МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

WEB-ТЕХНОЛОГІЇ
ЛАБОРАТОРНИЙ ПРАКТИКУМ

Навчальний посібник, 2-ге видання

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
за освітньою програмою «Автоматизація та комп'ютерно-інтегровані технології кібер-енергетичних систем»
спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології

Укладач: О. С. Бунке

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2022

Рецензент *Сірий О.А.,* к.т.н., доцент,
кафедра теплоенергетики, Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"

Відповідальний редактор *Новіков, П.В.*, к.т.н.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 6 від 24.06.2022 р.)
за поданням Вченої ради Теплоенергетичного факультету
(протокол № 8 від 31.05.2022 р.)

Посібник розроблений на підставі силабусу навчальної дисципліни «WEB-технології» та призначений для проведення лабораторних занять, підвищення розуміння основ WEB-технологій.

Призначений для студентів, які навчаються за освітньою програмою підготовки бакалаврів за спеціальністю 151 "Автоматизація та комп'ютерно-інтегровані технології".

Спрямований на формування у студентів умінь та навичок з верстки та програмування WEB-документів. Забезпечує студентів необхідними теоретичними знаннями для виконання практичних завдань, запланованих впродовж семестру.

Реєстр. № НП 21/22-696. Обсяг 1,92 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056
https://kpi.ua
Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

Зміст

ВСТУП

Дисципліна «Web-технології» є базовою частиною професійного циклу дисциплін. Це логічне продовження дисциплін «Інформатика і програмування», «Інформаційні технології», «Програмна інженерія», «Обчислювальні системи, мережі та телекомунікації», «Бази даних». Використання даної дисципліни необхідно як попереднє для курсів професійного циклу («Високорівневі методи інформатики та програмування», «Предметно-орієнтовані ЕІС»), а також для написання Випускної кваліфікаційної роботи.

Метою освоєння дисципліни «Web-технології» є розширення світогляду і формування знань, уявлень і навичок про промислову розробку інформаційних Web-ресурсів.

Основними завданнями дисципліни є:

- формування навичок роботи в мережі з Web-ресурсами та Web-послугами;
- формування уявлення про структуру та принципи функціонування і розробки сучасних Web-ресурсів;
- ознайомлення з основними методами сучасних Web-технологій у професійній діяльності, а також із засобами підтримки прийняття рішень і можливостями їх застосування в задачах управління інформаційними ресурсами підприємства.

В результаті вивчення дисципліни студент повинен:

1) знати:

- поняття Web-технології, Web-послуги та їх класифікацію;
- поняття Web-ресурси та їх класифікацію;
- характеристики основних секторів ринку Web-послуг і питання використання ділової інформації;
- технології створення Web-додатків і інформаційних ресурсів;
- мови клієнтських і серверних сценаріїв;

2) вміти:

- організувати роботу щодо доступу до ділової інформації на базі сучасних Web-технологій;
- проводити аналіз, проектування, якісні показники Web-додатків і інформаційних ресурсів;
- організовувати процес розробки Web-додатків і інформаційних ресурсів, вести документацію відповідно до сучасних стандартів.

1. Розробка статичної верстки WEB-сторінки (основні типи сторінок сайту обраної тематики)

Мета роботи: розробити WEB-сторінку, використовуючи різні елементи форматування вмісту і тексту.

1.1. Теоретичні відомості

HTML – це мова форматування тексту. HTML-документ – це просто текстовий файл з розширенням * .html.

Фрагмент тексту, укладений між символами <i>, називається тегом (від англійського слова TAG, в перекладі – мітка).

Тег – це символьна конструкція з <(відкриває) і> (закриває) кутових дужок, між якими знаходиться конкретний символ або рядок символів, які веліли браузеру відображення такого змісту документа відповідно до їх призначення.

Мова HTML використовує різні теги, що вводяться в текстові документи, які вказують, яким чином інформація повинна зчитуватися WEB-браузером. Більшість тегів HTML парні, тег відкриває і закриває. Вони охоплюють текст, що позначений ними. Тег пари, що закриває завжди починається з прямої косої риси. Теги цього тексту визначають, яким чином текстова інформація і графіка повинні бути представлені на екрані. Крім тегів <HTML>, <HEAD>, <TITLE> і <BODY>, які відповідно до стандартів HTML і WWW повинні бути присутніми в кожній WEB-сторінці, при введенні тексту HTML-файлу для розбиття його на смислові групи і стильового оформлення тексту використовуються кілька тегів розмітки.

Таблиця 1.1 – Теги, що задають структуру документа

<!DOCTYPE ..... >	Декларування типу документа	Використовується для вказівки, з яким стандартом HTML сумісний документ

Продовження таблиці 1.1

<HTML>	Тип структури HTML	Початок структури HTML
<HEAD>	Початок опису документа	Розділ опису документа може включати мітки <TITLE>, <META>, <BASE> і <LINK>
<TITLE> ..... </TITLE>	Ім’я документа	Те, що буде вважатися заголовком (Назвою) документа
<META	Мета-інформація	Служить для вказівки: а) технічної інформації про документ б) інформації про зміст документа
HTTP-EQUIV="ім’я" CONTENT="значення"	Інформація для HTTP-сервера: привласнити будь-яке значення якомусь заголовку	Для використання кирилиці на HTML-сторінках вкажіть в заголовку: <META HTTPEQUIV = "ContentType "CONTENT =" txt / ht ml; charset = Windows-1251 ">

Продовження таблиці 1.1

NAME=" ім’я " CONTENT="значення">	Завдання мета-змінної і присвоєння їй значення	Завдання мета-інформації про документ. Пошукові служби судять по ній про утримання документа.
<LINK ..... >	Вказівки про гіперзв'язок даного документа	Атрибути, що вказано – такі ж, як у мітки <A> (Якір, anchor). Служить для вказівки перетинів, а також відносин між документами
</HEAD>	Кінець опису документа	
<BODY	Початок документа	Вказуються установки для показу документа
BACKGROUND="URL"	Фонова картинка	В лапках вказується URL картинки (.gif або .jpg)
BGCOLOR="#$$$$$$"	Колір фону	В лапках вказується номер кольору або його назва
TEXT="#$$$$$$"	Колір тексту	
LINK="#$$$$$$"	Колір посилання	
VLINK="#$$$$$$"	Колір посилання, що переглянуто	
ALINK="#$$$$$$" >	Колір активного посилання	

Продовження таблиці 1.1

</BODY>	Кінець документа	
</HTML>	Кінець структури HTML	

Завдання

Крім використання тегів <HTML>, <HEAD>, <TITLE> і <BODY>, які відповідно до стандартів HTML і WWW повинні бути присутніми в кожній WEB-сторінці, при організації тексту всередині WEB-документа за допомогою списків, а також використанні попереднього форматування і графіки застосовуються відповідні елементи мови HTML з певним набором тегів і необхідних атрибутів.

1. Використовуючи редактор тексту Блокнот (Notepad++) розробіть сторінку, використовуючи всі наведені в таблиці теги мови HTML.

Збережіть сторінку в робочому каталозі. Відформатуйте текст Вашого варіанту завдання згідно вимогам у дужках.

2. Перегляньте на свою сторінку за допомогою встановленого на комп'ютері браузера і добийтеся максимальної подібності в поданні сторінки браузером.

Для виконання даної роботи вам необхідно використовувати наступні теги елементів мови HTML (див. табл. 1.2).

Таблиця 1.2 – Основні теги в лабораторній роботі

Приклад елементу	Пояснення
<HTML>...</HTML>; <HEAD>...</HEAD>; <BODY>...</BODY>	Обов’язкові елементи для будь-якої сторінки HTML
<TITLE>...</TITLE>	Назва сторінки

Продовження таблиці 1.2

<H1>...</H1>; <H2>...</H2>; <H3>...</H3>; <H4>...</H4>; <H5>...</H5>; <H6>...</H6> атрибут align="left|right|center"	Стилі заголовків всередині сторінки
<P>...</P> атрибут align="left|right|center|justify"	Елемент нового абзацу
 	Тег перекладу рядка. Не поділяє два фрагмента порожнім рядком
<HR> атрибути: size, noshade, width, align, color	Тег горизонтальної лінії
<B>...</B>; <I>...</I>; <TT>...</TT>;<S>...</ S>; <U>...</U>;<BIG>...</BIG>; <SMALL>...</ SMALL>; _{...};^{...}	Оформлення тексту
<FONT>...</FONT> атрибути: size, face, color	Завдання розміру, типу шрифту і кольору тексту
<BODY text="Колір">...</BODY>	Спосіб управління кольором тексту
&-послідовності	Відображення спеціальних символів

1.2. Приклад виконання лабораторної роботи

1. Відкрийте текстовий редактор Notepad++:

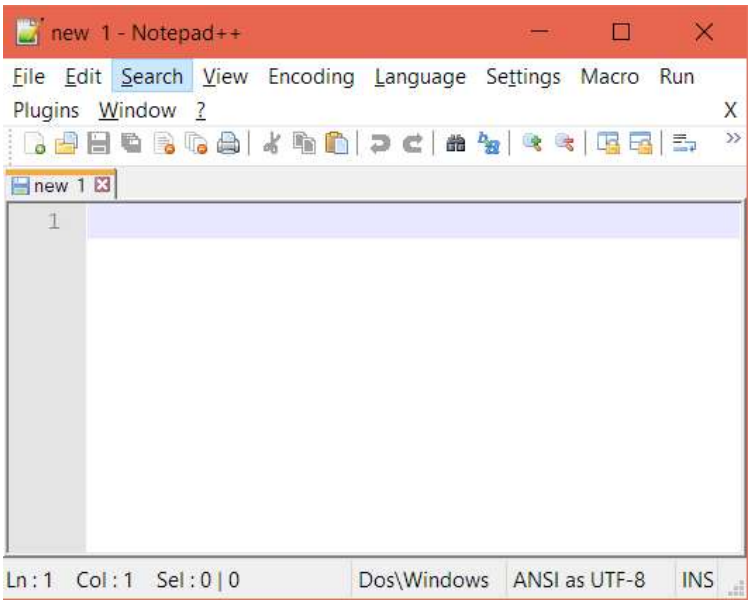

Рисунок 1.1 – Вікно текстового редактору Notepad++

2. Оберіть підсвітку синтаксису HTML:

