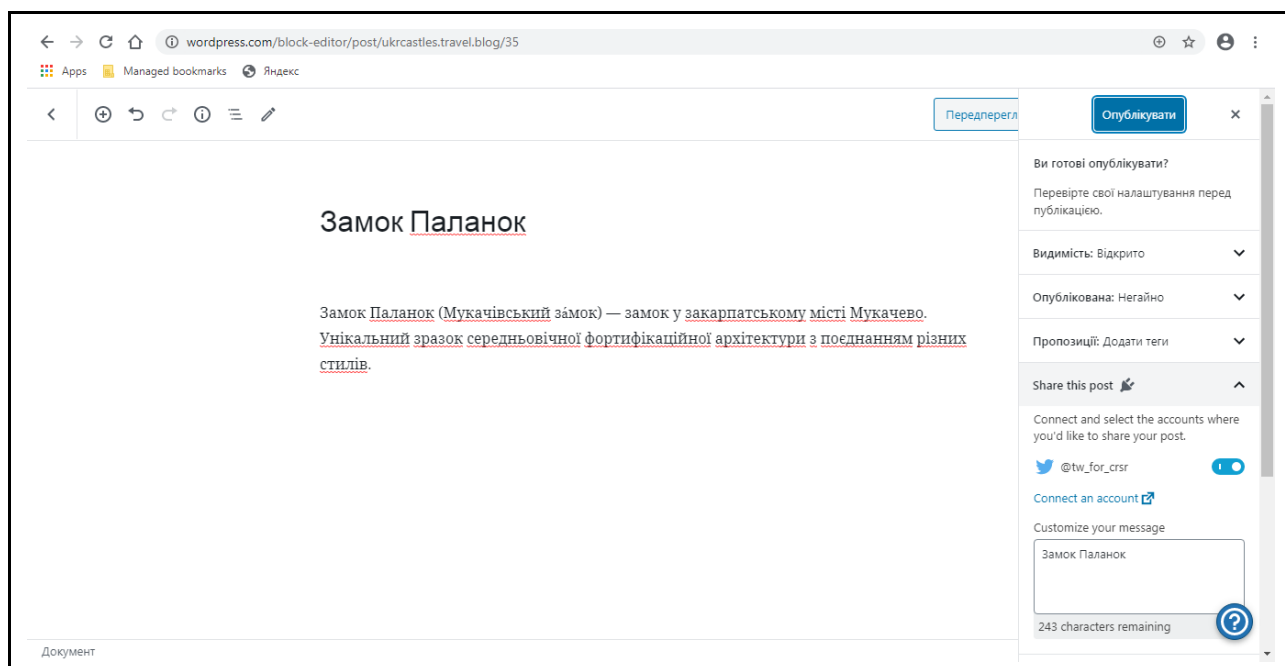


Рис. 7.19.5. Створення нового посту після підключення ділення

Якщо тепер ми перейдемо до зв'язаного акаунта Twitter, ми побачимо, що там з'явилася інформація про новий пост про Замок Паланок та посилання на сайт WordPress, який її опублікував (рис. 7.19.6). Звертаємо увагу, що, навіть якщо цю публікацію зробити приватною на сайті WordPress, вона все одно залишається публічною у Twitter.