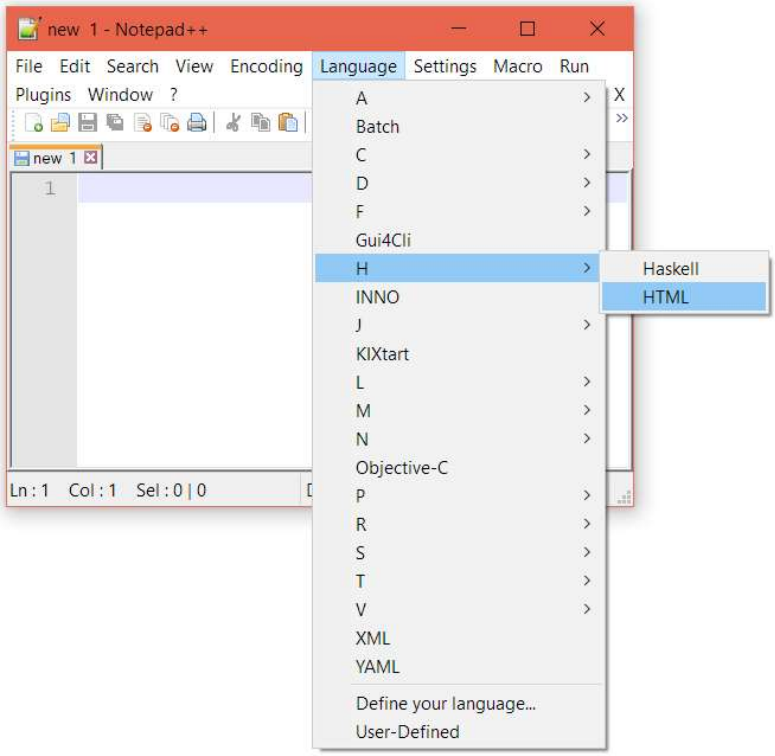

Рисунок 1.2 – Опція вибору підсвітки синтаксису

3. Скопіюйте HTML-розмітку:

```
<!DOCTYPE html>
<html>
<head>
    <title>WEB-технології</title>
</head>
<body text="#1772c2" BGCOLOR="#EEEEEE">

    <h1>Лабораторна робота №1</h1>
    <h2>Структура HTML сторінки</h2>
    <ol>
        <li>Починається з тегу <font color="green">&lt;html&gt;</font> і закінчується відповідним йому тегом - <font color="green">&lt;/html&gt;</font>.</li>

        <li>Містить два розділи, розміщені у строго визначеному порядку: "голова" (заголовок) і "тіло" (зміст). Заголовок HTML-документу помічається тегами: <strong>&lt;head&gt;</strong>...<strong>&lt;/head&gt;</strong> - та містить відомості про документ загалом. Зокрема, він повинен містити теги <u>&lt;title&gt;...&lt;/title&gt;</u>, між якими розміщують текст, що повинен відображатися у заголовку вікна браузера. Крім цього, у розділі заголовків може міститися тег <i>&lt;meta&gt;</i>, призначений для технічного опису документа (ці відомості для пошукових програм), а також тег <span style="color:red;">&lt;style&gt;</span> для опису стилів (наборів параметрів форматування), використаних у документі.</li>

        <li>Безпосередньо зміст (контент) HTML-документу міститься в "тілі", що розташоване між тегами &lt;body&gt;...&lt;/body&gt;.</li>
    </ol>
    <p>Отже, в основній структурі HTML-документу обов'язковими є чотири парні теги, які записуються у строго визначеній послідовності. Іформація взята <a href="https://sites.google.com/site/osnovihtml/algoritm/struktura-html-dokumentu" target="_blank">тут</a>.</p>
    <h2>Теги HTML</h2>
    <p>З основними HTML-тегами можна ознайомитися за <a href="https://css.in.ua/html/tags" target="_blank">посиланням</a>.</p>
    <table border="1">
        <caption>Таблиця з основними тегами в лабораторній роботі</caption>
        <thead bgcolor="#ccffcc">
            <tr>
                <th>Приклад елементу</th>
                <th>Пояснення</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>
                    &lt;HTML&gt;...&lt;/HTML&gt;;
                    &lt;HEAD&gt;...&lt;/HEAD&gt;;
                    &lt;BODY&gt;...&lt;/BODY&gt;
                </td>
                <td>Обов’язкові елементи для будь-якої сторінки HTML</td>
            </tr>
            <tr>
                <td>&lt;TITLE&gt;...&lt;/TITLE&gt;</td>
                <td>Назва сторінки</td>
            </tr>
            <tr>
                <td>
                    &lt;H1&gt;...&lt;/H1&gt;; &lt;H2&gt;...&lt;/H2&gt;;
                    &lt;H3&gt;...&lt;/H3&gt;; &lt;H4&gt;...&lt;/H4&gt;;
                    &lt;H5&gt;...&lt;/H5&gt;; &lt;H6&gt;...&lt;/H6&gt;
                    атрибут align="left|right|center"
                </td>
                <td>Стилі заголовків всередині сторінки</td>
            </tr>
            <tr>
                <td>
```

```
                    &lt;P&gt;...&lt;/P&gt;
                    атрибут align="left|right|center|justify"
                </td>
                <td>Елемент нового абзацу</td>
            </tr>
            <tr>
                <td>&lt;BR&gt;</td>
                <td>Тег перекладу рядка. Не поділяє два фрагмента порожнім рядком</td>
            </tr>
            <tr>
                <td>
                    &lt;HR&gt;
                    атрибути: size, noshade, width, align, color
                </td>
                <td>Тег горизонтальної лінії</td>
            </tr>
            <tr>
                <td>
                    &lt;B&gt;...&lt;/B&gt;; &lt;I&gt;...&lt;/I&gt;;
                    &lt;TT&gt;...&lt;/TT&gt;;&lt;S&gt;...&lt;/ S&gt;;
                    &lt;U&gt;...&lt;/U&gt;;&lt;BIG&gt;...&lt;/BIG&gt;;
&lt;SMALL&gt;...&lt;/ SMALL&gt;;
                    &lt;SUB&gt;...&lt;/SUB&gt;;&lt;SUP&gt;...&lt;/SUP&gt;
                </td>
                <td>Оформлення тексту</td>
            </tr>
            <tr>
                <td>
                    &lt;FONT&gt;...&lt;/FONT&gt;
                    атрибути: size, face, color
                </td>
                <td>Завдання розміру, типу шрифту і кольору тексту</td>
            </tr>
            <tr>
                <td>&lt;BODY text="Колір"&gt;...&lt;/BODY&gt;</td>
                <td>Спосіб управління кольором тексту</td>
            </tr>
            <tr>
                <td>&amp;-послідовності</td>
                <td>Відображення спеціальних символів</td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```

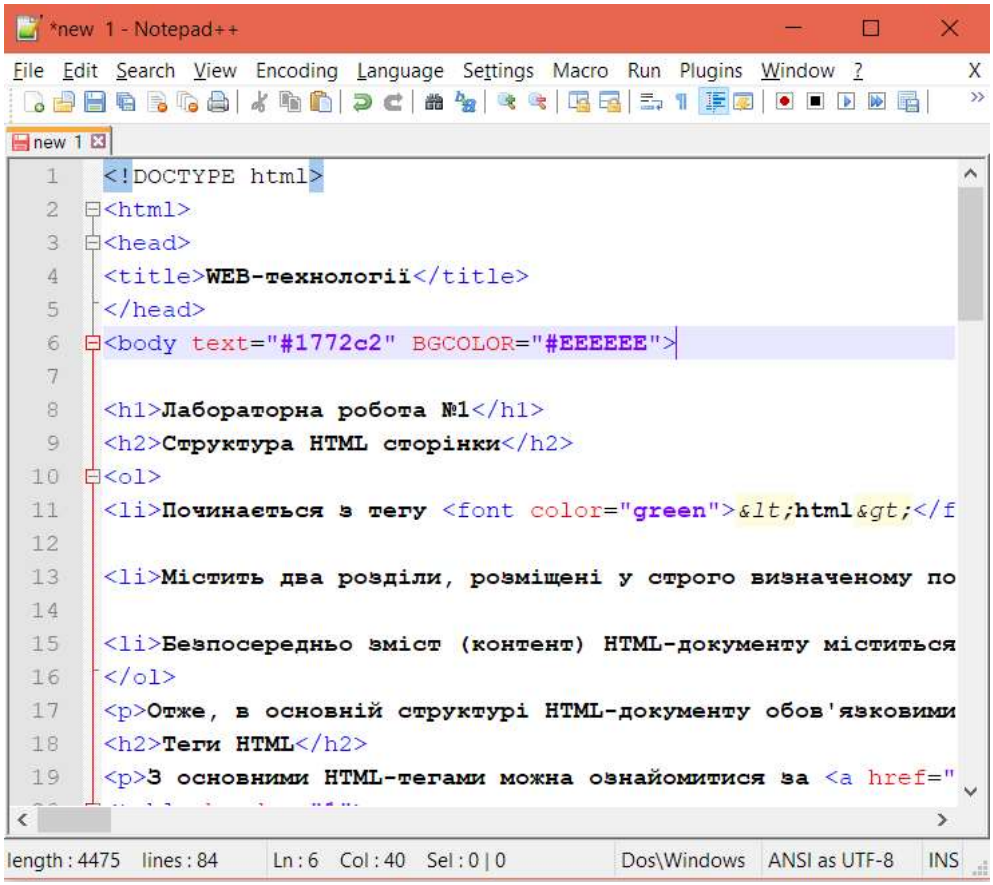

Рисунок 1.3 – Вигляд HTML-документу в редакторі Notepad++

4. Збережіть файл у форматі *.html:

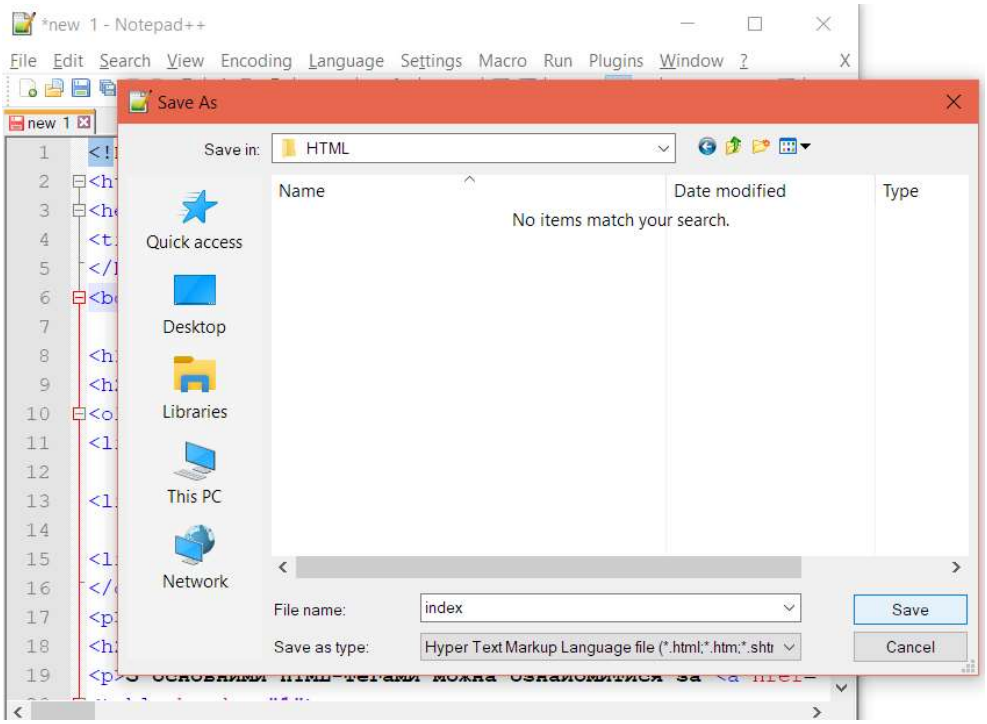

Рисунок 1.4 – Збереження файлу у форматі web-сторінки (.html)

5. Відкрийте збережений файл у Web-браузері:

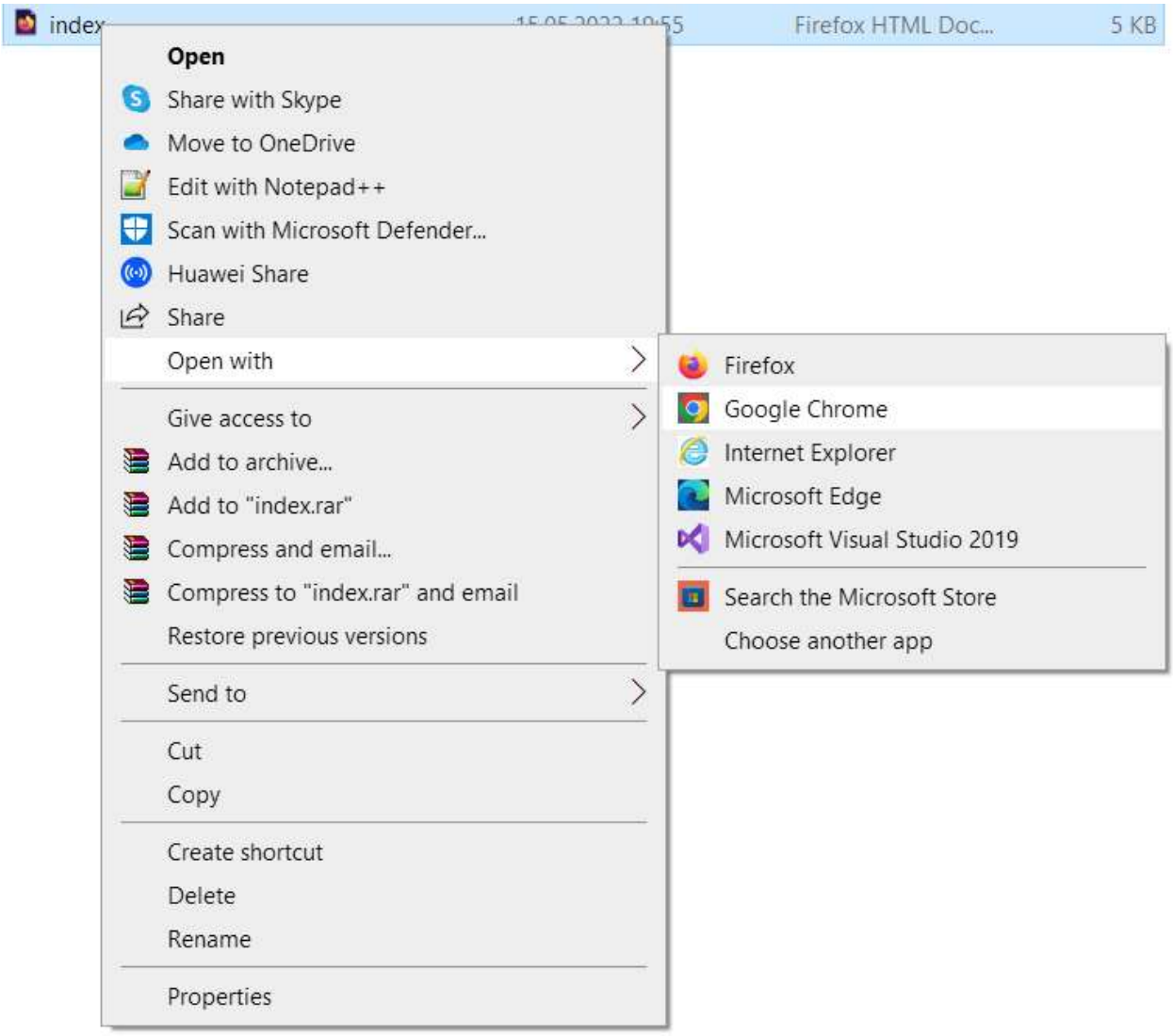

Рисунок 1.5 – Відкриття html-файлу у WEB-браузері

6. Результат має відображатися наступним чином:

Лабораторна робота №1

Структура HTML сторінки

1. Починається з тегу <html> і закінчується відповідним йому тегом - </html>.
2. Містить два розділи, розміщені у строго визначеному порядку: "голова" (заголовок) і "тіло" (зміст). Заголовок HTML-документу помічається тегами: **<head>...</head>** - та містить відомості про документ загалом. Зокрема, він повинен містити теги <title>...</title>, між якими розміщують текст, що повинен відображатися у заголовку вікна браузера. Крім цього, у розділі заголовків може міститися тег <*meta*>, призначений для технічного опису документа (ці відомості для пошукових програм), а також тег <style> для опису стилів (наборів параметрів форматування), використаних у документі.
3. Безпосередньо зміст (контент) HTML-документу міститься в "тілі", що розташоване між тегами <body>...</body>.

Отже, в основній структурі HTML-документу обов'язковими є чотири парні теги, які записуються у строго визначеній послідовності. Іформація взята тут.

Теги HTML

З основними HTML-тегами можна ознайомитися за посиланням.

Таблиця з основними тегами в лабораторній роботі

Приклад елементу	Пояснення
<HTML>...</HTML>; <HEAD>...</HEAD>; <BODY>...</BODY>	Обов'язкові елементи для будь-якої сторінки HTML
<TITLE>...</TITLE>	Назва сторінки
<H1>...</H1>; <H2>...</H2>; <H3>...</H3>; <H4>...</H4>; <H5>...</H5>; <H6>...</H6> атрибут align="left|right|center"	Стилі заголовків всередині сторінки
<P>...</P> атрибут align="left|right|center|justify"	Елемент нового абзацу
	Тег перекладу рядка. Не

Рисунок 1.6 – Вигляд html-документу у WEB-браузері

1.3. Питання для самоконтролю

1. Як називається стандартна мова розмітки документів у всесвітній павутині, яка обробляється спеціальними програмами і відображається у вигляді документа, у зручній для людини формі називається?
2. Яку назву мають команди розмітки HTML?
3. Що означає "index.html"?
4. Який тег задає колір тла документа?
5. Запишіть структуру HTML сторінки.
6. Назвіть тег, який дозволяє створити нумерований список.
7. Який тег визначає найважливіший заголовок?
8. У який тег заключається основне наповнення веб-сторінки?
9. Що означає контейнерний тег - ?
10. У якому форматі або з яким розширенням слід зберегти текстовий файл, щоб він став веб-документом?
11. Що означає тег
?
12. Що означає тег ?
13. Наведіть розмітку таблиці з підписом і шапкою.
14. Що означає атрибут target=”_blank” тегу <a>…?

2. Застосування CSS для оформлення інтерфейсу web-додатку

Мета роботи: ввести поняття каскадних таблиць стилів, використати їх переваги.

2.1. Теоретичні відомості

У зв'язку з тим, що HTML не мав можливості управління зовнішнім виглядом Веб-сторінок, щоб вирішити ці проблеми, консорціумом W3C була розроблена технологія Cascading Style Sheets або CSS (Каскадні таблиці стилів) - технологія опису зовнішнього вигляду документа, написана мовою розмітки. В CSS надається можливість призначити усім об'єктам стиль, опис якого може зберігатися в самому HTML-файлі, або в окремому файлі.

ПРАВИЛА В CSS. CSS надає можливість створювати правила, які легко змінювати, редагувати і застосовувати до усіх визначених нами елементів. Кожне правило, складається з двох частин. У лівій частині міститься *селектор (selector),* а у правій - *блок визначення (declaration block).* Блок визначення складається з набору *властивостей (property)* та їх *значень (value)*.

Селектор – елемент, який визначає правило.

Властивість описує елемент, що вводиться.

Значення визначають природу властивостей.

Приклад: **h1** *(селектор)* **{color: red}** *(блок-значення)*

ДОДАВАННЯ ТАБЛИЦЬ СТИЛІВ В HTML-ДОКУМЕНТ. Існують **три способи** додавання правил CSS в HTML-документи:

- ***<u>СПОСІБ ПЕРШИЙ</u>: ДОДАВАННЯ CSS В HTML-ТЕГ.***

У цьому способі CSS додається в HTML-документ за допомогою HTML-атрибуту style у середині будь-якого HTML-тегу, що знаходиться у контейнері <body>.