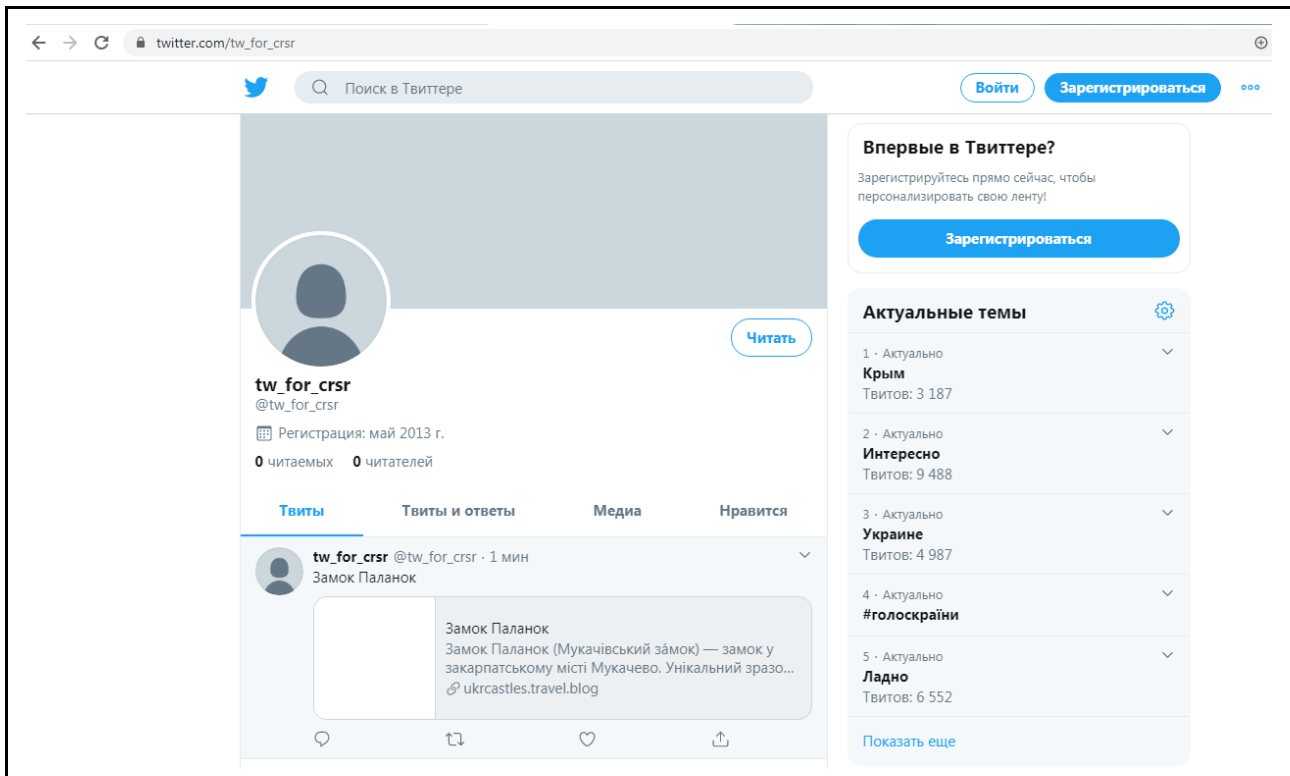


Рис. 7.19.6. Результат публікації нового посту, що з'являється у Twitter

Люди

Перейдіть до пункту меню "Люди" в секції "Керувати" в лівому меню (див. рис. 7.19.1). Сторінка, що має з'явитися, наведена на рис. 7.19.7. Ця сторінка містить 4 закладки, перша з яких містить поточну команду сайту, друга – підписників, третя – підписників за електронною поштою. Остання вкладка "Запрошення" дозволяє запросити когось до участі у роботі сайту. На кожній із попередніх вкладок також є кнопка "Запросити". Список e-mail-підписників також можна завантажити у вигляді csv-файлу.

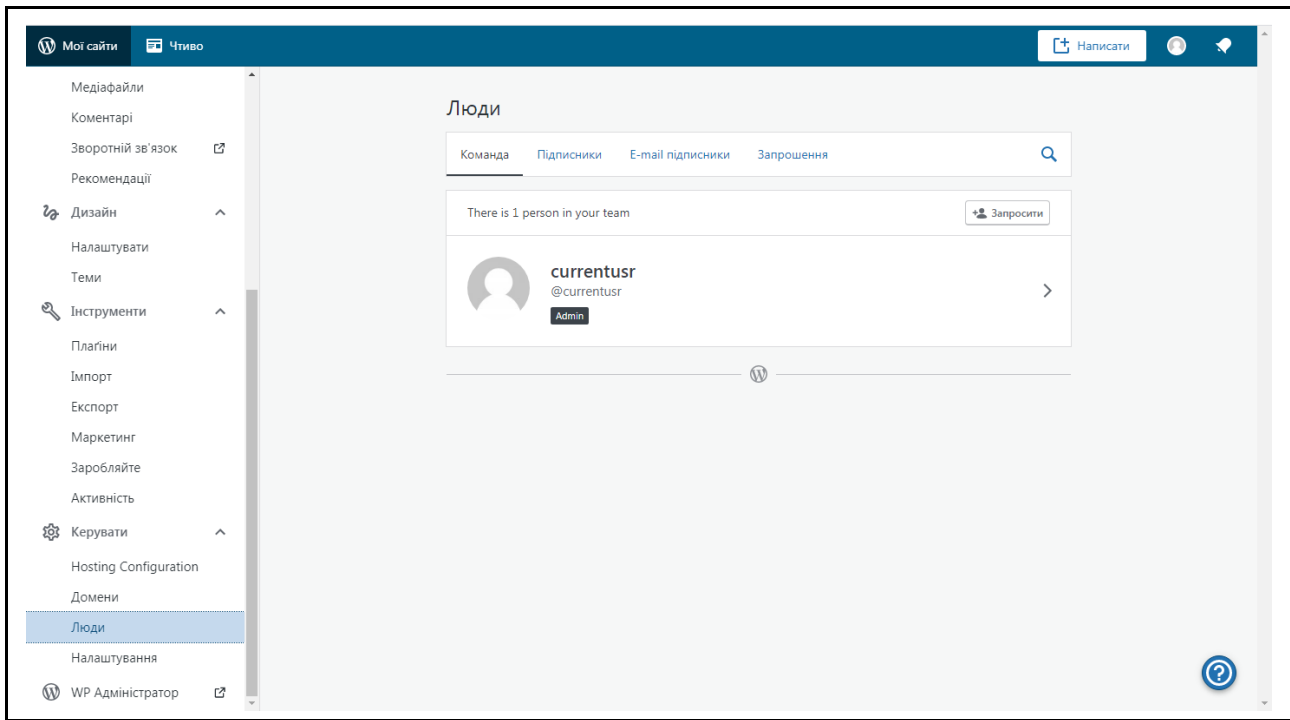


Рис. 7.19.7. Зареєстровані користувачі сайту

Не будемо приділяти увагу пунктам головного меню (див. рис. 7.19.1) "Інструменти" / "Плагіни" та "Керування" / "Домени", оскільки робота із ними можлива лише в платних підписках на сайті WordPress.com.

На сторінці "Керувати" / "Налаштування" є галочка Do not allow search engines to index my site, яка забороняє пошуковим системам індексувати сайт, за замовчуванням індексація дозволена.

Онлайн-адміністрування

Сайтом можна керувати не лише через інтерфейс WordPress.com, але й у онлайн-режимі. Для цього потрібно використати гіперпосилання вигляду: <https://ukrcastlestravel.wordpress.com/wp-admin>, де за запитом ввести свій логін та пароль від WordPress. Можна також обрати пункт меню WP Administrator, яке розташоване на останній позиції меню ліворуч (див. рис. 7.19.7). У результаті має завантажитися сторінка, наведена на рис. 7.19.8.

Ця сторінка надає декілька можливостей, які важко або неможливо знайти на звичайній панелі керування, яку ми використовували раніше. Наприклад, у розділі "Записи" є підрозділ "Категорії" для зручного додавання, видалення та перегляду категорій.