```
<html>
  <head>
    <title>Sample BG color</title>
  </head>
  <body style="background-color: #FF0000;">
    <p>You see red color</p>
  </body>
</html>
```

Цей спосіб використовується у тому разі коли окремому елементу потрібно надати декілька стилів не використовуючи вбудовані або зовнішні стилі.

Застосування цього способу несе за собою певні недоліки:

- збільшується об’єм файлу, що приводить до збільшення часу завантаження веб-сторінки;

- ускладнює редагування документів.

- ***СПОСІБ ДРУГИЙ: ВСТАНОВЛЕННЯ СТИЛЮ ДЛЯ ТЕГІВ HEAD БЛОЦІ***

CSS додається в HTML-документ за допомогою HTML-тегу <style> в середині контейнеру <head>. В ньому описуються всі стилі, що будуть використані.

```
<html>
  <head>
    <title>Sample head</title>
    <style type="text/css">
      body {background-color: #FF0000;}
```

```
    </style>
    <p>You see red color</p>
  </body>
</html>
```

- ***<u>СПОСІБ ТРЕТІЙ</u>: ПОСИЛАННЯ НА ТАБЛИЦЮ СТИЛІВ***

Зовнішня таблиця стилів являє собою звичайний текстовий файл з розширенням css.

Для того щоб зробити посилання на зовнішній файл із HTML-документа (index.htm) на файл таблиці стилів (style.css) треба у контейнері <head>вставити наступну стрічку:

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

Це посилання указує браузеру, що він повинен використовувати правила відображення HTML-файлу з CSS-файлу.

```
<html>
  <head>
    <title>Sample of CSS link</title>
  <link rel="stylesheet" type="text/css" href="style/style.css" />
  </head>
  <body>
  </body>
</html>
```

Найважливішим тут є те, що один CSS-файл можна використовувати для управління відображення багатьох HTML-документів.

Управління шрифтом. В старих версіях HTML шрифт оформлявся за допомогою тегу <font> але цей тег треба було додавати кожного разу, коли було необхідно встановити шрифт, від цього збільшувався розмір веб-сторінки, незручно було змінювати властивість шрифту.

Щоб задати сімейство шрифтів використовується властивість font-family. В цій властивості завжди вказується ряд шрифтів, розділених комою, наприкінці списку вказується сімейство шрифтів. При застосуванні шрифтів до веб-сторінки завжди задається основний шрифт, а потім альтернативний. У разі відсутності на комп'ютері користувача заданих шрифтів то веб-сторінка, як мінімум буде відображена шрифтом, що входить до цього сімейства.

Таблиця 2.1 – Основні п'ять сімейств шрифтів

Шрифт	**Сімейство шрифтів**
Times New Roman Georgia	serif(з засічками)
Arial Candara	sans-serif (без засічок)
Courier Courier New	monospace (моноширинний)
Bradley Hand ITC Edwardian Script ITC	cursive (рукописний)
Luminari, fantasy	fantasy (декоративний)

Шрифти сімейства serif найкраще підходять для основного тексту сторінки. Засічки допомагають направляти увагу читача уздовж рядка

Шрифти сімейства sans-serif використовуються для оформлення заголовків, панелей посилань та посилань. Шрифти без засічок звертають на себе більше уваги, але погано підходять для довгого читання.

У моноширинних шрифтах усі символи мають однакову ширину. Слід зазначити, що у моноширинних шрифтах збільшується між символьний інтервал, як зліва так і з права символу (цей інтервал є невід'ємною частиною символу). Моноширинні шрифти допускаються тільки для створення якихось особливих ефектів оформлення — наприклад, у даній роботі моноширинним шрифтом набрано фрагменти коду HTML та CSS. У моноширинному шрифті всякий символ має одну і ту ж ширину.

У файлі style.css створимо для оформлення абзацу таке правило:

```
p{font-family: Arial, Candara, Century-Gothic, sans-serif}
```

```
<html>
  <head>
  <title>Sample Sans</title>
  <link rel="stylesheet" type="text/css" href="style/style.css" />
  </head>
  <body>
    <p>Sans-serif text font</p>
  </body>
</html>
```

У цьому прикладі основним шрифтом абзацу є шрифт Arial у разі його відсутності буде загружено наступний, у разі відсутності усіх шрифтів абзац буде оформлено шрифтом сімейства sans-serif.

Властивість font-style визначає стиль шрифту з обраного сімейства може мати наступні значення:

- normal – звичайний шрифт;
- italic – курсивний шрифт (більш декоративний шрифт з нахилом в право);
- oblique – нахилений шрифт (звичайний шрифт нахилений в право).

У файлі style.css створимо таке правило:

```
h1{
   font-family: Arial, Candara, Century-Gothic, sans-serif;
   font-style: normal;
}
h2{
   font-family: Times New Roman, Georgia, serif;
   font-style: italic;
}
```

```
p{
    font-family: Times New Roman, Georgia, serif;
    font-style: oblique;
}

<html>
  <head>
  <title>Tag css sample</title>
  <link rel="stylesheet" type="text/css" href="style/style.css" />
  </head>
  <body>
  <h1>Шрифт цього заголовку звичайний</h1>
  <h2>Шрифт цього заголовку курсив</h2>
    <p> Шрифт цього абзацу нахилено вправо</p>
  </body>
</html>
```

Псевдокласи. Властивості гіперпосилання засобами CSS можна визначати по-різному, залежно від того, відвідали вже посилання, чи активне воно, чи знаходиться покажчик миші над посиланням. Це дозволяє додати цікаві ефекти на ваш веб-сайт. Щоб оформити гіперпосилання засобами CSS, треба використовувати так звані псевдокласи.

Псевдоклас дозволяє враховувати різні стани або події при визначенні властивостей html-тега. У гіперпосилання є декілька станів та подій.

Псевдоклас : link використовується для посилань на сторінки, які користувач ще не відвідував.

```
a: link{
color: #0000ff;
font-weight: normal;
}
```

Псевдоклас: active використовується для активних посилань.

```
a: active {
color: #00bfff;
```

```
background-color:ffd700;
{
```

У цьому прикладі у посилання буде змінено колір шрифту та колір фону.

Псевдоклас: visited використовується для посилань на сторінки, які відвідав користувач.

```
a: visited {
color: #0000ff;
font-weight: 400;
}
```

Псевдоклас: hover використовується для посилань, над котрими находиться вказівник миші.

```
a: hover{
font-weight: normal;
text-trasform: uppercase;
letter-spacing: 10px;
}
```

У цьому прикладі при шрифт посилання буде нормальної товщини, текст буде відображатися прописними літерами, відстань між літерами буде складати 10px.

Видалення підкреслювання посилань. Видалити підкреслювання посилань дуже просто. Для видалення підкреслювання достатньо встановити властивість text-decoration зі значенням none.

```
a: link{
color: #0000ff;
font-weight: normal;
text-decoration: none;
}
```

Ідентифікація і групування елементів. Атрибут class вказує, що елемент є членом певного класу. Для прикладу візьмемо документ в якому є два списки посилань, але треба щоб кольори посилань цих списків відрізнялися.