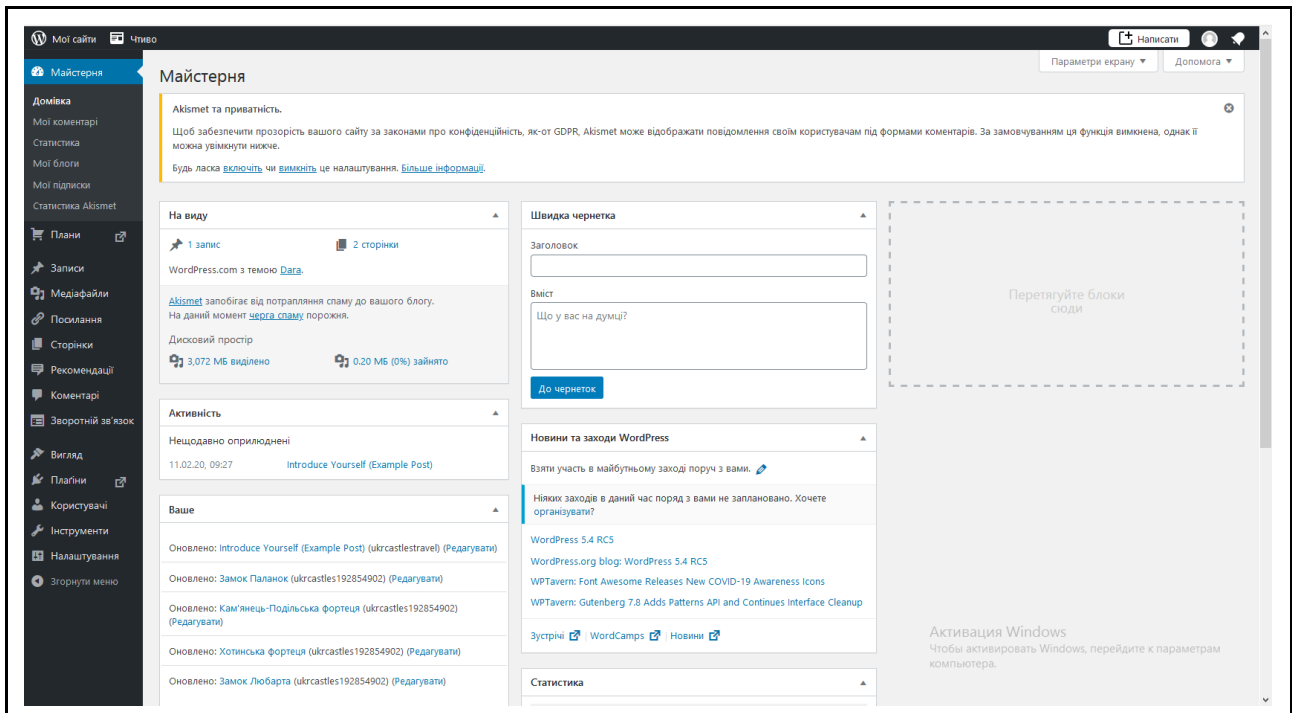


Рис. 7.19.8. Зовнішній вигляд панелі онлайн-адміністрування

Webmaster Tools (Site Verification Services)

Давайте перевіримо, чи проіндексований уже сайт пошуковою системою Google. Для цього потрібно виконати Google-пошук із застосуванням оператора site: (рис. 7.19.9). Як можна побачити, шуканий сайт ще відсутній у цій пошуковій системі. Індекссація – це процес додавання сторінок сайту в індекс (внутрішню базу даних) пошукової системи, щоб вона могла видавати ці сторінки в результаті пошуку. Цей процес виконується постійно та автоматично пошуковими роботами-"павуками".

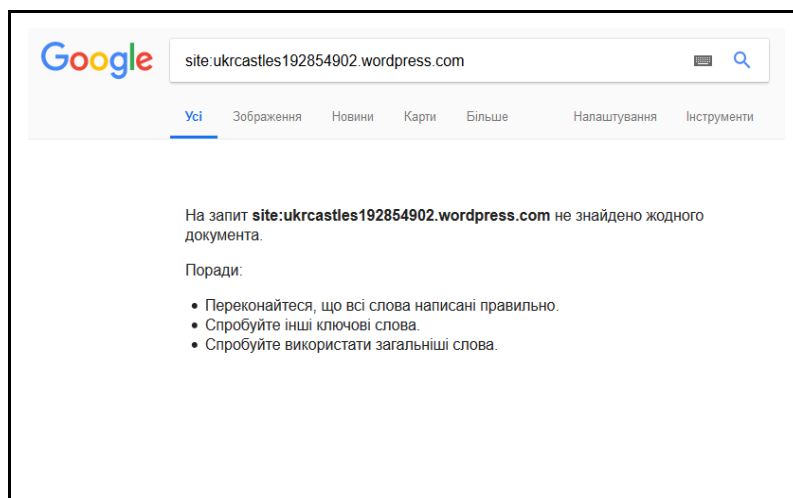


Рис. 7.19.9. Пошук посилань на власний сайт

Зазначимо також, що, для того щоб сайт міг бути проіндексований, налаштування приватності у розділі "Налаштування" / "Конфіденційність" повинно мати значення "Публічно".

Як вказує WordPress.com, нові сайти зазвичай проходять індексацію впродовж 4 – 6 тижнів, такий саме час потрібний для оновлення, якщо налаштування приватності були змінені.

Індексацію можна прискорити, підтвердивши, що ви є власником цього сайту. Для цього потрібно використати відповідний сервіс пошукової системи та налаштування WordPress.

Для того щоб сайт був розташований високо у результатах пошукових систем, недостатньо виконати лише його індексацію. Необхідно також зробити як технічну оптимізацію самого сайту, так і оптимізацію тексту на ньому. Оптимізація полягає у написанні текстів із використанням ключових слів, що притаманні цьому сайту та цій сторінці із необхідною густиною слів. Крім того, контент на сайті має бути унікальним, високої якості та регулярно оновлюватися. Втім, оптимізація не гарантує успішності сайту, так часом як і зворотне твердження також є правильним: сайт, який не оптимізовано, може бути успішним.

Розглянемо використання сервісу Google Search Console. Це безкоштовний сервіс, який дозволяє отримувати інформацію про те, як ваш сайт представлений у результатах пошуку Google, оптимізувати його контент і усувати можливі неполадки. Не обов'язково реєструватися в цьому сервісі, щоб ваш сайт потрапив в індекс Google. Однак це дозволить зробити індексацію швидше, дізнатися, як ваш ресурс сканується нашими системами, і поліпшити його взаємодію з ними.

Отже, інструменти Google Search Console дозволяють:

- переконатися, що Google може знайти і просканувати ваш сайт;
- виявити проблеми з індексуванням і надіслати запит на повторне індексування нового або оновленого контенту;
- оцінити трафік, який надходить на сайт із Google-пошуку: як часто і за якими запитамі посилання на ваші сторінки з'являються в результатах, як часто користувачі натискають на них;
- отримувати повідомлення про проблеми з індексуванням;
- дізнатися, які сайти посилаються на ваш ресурс.

Перейдіть за посиланням <https://www.google.com/webmasters/tools/> та введіть логін і пароль власного облікового запису Google. Після авторизації вам потрібно обрати в лівому верхньому куті опцію "Додати ресурс"

та ввести повну адресу до вашого блогу із префіксом `https://` та натиснути на "Продовжити" (рис. 7.19.10).

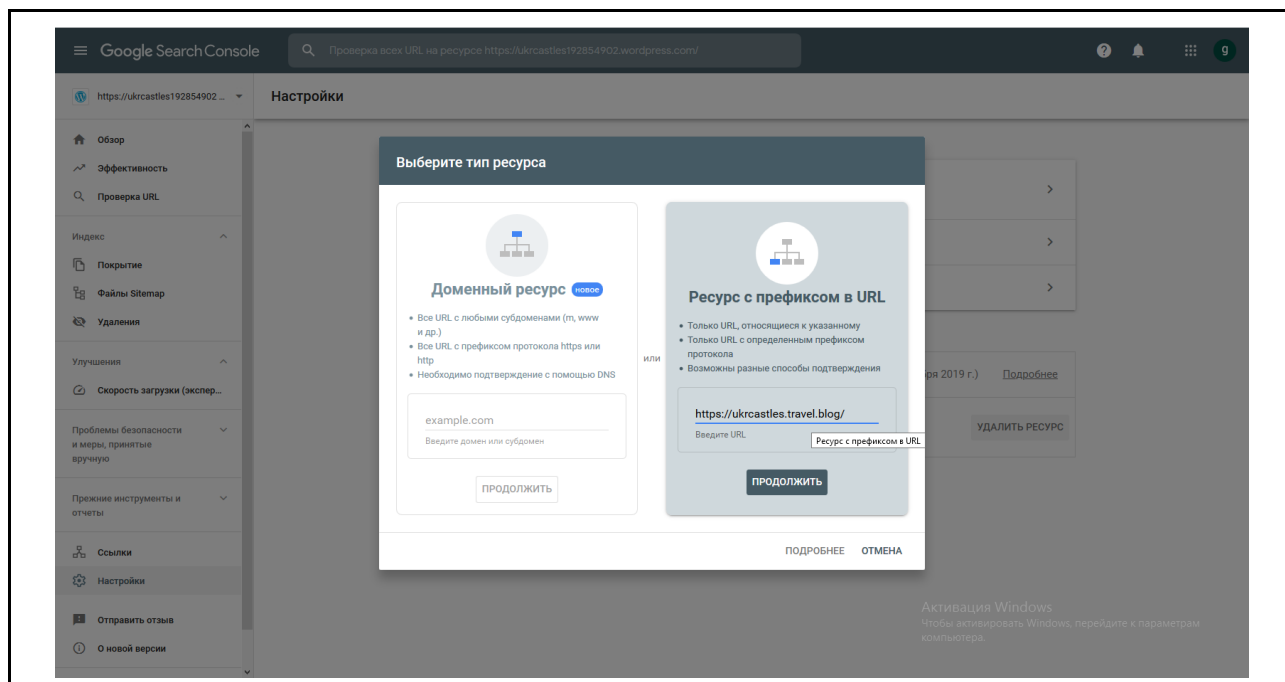


Рис. 7.19.10. Додавання сайту за допомогою Google Search Console

Вам буде запропоновано рекомендований спосіб підтвердити (рис. 7.19.11), що саме ви є власником сайту, для налаштування WordPress нам потрібно обрати "Тег HTML" (рис. 7.19.11).

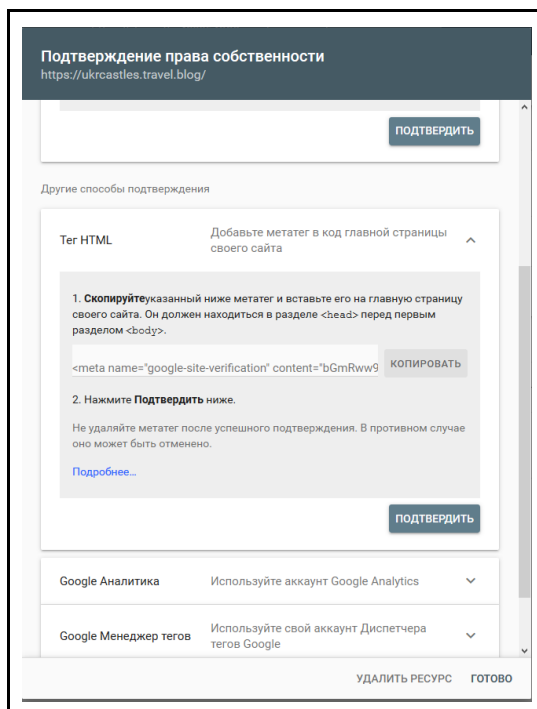


Рис. 7.19.11. Обрання способу підтвердження права власності

Згідно з інструкцією нам необхідно скопіювати значення, наведене на цій сторінці (див. рис. 7.19.11, у нашому випадку це `<meta name="google-site-verification" content="bGmRww9yOG0Hc_T6tHcODHIZbxHQ3D5Kg6ix-HZGrGg" />`) та вставити його у відповідне поле на нашому сайті WordPress.