```
<p>Список №1:</p>
<ul>
<li><a href="link1_1.htm">Посилання 1,1</a></li>
<li><a href="link1_2.htm">Посилання 1,2</a></li>
<li><a href="link1_3.htm">Посилання 1,3</a></li>
</ul>
<p>Список №2:</p>
<ul>
<li><a href="link2_1.htm">Посилання 2,1</a></li>
<li><a href="link2_2.htm">Посилання 2,2</a></li>
<li><a href="link2_3.htm">Посилання 2,3</a></li>
</ul>
<a href="index.htm">На головну</a>
```

Щоб досягти нашої мети розділимо посилання на дві категорії за допомогою привласнення класу кожному посиланню атрибутом class.

У файлі з html-кодом додамо class="classname" в тег до якого ми будемо застосовувати клас.

```
<p>Список №1:</p>
<ul>
<li><a href="link1_1.htm" class="list1">Посилання 1,1</a></li>
<li><a href="link1_2.htm" class="list1">Посилання 1,2</a></li>
<li><a href="link1_3.htm" class="list1">Посилання 1,3</a></li>
</ul>
<p>Список №2:</p>
<ul>
<li><a href="link2_1.htm" class="list2">Посилання 2,1</a></li>
<li><a href="link2_2.htm" class="list2">Посилання 2,2</a></li>
<li><a href="link2_3.htm" class="list2">Посилання 2,3</a></li>
</ul>
<a href="index.htm">На головну</a>
```

У файлі style.css опишемо властивості класів для посилань. Синтаксис опису класу має наступний вигляд:

[selector].[className] {property: value}

```
a {
color: red;
}
a.list1 {
color: #FFCC00;
}
a.list2 {
color: #700000;
}
```

Окрім групування елементів вам може знадобитися ідентифікувати один унікальний елемент. Це можна реалізувати за допомогою атрибуту id.

Особливість id в тому, що в документі не може бути більш за один елемент з даним конкретним id. Кожен id має бути унікальним. У інших випадках треба використовувати атрибут class.

Завдання

1. Створіть новий HTML документ, що містить основний заголовок H1, Таблицю із двох колонок, в лівій — ненумерований список, в правій заголовок H2 та абзац з текстом (рис. 1).

2. Створити набір CSS стилів, розташувавги його у HEAD, оформити всі елементи сторінки на свій смак. Лівий список зробити посиланнями, що реагують на наведення миші та змінюють колір (застосувати :hover)

3. Перенесить створені стилі у окрейми файл css/main.css та підключіть його за допомогою <link> до документу.

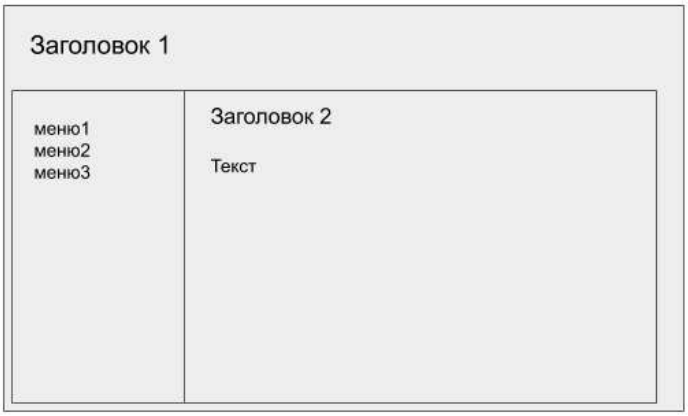

Рис. 2.1. Шаблон HTML-документу

4. Створіть аналогічний документ з іншим заголовок та текстом, підключіть до нього той самий файл стилів, та впевніться в його роботі.

5. Зробіть висновки щодо зручності використання CSS.

2.2. Приклад виконання лабораторної роботи

1. Створіть html-документ з наступною розміткою:

```
<!DOCTYPE html>
<html>
<head>
    <title>Parfums</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/style.css" media="screen" type="text/css" />
</head>
<body>
    <div id="site">
        <header id="masthead">
            <h1>Parfums <span class="tagline">Parfums for web developers since 1999</h1>
        </header>
        <div id="content">
            <div id="products">
                <ul>
                    <li>
                        <div class="product-image">
                            <img src="images/wine1.jpg" alt="" />
                        </div>
                        <div class="product-description">
                            <h3 class="product-name">Parfume #1</h3>
                            <p class="product-price">&euro; 5</p>
                            <form class="add-to-cart" action="cart.html" method="post">
                                <div>
                                    <label for="qty-1">Quantity</label>
                                    <input type="text" name="qty-1" id="qty-1"
class="qty" value="1" />
                                </div>
                                <p><input type="submit" value="Add to cart" class="btn"
/></p>
                            </form>
                        </div>
                    </li>
                    <li>
                        <div class="product-image">
```

```
                        <img src="images/wine2.jpg" alt="" />
                    </div>
                    <div class="product-description">
                        <h3 class="product-name">Parfume #2</h3>
                        <p class="product-price">&euro; 8</p>
                        <form class="add-to-cart" action="cart.html" method="post">
                            <div>
                                <label for="qty-2">Quantity</label>
                                <input type="text" name="qty-2" id="qty-2"
class="qty" value="1" />
                            </div>
                            <p><input type="submit" value="Add to cart" class="btn"
/></p>
                        </form>
                    </div>
                </li>
                <li>
                    <div class="product-image">
                        <img src="images/wine3.jpg" alt="" />
                    </div>
                    <div class="product-description">
                        <h3 class="product-name">Parfume #3</h3>
                        <p class="product-price">&euro; 11</p>
                        <form class="add-to-cart" action="cart.html" method="post">
                            <div>
                                <label for="qty-3">Quantity</label>
                                <input type="text" name="qty-3" id="qty-3"
class="qty" value="1" />
                            </div>
                            <p><input type="submit" value="Add to cart" class="btn"
/></p>
                        </form>
                    </div>
                </li>
            </ul>
        </div>
    </div>
</div>
<footer id="site-info">
    Copyright &copy; Parfums
</footer>
</body>
</html>
```

2. В папці з html-документом створіть папку з назвою "css".

3. Відкрийте текстовий редактор Notepad++ і скопіюйте в нього код стилізації html-документу:

```
@import
url(http://fonts.googleapis.com/css?family=PT+Serif:400,700,400italic);
/*!    normalize.css    v2.1.2    |    MIT    License    |    git.io/normalize
*/article,aside,details,figcaption,figure,footer,header,hgroup,main,nav,secti
on,summary{display:block}audio,canvas,video{display:inline-
block}audio:not([controls]){display:none;height:0}[hidden],template{display:n
one}script{display:none!important}html{font-family:sans-serif;-ms-text-size-
adjust:100%;-webkit-text-size-adjust:100%}body{margin:0}a{background:0
0}a:focus{outline:thin                dotted}a:active,a:hover{outline:0}h1{font-
```

```
size:2em;margin:.67em 0}abbr[title]{border-bottom:1px dotted}b,strong{font-
weight:700}dfn{font-style:italic}hr{-moz-box-sizing:content-box;box-
sizing:content-
box;height:0}mark{background:#ff0;color:#000}code,kbd,pre,samp{font-
family:monospace,serif;font-size:1em}pre{white-space:pre-
wrap}q{quotes:"\201C" "\201D" "\2018" "\2019"}small{font-
size:80%}sub,sup{font-size:75%;line-height:0;position:relative;vertical-
align:baseline}sup{top:-.5em}sub{bottom:-
.25em}img{border:0}svg:not(:root){overflow:hidden}figure{margin:0}fieldset{bo
rder:1px solid silver;margin:0 2px;padding:.35em .625em
.75em}legend{border:0;padding:0}button,input,select,textarea{font-
family:inherit;font-size:100%;margin:0}button,input{line-
height:normal}button,select{text-transform:none}button,html
input[type=button],input[type=reset],input[type=submit]{-webkit-
appearance:button;cursor:pointer}button[disabled],html
input[disabled]{cursor:default}input[type=checkbox],input[type=radio]{box-
sizing:border-box;padding:0}input[type=search]{-webkit-appearance:textfield;-
moz-box-sizing:content-box;-webkit-box-sizing:content-box;box-sizing:content-
box}input[type=search]::-webkit-search-cancel-button,input[type=search]::-
webkit-search-decoration{-webkit-appearance:none}button::-moz-focus-
inner,input::-moz-focus-
inner{border:0;padding:0}textarea{overflow:auto;vertical-
align:top}table{border-collapse:collapse;border-spacing:0}
```

```
html, body {
    height: 100%;
    min-height: 100%;
}
body {
    font: 100%/1 'PT Serif', serif;
    background: #fff;
    color: #333;
    border-top: 0.5em solid #cc8500;
}
:active, :focus {
    outline-style: none;
}
a:link {
    color: #cc8500;
    text-decoration: none;
}
a:hover {
    color: #cc1400;
    text-decoration: underline;
}
h1, h2, h3, h4, h5, h6 {
    font-weight: normal;
    font-size: 100%;
}
#site {
    max-width: 960px;
    margin: 0 auto;
    padding: 0 1em;
}
```

```
#masthead {
    margin-top: 3em;
    padding: 2em 0 0.5em 0;
    border-bottom: 1px solid #ddd;
}
#masthead h1 {
    margin: 0;
    font-size: 3.5em;
    overflow: hidden;
    padding-left: 85px;
    min-height: 82px;
    line-height: 2.3;
    background: url(../images/logo.png) no-repeat 0 50%;
    color: #900;
}
.tagline {
    float: right;
    color: #666;
    font-size: 14px;
    padding-top: 3.5em;
}
#products ul {
    margin: 1.5em 0;
    padding: 0;
    list-style: none;
    overflow: hidden;
}
#products li {
    float: left;
    width: 31%;
    margin: 0 1%;
    display: block;
}
.product-image {
    margin: 0 auto 1em auto;
    background: #000;
    width: 150px;
    height: 150px;
    overflow: hidden;
    border-radius: 50%;
}
.product-description {
    padding: 1em;
    background: #000;
    color: #fff;
    border-radius: 5px;
}
.product-name {
    text-align: center;
    color: #fc0;
    margin: 0;
    font-size: 1.4em;
    padding-bottom: 0.2em;
    border-bottom: 1px dotted #666;
    text-transform: uppercase;
    letter-spacing: 0.1em;
}
.product-price {
    width: 4em;
    height: 4em;
    font-size: 1.2em;
    text-align: center;
    margin: 1em auto;
    background: #fff;
    color: #800;
```