Знайти це поле можна на панелі онлайн-адміністрування в пункті меню "Інструменти" (рис. 7.19.12). Там потрібно знайти необхідну пошукову систему, ввести скопійоване значення та натиснути "Зберегти зміни".

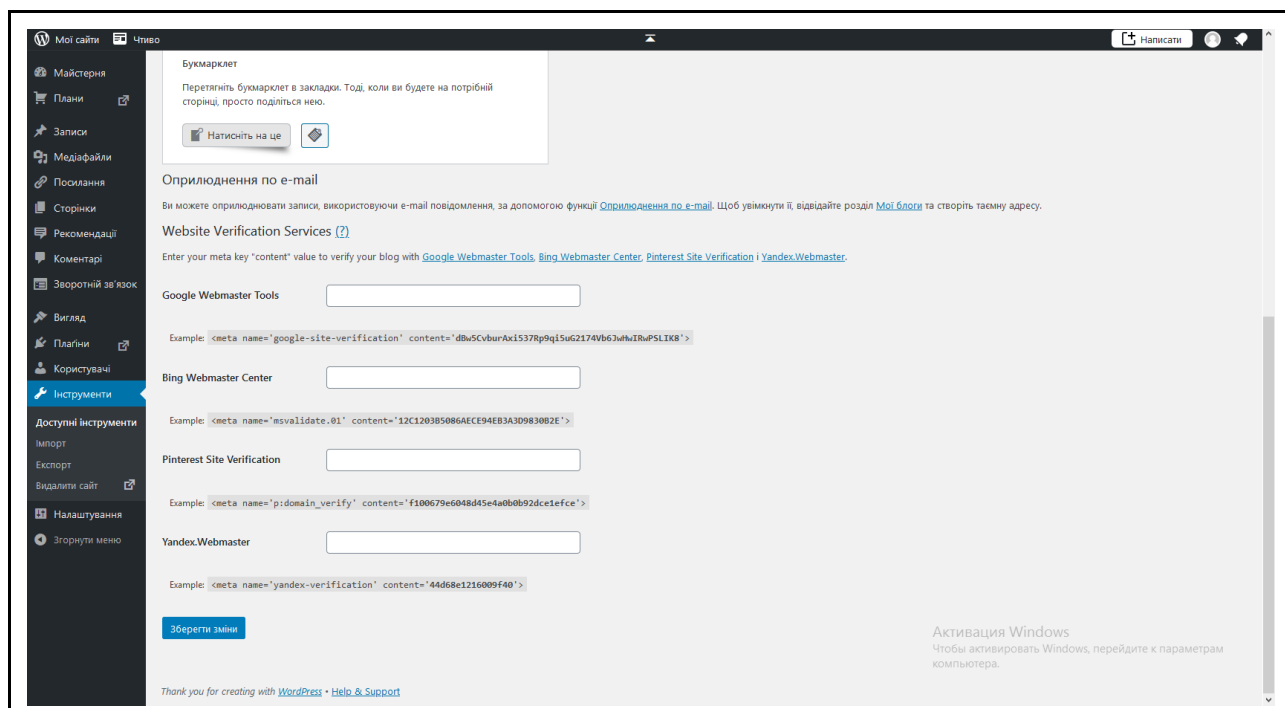


Рис. 7.19.12. Верифікація сайту в пошуковій системі Google

Тепер поверніться знову до Google Search Console та натисніть "Підтвердити". У результаті має з'явитися сторінка, наведена на рис. 7.19.13.

Також, незважаючи на те що WordPress.com стверджує, що він сам відправляє файли sitemap.xml до пошукових систем, можна відправити їх також за допомогою Google Search Console, клацнувши на "Файли Sitemap" у лівій частині сторінки, наведеної на рис. 7.19.13.

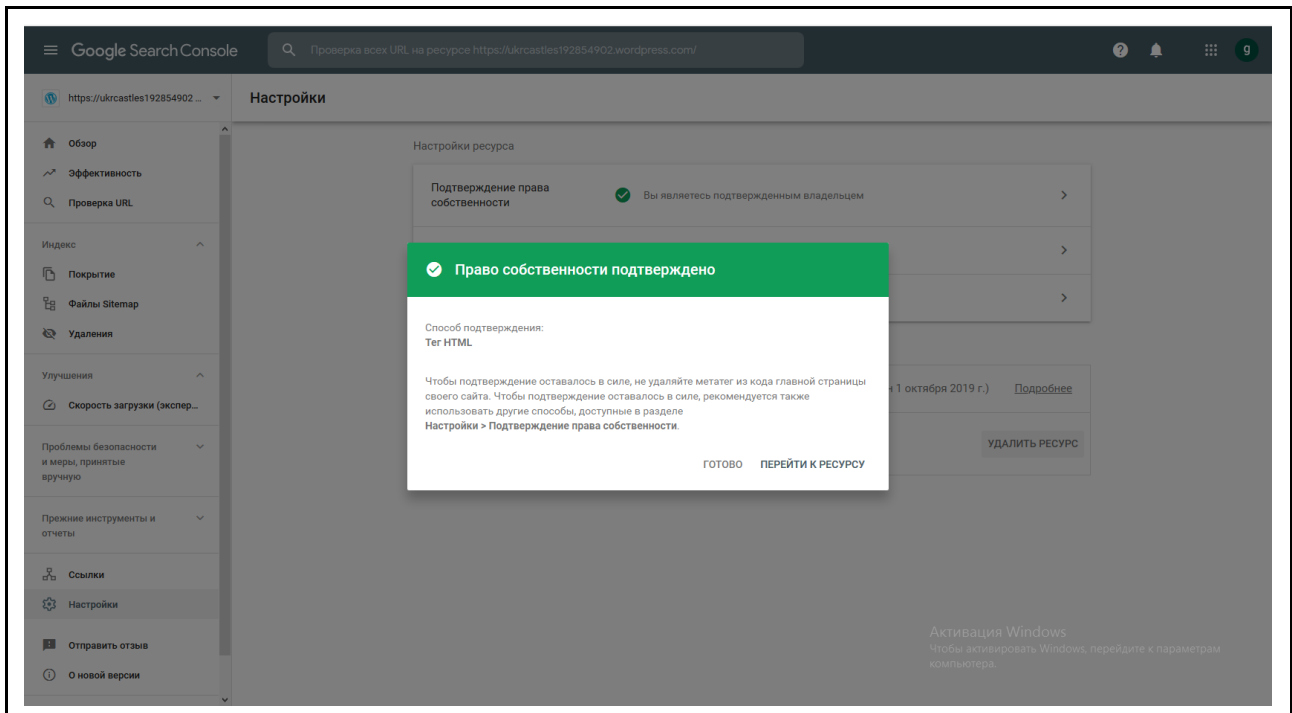


Рис. 7.19.13. Сторінка вдалого підтвердження власності на сайт

На рис. 7.19.14 наведено результат запиту в пошуковій системі після того, як сайт було проіндексовано. Із моменту додавання сайту в чергу на індексацію (див. рис. 7.19.13) минуло близько 6 годин.

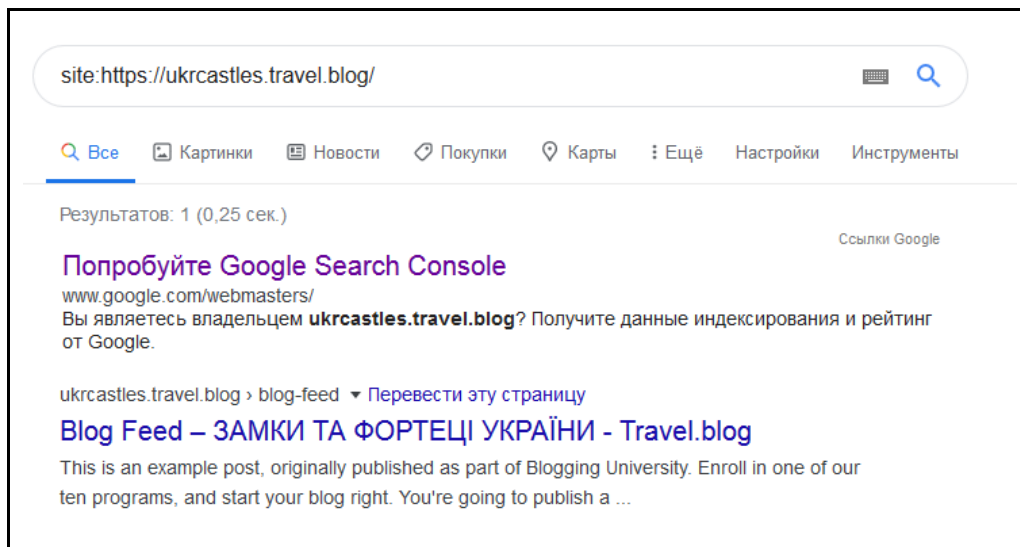


Рис. 7.19.14. Вдалиий пошук сайту в Google

Практичне виконання роботи

Завдання 19.1. Налаштування модерації коментарів

Вирішення

Перейдіть за посиланням "Налаштування" в секції "Керувати" в меню ліворуч. На останній вкладці "Обговорення" сторінки, що відкрилася, в нижній частині є секція "Модерація коментарів".

Завдання 19.2. Перевірка, чи проіндексований сайт у пошуковій системі Google

Вирішення

Перейдіть на сторінку пошуку www.google.com.ua та внесіть у пошукове вікно адресу сайту, додавши "site:" перед адресою без пробілу. Якщо пошукова система нічого не знаходить, сайт ще не було проіндексовано.

Індивідуальні завдання

За результатами виконання роботи має бути оформлений невеликий звіт в електронному вигляді. Він має складатися з титульної сторінки, скриншотів, які підтверджують виконання всіх завдань та коротких пояснень до цих скриншотів.

1. Оновіть статистику відвідування сайту для студента, який сидить біля вас, та надайте підтвердження у вигляді скриншотів, що вона оновлена.

2. Підпишіться на 2 сайти інших студентів групи.

3. Налаштуйте модерацію коментарів таким чином, щоб затримувати для підтвердження тільки будь-які коментарі з трьома гіперпосиланнями в ньому. Перевірте, чи цей механізм працює, та надайте підтвердження у вигляді скриншотів та опису.

4. Налаштуйте ділення постами з одним із сторонніх соціальних інструментів за вашим вибором, наприклад, Twitter, Facebook, Instagram тощо.

5. Надішліть двом сусіднім студентам (налаштування "Люди") запрошення на: вподобання і відстежування вашого сайту та на редагування

(роль редактора) матеріалів. Попросіть їх підтвердити запрошення, перевірте, чи отримав зовнішній редактор право писати пости на вашому сайті чи редагувати ті, що вже існують. Ознайомтеся із описом ролей WordPress.com, що існують.

6. Ознайомтеся з інтерфейсом онлайн-адміністрування сайту та вкажіть у текстовому та/або графічному форматі відмінності між ним та традиційним інтерфейсом.

7. Верифікуйте сайт у пошуковій системі Google. Підтвердіть скріншотами, що сайт проіндексовано.