```
    line-height: 4;
    border-radius: 50%;
}
form.add-to-cart div,
form.add-to-cart p {
    text-align: center;
}
form input.qty {
    width: 40px;
    border: 1px solid #eee;
    font: 1em 'PT Serif', serif;
    background: #f9f9f9;
    color: #000;
    border-radius: 3px;
    margin-left: 0.4em;
}
.btn, a.btn {
    display: inline-block;
    background: #cc1400;
    color: #fff;
    font: 1em 'PT Serif', serif;
    padding: 0.3em 1em;
    text-align: center;
    border-radius: 4px;
    border: 1px solid #a00;
}
#site-info {
    height: 3em;
    width: 100%;
    line-height: 3;
    text-align: center;
    background: #cc8500;
    position: absolute;
    color: #fff;
    left: 0;
    bottom: 0;
}
body#checkout-page #site-info {
    position: static;
}
#shopping-cart {
    margin: 1.5em 0;
}
.shopping-cart {
    border: 1px solid #ddd;
    border-collapse: collapse;
    border-spacing: 0;
    width: 100%;
    table-layout: fixed;
}
.shopping-cart th {
    font-size: 1.3em;
    padding: 0.3em;
    width: 33.3%;
    border: 1px solid #ddd;
    text-transform: uppercase;
}
.shopping-cart td {
    padding: 0.3em;
    width: 33.3%;
    border: 1px solid #ddd;
}
.shopping-cart tr:nth-child(even) {
    background: #fafafa;
}
```

```
,
.shopping-cart td.pdelete {
    text-align: center;
}
.pdelete a,
.pdelete a:hover {
    color: #c00;
    text-decoration: none;
    font-size: 2.5em;
    display: block;
    text-align: center;
}
#shopping-cart-actions {
    margin: 1.5em 0;
    padding: 0;
    list-style: none;
    text-align: center;
}
#shopping-cart-actions li {
    display: inline-block;
    margin-right: 1em;
}
#pricing {
    padding: 0.5em;
    margin: 1em 0;
    background: #fafafa;
}
#sub-total, #shipping {
    margin: 1.5em 0;
    text-align: right;
}
#sub-total span,
#shipping span {
    margin-left: 1em;
}
#content > h1,
#checkout-order-form h2 {
    font-size: 2em;
    text-align: center;
}
#pricing #sub-total,
#pricing #shipping {
    margin: 1em 0;
}
#checkout-order-form {
    margin: 1.5em 0;
}
#checkout-order-form fieldset {
    border: 1px solid #ddd;
    border-radius: 6px;
    padding: 1em;
    margin-bottom: 1.3em;
}
#checkout-order-form legend {
    padding: 0.3em;
    background: #fafafa;
    font-weight: bold;
}
#checkout-order-form div {
    margin-bottom: 1em;
}
#checkout-order-form label {
    display: block;
    font-weight: bold;
    margin-bottom: 0.3em;
```

```
    text-align: left;
}
#checkout-order-form input[type="text"] {
    width: 200px;
    display: block;
    background: #fff;
    border: 1px solid #ddd;
    border-radius: 4px;
    padding: 0.3em;
}
#checkout-order-form select {
    width: 200px;
    display: block;
}
.message {
    display: block;
    margin: 0.5em 0;
    color: red;
}
#user-details {
    margin: 1.5em 0;
}
#user-details > h2 {
    text-align: center;
    font-size: 2em;
}
#user-details-content {
    margin: 1.5em 0;
    padding: 1em;
    border: 1px solid #ddd;
    border-radius: 6px;
    overflow: hidden;
}
#user-details-content .detail {
    float: left;
    width: 46%;
}
#user-details-content .detail.right {
    float: right;
}
#user-details-content .detail > h2 {
    text-align: center;
    font-size: 1.6em;
    margin-bottom: 0.5em;
}
#user-details-content ul {
    margin: 0 0 1em 0;
    padding: 0;
    list-style: none;
}
#user-details-content li {
    display: block;
    margin-bottom: 0.5em;
    padding-bottom: 0.3em;
    border-bottom: 1px solid #ddd;
}#paypal-form {
    margin: 1.5em 0;
}
```

4. Збережіть цей файл з назвою *style* у форматі *.css в папці *css*:

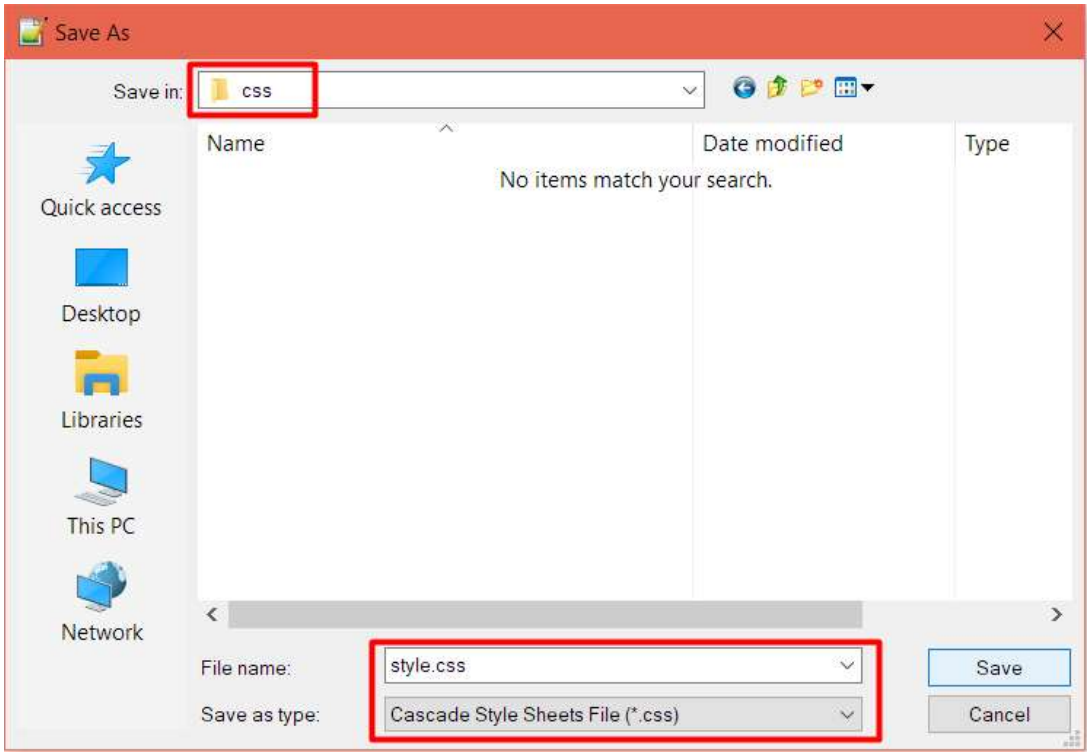

Рисунок 2.2 – Збереження файлу в форматі *.css

5. Можна відкрити стилізований html-документ:

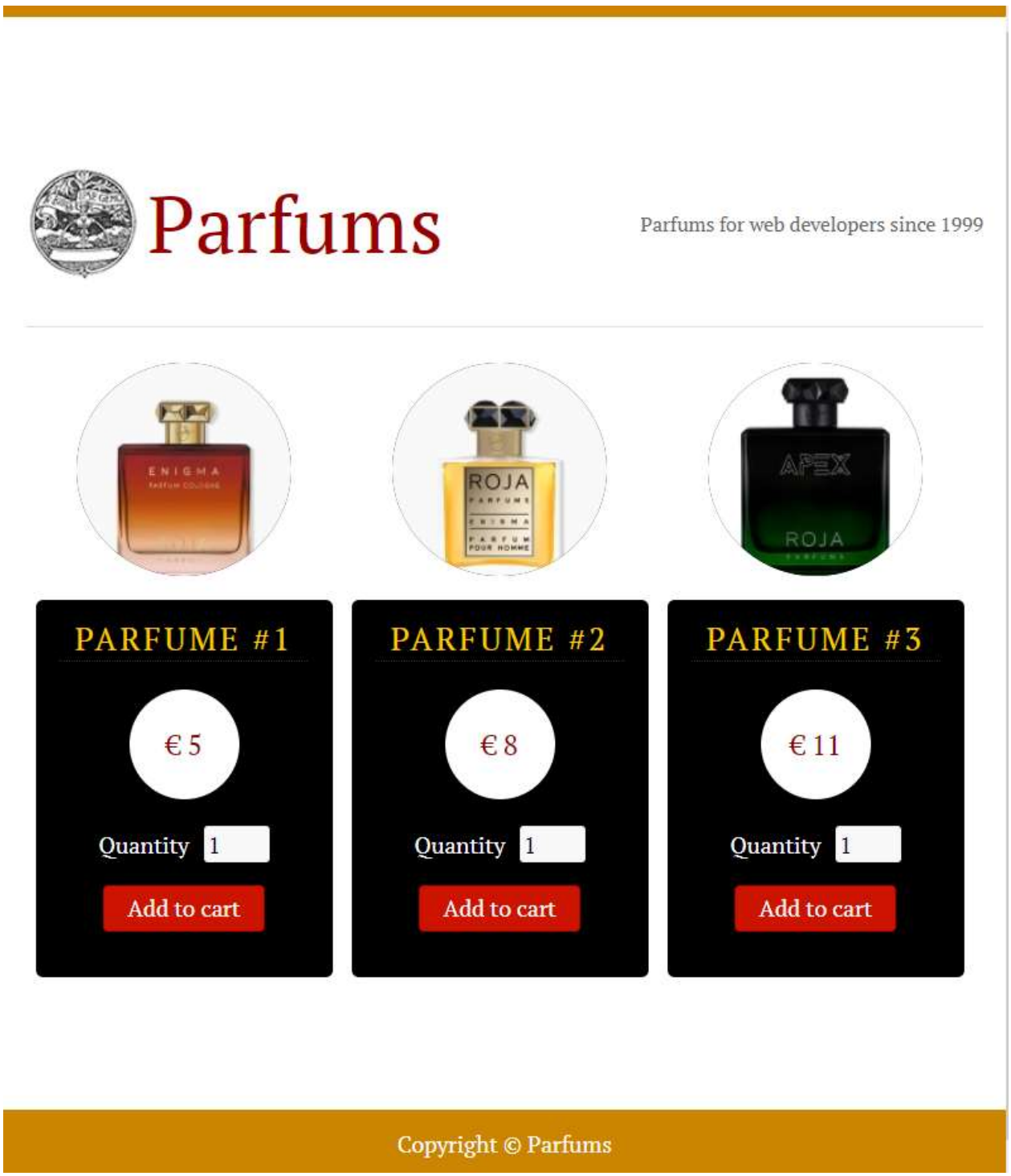

Рисунок 2.3 – Вигляд html-документу у WEB-браузері

2.3. Питання для самоконтролю

1. Що таке CSS?
2. Як під'єднати зовнішні стилі до сторінки?
3. В якому із тегів html- документу прописується таблиця стилів, яка буде використовуватися?
4. Який тип (формат) файлу стилів?
5. Назвіть переваги використання зовнішньої таблиці стилів?
6. Що таке селектор в CSS-правилі?
7. Які лапки використовують в оголошенні CSS правила?
8. Який знак розділяє CSS властивість та значення?
9. Як звернутися до CSS селектора по класу у файлі стилів?
10. Для яких CSS властивостей підійде значення cursive; ?
11. Для яких CSS властивостей підійде значення #fff; ?
12. В яких одиницях можна задати блокового тегу?
13. Що означає запис *margin: 50px auto;*?
14. До якої CSS властивості підійде значення - 1px solid red;
15. За допомогою якої властивості задається скруглення кутів блоку?
16. Як зробити так щоб текст обтікав картинку справа. Що потрібно прописати для картинки?
17. Як розмістити кнопку "На гору" у правій нижній частині екрану?
18. Що потрібно прописати батьківському елементу, щоб дочірні елементи позиціюватися відносно нього?
19. Що потрібно прописати елементу, щоб позиціювати його відносно екрана браузера?
20. До яких елементів можна застосувати z-index?
21. При розробці макету з двох колонок, що слід прописати боковій колонці, щоб вона була справа?

3. Застосування Javascript для динаміки інтерфейсу та валідації форм на сторінці

Мета роботи: ввести поняття сценарію, теги інтерактивності

3.1. Теоретичні відомості

Сценарії

Мова HTML служить для опису структури тексту, мова CSS - для опису того, як ця структура повинна відображатися браузером. Ці дві мови не передбачають можливості обміну інформацією з користувачем. Однак часто це буває необхідно.

Наприклад, при створенні навчального сайту є потреба з обміном інформацією веб-сторінкою, що реалізує тестування.

Існує мова, звана мовою сценаріїв, яка дозволяє реалізовувати найпростіші дії над даними, які вводить користувач, а також над властивостями елементів веб-сторінки. У нашому випадку ця мова називається JScript.

Модель веб-сторінки, в якій всі елементи утворюють ієрархію об'єктів, де на вершині знаходиться об'єкт window, називається Document Object Model (DOM). У кожного об'єкта є складові, до яких можна звертатися за допомогою символу "точка" (Перш за все це властивості):

об'єкт.властивість

Приклад: звернення до властивості в команді присвоювання

window.status = 'Це рядок стану'

Ця команда ("записати в рядок стану вікна даний текст") могла б працювати в будь-якій певній ситуації, наприклад, при наведенні покажчика миші на той чи інший елемент веб-сторінки. Подібні ситуації називаються подіями. Сценарій - це одна або кілька команд, які запускаються при настанні події.

Розглянемо типовий випадок, коли в якості властивості розглядається стиль деякого елемента. Для звернення до об'єкта (Елементу P) використовується слово this.

Подією буде наведення миші: onmouseover