8. Ознайомтеся із переліком контрольних запитань (та матеріалами відповідної лекції) та переконайтеся, що ви розумієте та знаєте відповіді на них.

Запитання для самоконтролю

1. Подумайте, навіщо потрібен інтерфейс онлайн-адміністрування.
2. Навіщо потрібен механізм "Ділення"?
3. Які можливості надає сторінка WordPress.com "Люди"?
4. Навіщо потрібно завжди читати інформаційні вікна, аналогічні тому, що зображено на рис. 7.19.3? Наведіть приклади.
5. Навіщо потрібні файли robots.txt та sitemap.xml?
6. Що таке "пошуковий робот"? Як він працює?
7. Що таке Google Search Console?
8. Навіщо підтверджувати право власності на сайт?

Тестові завдання за розділом 7

1. Дайте визначення двофакторної автентифікації:

- а) така автентифікація вимагає другого підтвердження операції засобами, що не пов'язані безпосередньо із мережею "Інтернет" (SMS, властивість речі і т. д.);
- б) це те ж саме, що й однофакторна автентифікація, лише виконується 2 рази підряд;
- в) такий вид автентифікації вимагає особистої присутності особи, яка ініціює транзакцію;
- г) це окремий тип протоколу для комунікації у незахищеному середовищі.

2. Які типи сайтів можна створити із використанням WordPress:

- а) блог;
- б) рекламний сайт;
- в) статичний сайт;
- г) портфоліо?

3. Оберіть коректні твердження:

- а) WordPress.org краще підходить для новачків, ніж WordPress.com;
- б) WordPress.com надає послуги безкоштовного хостингу;
- в) WordPress.org надає послуги безкоштовного хостингу;
- г) WordPress.com краще підходить для новачків, ніж WordPress.org.

4. Оберіть найбільш правильне визначення "павука" пошукової системи:

- а) відвідує сторінки закріпленого за ним сегменту мережі "Інтернет", перевіряє їх та видаляє з індексу пошукової системи;
- б) виконує пошуковий запит користувача пошукової системи;
- в) відвідує сторінки закріпленого за ним сегменту мережі "Інтернет", перевіряє їх та виконує індексацію;
- г) збирає статистику відвідування закріплених за ним сайтів користувачами.

5. Оберіть недоліки динамічних сайтів:

- а) необхідність запускати файли на оброблення сервером, щоб отримати HTML;

- б) незручна підтримка цілісності;
- в) необхідність знати серверну мову програмування (а не тільки HTML);
- г) більшість контенту є зображеннями.

6. Оберіть переваги статичних сайтів:

- а) незначна кількість необхідних технічних ресурсів;
- б) можливість легко розподілити користувачів за ролями;
- в) зручна підтримка цілісності;
- г) дешеве розроблення.

7. Оберіть правильні твердження щодо необхідності оптимізації сайту:

- а) в першу чергу це є елементом маркетингу та показує перевагу одного сайту над іншим;
- б) більшість потенційних клієнтів знаходять сайт засобами пошукових систем;
- в) сайт, який не було оптимізовано, може бути успішним;
- г) оптимізація гарантує успішність сайту.

8. Навіщо потрібен протокол HTTPS:

- а) цей протокол введений тому, що доменні імена для протоколу HTTP вже вичерпані, сам протокол не відрізняється від HTTP;
- б) цей протокол дозволяє передавати дані у зашифрованому вигляді;
- в) цей протокол використовується тільки для безпеки банківських операцій;
- г) HTTPS є більш сучасним протоколом та працює швидше, ніж HTTP?

9. Оберіть технічні прийоми підтримки високого рівня безпеки робочого місця:

- а) оновлення операційної системи;
- б) оновлення антивірусу та його баз;
- в) використання сильних паролів;
- г) використання протоколу HTTPS;
- д) ігнорування листів невідомо від кого.

10. Чим відрізняються концепції Web 2.0 та Web 3.0:

- а) концепція 2.0 не описує якість контенту жодним чином;
- б) концепція 3.0 говорить про масове використання соціальних мереж для розповсюдження інформації;
- в) концепція 2.0 говорить про якість контенту в соціальних мережах;
- г) згідно з концепцією 3.0 головна вимога до контенту – це якість?

11. Що є найголовнішим на web-сайті:

- а) зовнішній вигляд, оскільки яким би якісним не був контент, на сайті не буде користувачів, якщо він має незручний інтерфейс;
- б) якісний, унікальний, корисний контент;
- в) швидкість завантаження, оскільки користувачі не люблять довго чекати завантаження сайту;
- г) зручне доменне ім'я?

12. Що таке CTR:

- а) відношення кількості кліків по рекламному банеру до кількості його показів;
- б) відношення кількості відвідувачів сайту, які виконали на ньому якісь цільові дії, до загальної кількості відвідувачів сайту, виражене у відсотках;
- в) оплата за кількість показів реклами користувачам;
- г) оплата за кількість відвідувачів сайту користувачами?

13. Що таке "конверсія":

- а) відношення кількості кліків по рекламному банеру до кількості його показів;
- б) відношення кількості відвідувачів сайту, які виконали на ньому якісь цільові дії, до загальної кількості відвідувачів сайту, виражене у відсотках;
- в) оплата за кількість показів реклами користувачам;
- г) оплата за кількість відвідувачів сайту користувачами?

14. Що таке "фішинг":

- а) "нігерійський спам";
- б) процес демонстрації небажаної реклами на комп'ютері чи сервісі (наприклад, в електронній пошті);

- в) надсилання листів, які нагадують листи від офіційних установ, з проханням надати пароль чи іншу конфіденційну інформацію;
- г) використання сайтом технології cookies?

15. Які збитки можна отримати, розміщуючи рекламу на власному популярному сайті:

- а) реклама зазвичай розміщується на популярних місцях сторінки;
- б) розміщення реклами на власному сайті – це майже завжди втрата іміджу;
- в) розміщення реклами на власному сайті має лише переваги у вигляді заробітку від реклами;
- г) будь-які збитки від реклами компенсуються заробітком на ній?