```
<P style = "color: green"
      onmouseover = "this.style.color = 'red'"> Інтерактивний
абзац </ p>
```

Ми можемо бачити, що колір тексту абзацу при наведенні на текст покажчика миші змінюється на червоний. Щоб спрацювала зворотна ситуація (колір змінювався знову на зелений), потрібно внести сценарій в опис події onmouseout:

```
<P style = "color: green"
       onmouseover = "this.style.color = 'red'"
       onmouseout = "this.style.color = 'green'"> Інтерактивний
абзац </ p>
```

Обробка даних користувача

Розглянемо два способи отримання інформації від користувача:

- введення тексту;
- підтвердження клацанням.

Для прийому тексту від користувача призначений елемент

```
<Input type = "text" name = "ім'я" />
```

Для створення кнопки призначений елемент

```
<Input type = "button" value = "Напис" onclick = "сценарій" />
```

Скористаємося командою alert (текст) для виведення на екран вікна з повідомленням:

```
<Input type = "text" name = "поле" />
<Input type = "button" value = "Значення поля" onclick = "alert
      (Поле.value) "/>
```

Рисунок 3.1 – Поле типу *input*

після клацання по кнопці з'являється текст, який Ви самі ввели

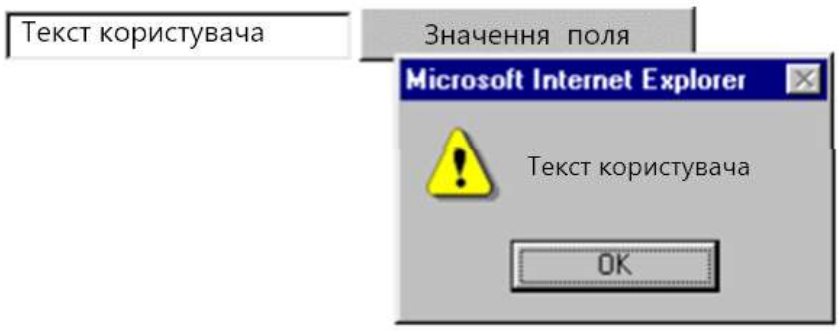

Рисунок 3.2 – Результат натиснення на кнопку

Застосування сценаріїв

Точно так же, як замість повторень одного і того ж вмісту атрибута style користуються одноразовим записом стилів у зовнішньому файлі або в елементі <style> </ style>, замість запису безпосередньо в описі події можна посилатися на сценарій, який розміщений в елементі <script> </ script>.

Як і <style> </ style>, він розміщується в елементі <head> </ head>.

Найчастіше сценарій з'являється там у вигляді функції.

```
function імя_функції (нуль або більше аргументів) {
        команди сценарію
}
```

Розглянемо приклад:

```
<script>
    function вітання (кого) {
       alert ( "Привіт," + кого + "!")
    }
</ script>
...
<input type = "text" name = "хто" />
<input type = "button" value = "Вітатися!"
      onclick = "вітання (хто.value)" />
```

Ми почали обговорення сценаріїв із згадки тестів на веб-сторінці. Щоб створити примітивний тест, нам знадобиться ще один елемент HTML, який приймає введення – радіоперемикач – і конструкція мови JScript, що виконує одну або іншу дію в залежності від умови – умовний оператор. Якщо обрано

правильний варіант відповіді, сценарій виводить позитивну реакцію, інакше – негативну.

```
if (умова) команда1; else команда2
```

Якщо умова істинна, то буде виконана команда1,

інакше буде виконана команда2.

Приклад:

В якому році закладено м. Київ?