Відповіді до тестових завдань

Відповіді до тестових завдань за розділом 5

- | | | | | |
|--------|--------|-----------------|--------|--------|
| 1. в; | 2. а; | 3. в, г; | 4. а; | 5. г; |
| 6. д; | 7. а; | 8. а, д; | 9. г; | 10. в; |
| 11. г; | 12. б; | 13. а, б, в, г; | 14. в; | 15. г. |

Відповіді до тестових завдань за розділом 6

- | | | | | |
|-------------|-------------|-----------|----------|-----------|
| 1. а, б, г; | 2. б, в, г; | 3. в, г; | 4. а, в; | 5. г; |
| 6. а, б, в; | 7. а; | 8. а, в; | 9. в; | 10. а, г; |
| 11. б; | 12. в; | 13. б, в; | 14. в; | 15. б. |

Відповіді до тестових завдань за розділом 7

- | | | | | |
|----------|-------------|----------|----------|-----------|
| 1. а; | 2. а, б, г; | 3. б, г; | 4. в; | 5. а, в; |
| 6. а, г; | 7. б, в; | 8. б; | 9. а, б; | 10. а, г; |
| 11. б; | 12. а; | 13. б; | 14. в; | 15. а, б. |

Рекомендована література

1. Бартлетт Д. WordPress для начинающих / Д. Бартлетт ; [пер. с англ. М. А. Райтмана]. – Москва : Изд-во "Э", 2017. – 208 с.
2. Грачев А. Г. Создаем свой сайт на WordPress: быстро, легко и бесплатно. Работа с CMS WordPress 3 / А. Г. Грачев. – Санкт-Петербург : Питер, 2011. – 288 с.
3. Огурцов В. В. Основы Web та Web-дизайн, програмування на боці клієнта. Лабораторний практикум з навчальної дисципліни "Web-технології та Web-дизайн" / В. В. Огурцов, Д. В. Гриньов, О. В. Щербаков. – Харків : ХНЕУ ім. С. Кузнеця, 2015. – 228 с.
4. Уильямс Б. WordPress для профессионалов / Б. Уильямс, Д. Дэмстра, Х. Стэрн. – Санкт-Петербург : Питер, 2014. – 464 с.
5. Hussey T. WordPress Absolute Beginner's Guide / T. Hussey. – USA : Que Publishing, 2014. – 400 с.
6. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. – Санкт-Петербург : Изд-во "Питер Пресс", 2017. – 272 с.
7. WordPress all-in-one for dummies / Lisa Sabin-Wilson, Cory Miller, Kevin Palmer et al. – USA : Wiley publishing, 2011. – 916 с.
8. Довідник Web-мов [Електронний ресурс]. – Режим доступу : www.spravkaWeb.ru.
9. Сайт персональних навчальних систем ХНЕУ ім. С. Кузнеця. – Режим доступу : <https://pns.hneu.edu.ua/course/view.php?id=4247>.
10. Шпаргалка по Flexbox CSS [Электронный ресурс]. – Режим доступа : <http://tpverstak.ru/flex-cheatsheet>.
11. Bootstrap 4 – Сетка [Электронный ресурс]. – Режим доступа : <https://itchief.ru/bootstrap/grid-v4>.
12. HTML-довідник [Електронний ресурс]. – Режим доступу : <http://htmlbook.ru>.
13. Julius N. WordPress for beginners / N. Julius [Electronic resource]. – Access mode : <http://www.historic-kirkwood.org/wp-content/uploads/2013/12/wordpress-for-beginners-free-ebook-wp35.pdf>.

Зміст

Вступ.....	3
Розділ 5. Основи web-дизайну	5
13. Знайомство з web. Перша HTML-сторінка	7
Індивідуальні завдання	25
Запитання для самоконтролю	25
14. Таблична верстка сайту. Створення форм на web-сторінці	26
Індивідуальні завдання	41
Запитання для самоконтролю	45
Тестові завдання за розділом 5	46
Розділ 6. Створення сайтів (блогів) з використанням HTML5 та CSS3.....	49
15. Форматування елементів web-сторінки засобами CSS.....	50
Індивідуальні завдання	60
Запитання для самоконтролю	61
16. Блокова верстка сайту	62
Індивідуальні завдання	75
Запитання для самоконтролю	79
17. Адаптивна верстка сайту. Створення сайтів з використанням технології Bootstrap.....	80
Індивідуальні завдання	96
Запитання для самоконтролю	96
Тестові завдання за розділом 6	97
Розділ 7. Інформаційні системи створення, редагування та управління контентом сайту	100
18. Система управління контентом сайту WordPress	102
Індивідуальні завдання	132
Запитання для самоконтролю	134
19. Просування сайтів у мережі "Інтернет"	135
Індивідуальні завдання	146
Запитання для самоконтролю	147
Тестові завдання за розділом 7	148
Відповіді до тестових завдань.....	152
Рекомендована література.....	153

НАВЧАЛЬНЕ ВИДАННЯ

Удовенко Сергій Григорович
Затхей Володимир Анатолійович
Гороховатський Олексій Володимирович та ін.

ІНФОРМАТИКА В СФЕРІ КОМУНІКАЦІЙ

Навчально-практичний посібник
у 3-х частинах

Частина 3. Використання web-технологій
у сфері комунікацій

За загальною редакцією
д-ра. техн. наук, професора С. Г. Удовенка

Відповідальний за видання *С. Г. Удовенко*

Відповідальний редактор *М. М. Оленич*

Редактор *В. Ю. Степаненко*

Коректор *В. Ю. Степаненко*

План 2020 р. Поз. № 19-ЕНП. Обсяг 155 с.

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А

Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
ДК № 4853 від 20.02.2015 р.