```
<br/>
<Input type = "radio" name = "питання1" /> 1812 <br/>
<Input type = "radio" name = "питання1" /> 1703 <br/>
<Input type = "radio" name = "питання1" /> 1624 <br/>
<Input type = "button" value = "Перевірити!"
 onclick = "if (вопрос1 [1] .checked) alert ( 'Вірно!'); else
alert ( 'Немає вірної відповіді!') "/>
```

В даному прикладі умова – це обрання (checked) одного з радіоперемикачів. Всього є три перемикача, у яких одне ім'я (Питання1). Доступ до них здійснюється за допомогою індексу (номера в квадратних дужках, який починається з нуля – Питання1 [0], Питання 1 [1] і Питання 1 [2]).

Якщо питань два і більше, зручніше створити одну на всіх функцію перевірки. Вона могла б виглядати наступним чином:

```
function перевірка (вірний) {
 if (верний.checked) alert ( 'Вірно!'); else alert ( 'Ні вірної відповіді!')
}
...
<Input type = "button" value = "Перевірити!"
onclick = "перевірка (Питання1 [1])" />
```

Завдання

1. Створіть функцію, яку можна підключати як до події onmouseover, так і до події onmouseout, яка змінює колір кордону або на колір фону, або на

контрастний. Тоді при наведенні миші на елемент навколо нього буде вимальовуватися межа, а при видаленні – зникати.

2. Нехай є два текстових поля: число1 і число2. Команда alert (число1.value + число2.value) не даватиме суму, якщо в полях введені два числа.

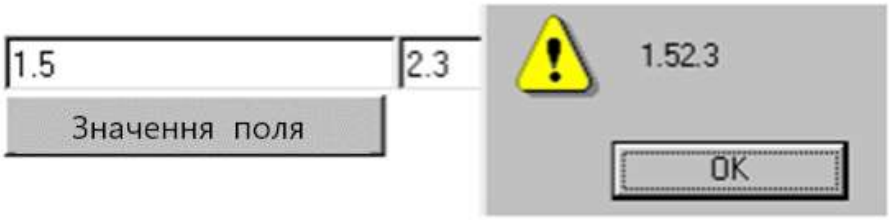

Рисунок 3.3 – Результат операції додавання двох значень полів *input*

Щоб вміст поля трактувався як число, потрібно піддати його перетворенню: parseFloat (число1.value)

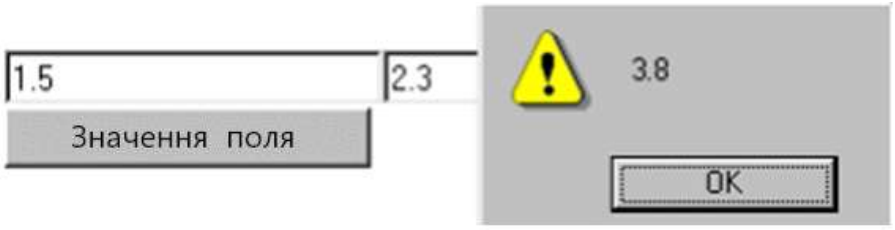

Рисунок 3.4 – Результат операції додавання з використанням функції *parseFloat*

Створіть калькулятор, який виконує чотири арифметичних дії (+, -, * і /). Реалізуйте для функції, що відповідає за розподіл, перевірку ділення на нуль.

3. Створіть тест з п'яти питань.

3 *. Якщо в елементі Script помістити команду **var ім'я**, то в пам'яті буде створення змінна *ім'я*. Створіть функцію, в якій значення змінної збільшується на 1 (ім'я = ім'я + 1 або ім'я ++) в разі правильної відповіді, а потім видається загальне число правильних відповідей. Відповідно, в цьому тесті буде не п'ять кнопок після кожного питання, а одна після всього тесту.

Додавання:

1. Якщо в блоці розгалуження потрібно використовувати кілька команд, застосовується форма:

```
if (умова) {
блок 1
} Else {
блок 2
 }
```

2. При необхідності використовувати цикл застосовують конструкцію for:

```
 for (змінна = значення; умова; операція) {
тіло циклу
}
```

Наприклад, підрахуємо суму чисел 1..10:

```
s = 0;
for (i = 1; i <11; i ++) {
s + = I;
alert (i);
}
```

3.2. Приклад виконання лабораторної роботи

1. Розробка інтернет магазину. Створіть чотири типові html-документи: index.html, cart.html, checkout.html, order.html. Таблиці стилів можна взяти із прикладу виконання лабораторної роботи №2. Їх вміст наступний:

index.html // Головна сторінка

```
<!DOCTYPE html>
<html>
<head>
    <title>Winery</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/style.css" media="screen" type="text/css" />
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
    <script type="text/javascript" src="js/jquery.shop.js"></script>
</head>
<body>
    <div id="site">
        <header id="masthead">
```

```
            <h1>Winery <span class="tagline">Wines for web developers since 1999</h1>
        </header>
        <div id="content">
            <div id="products">
                <ul>
                    <li>
                        <div class="product-image">
                            <img src="images/wine1.jpg" alt="" />
                        </div>
                        <div class="product-description" data-name="Wine #1" data-
price="5">
                            <h3 class="product-name">Wine #1</h3>
                            <p class="product-price">&euro; 5</p>
                            <form class="add-to-cart" action="cart.html" method="post">
                                <div>
                                    <label for="qty-1">Quantity</label>
                                    <input type="text" name="qty-1" id="qty-1"
class="qty" value="1" />
                                </div>
                                <p><input type="submit" value="Add to cart" class="btn"
/></p>
                            </form>
                        </div>
                    </li>
                    <li>
                        <div class="product-image">
                            <img src="images/wine2.jpg" alt="" />
                        </div>
                        <div class="product-description" data-name="Wine #2" data-
price="8">
                            <h3 class="product-name">Wine #2</h3>
                            <p class="product-price">&euro; 8</p>
                            <form class="add-to-cart" action="cart.html" method="post">
                                <div>
                                    <label for="qty-2">Quantity</label>
                                    <input type="text" name="qty-2" id="qty-2"
class="qty" value="1" />
                                </div>
                                <p><input type="submit" value="Add to cart" class="btn"
/></p>
                            </form>
                        </div>
                    </li>

                    <li>
                        <div class="product-image">
                            <img src="images/wine3.jpg" alt="" />
                        </div>
                        <div class="product-description" data-name="Wine #3" data-
price="11">
                            <h3 class="product-name">Wine #3</h3>
                            <p class="product-price">&euro; 11</p>
                            <form class="add-to-cart" action="cart.html" method="post">
                                <div>
                                    <label for="qty-3">Quantity</label>
                                    <input type="text" name="qty-3" id="qty-3"
class="qty" value="1" />
                                </div>
                                <p><input type="submit" value="Add to cart" class="btn"
/></p>
                            </form>
                        </div>
                    </li>
                </ul>
            </div>
```

```
        </div>
    </div>
    <footer id="site-info">
        Copyright &copy; Winery
    </footer>
</body>
</html>
```

cart.html // Сторінка корзини товарів

```
<!DOCTYPE html>
<html>
<head>
    <title>Winery: Your Shopping Cart</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/style.css" media="screen" type="text/css" />
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
    <script type="text/javascript" src="js/jquery.shop.js"></script>
</head>
<body>
    <div id="site">
        <header id="masthead">
            <h1>Winery <span class="tagline">Wines for web developers since 1999</h1>
        </header>
        <div id="content">
            <h1>Your Shopping Cart</h1>
            <form id="shopping-cart" action="cart.html" method="post">
                <table class="shopping-cart">
                    <thead>
                        <tr>
                            <th scope="col">Item</th>
                            <th scope="col">Qty</th>
                            <th scope="col" colspan="2">Price</th>
                        </tr>
                    </thead>
                    <tbody>
                    </tbody>
                </table>
                <p id="sub-total">
                    <strong>Sub Total</strong>: <span id="stotal"></span>
                </p>
                <ul id="shopping-cart-actions">
                    <li>
                        <input type="submit" name="update" id="update-cart" class="btn"
value="Update Cart" />
                    </li>
                    <li>
                        <input type="submit" name="delete" id="empty-cart" class="btn"
value="Empty Cart" />
                    </li>
                    <li>
                        <a href="index.html" class="btn">Continue Shopping</a>
                    </li>
                    <li>
                        <a href="checkout.html" class="btn">Go To Checkout</a>
                    </li>
                </ul>
            </form>
        </div>
    </div>
    <footer id="site-info">
        Copyright &copy; Winery
    </footer>
```

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

checkout.html // Сторінка заповнення даних клієнта

```
<!DOCTYPE html>
<html>
<head>
    <title>Winery: Checkout</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/style.css" media="screen" type="text/css" />
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
    <script type="text/javascript" src="js/jquery.shop.js"></script>
</head>
<body id="checkout-page">
    <div id="site">
        <header id="masthead">
            <h1>Winery <span class="tagline">Wines for web developers since 1999</h1>
        </header>
        <div id="content">
            <h1>Checkout</h1>
            <table id="checkout-cart" class="shopping-cart">
                <thead>
                    <tr>
                        <th scope="col">Item</th>
                        <th scope="col">Qty</th>
                        <th scope="col">Price</th>
                    </tr>
                </thead>
                <tbody>
                </tbody>
            </table>
            <div id="pricing">
                <p id="shipping">
                    <strong>Shipping</strong>: <span id="sshipping"></span>
                </p>
                <p id="sub-total">
                    <strong>Total</strong>: <span id="stotal"></span>
                </p>
            </div>
            <form action="order.html" method="post" id="checkout-order-form">
                <h2>Your Details</h2>

                <fieldset id="fieldset-billing">
                    <legend>Billing</legend>
                    <div>
                        <label for="name">Name</label>
                        <input type="text" name="name" id="name" data-type="string" data-
message="This field cannot be empty" />
                    </div>
                    <div>
                        <label for="email">Email</label>
                        <input type="text" name="email" id="email" data-type="expression"
data-message="Not a valid email address" />
                    </div>
                    <div>
                        <label for="city">City</label>
                        <input type="text" name="city" id="city" data-type="string" data-
message="This field cannot be empty" />
                    </div>
                    <div>
                        <label for="address">Address</label>
```

```
                        <input type="text" name="address" id="address" data-type="string"
data-message="This field cannot be empty" />
                    </div>
                    <div>
                        <label for="zip">ZIP Code</label>
                        <input type="text" name="zip" id="zip" data-type="string" data-
message="This field cannot be empty" />
                    </div>
                    <div>
                        <label for="country">Country</label>
                        <select name="country" id="country" data-type="string" data-
message="This field cannot be empty">
                            <option value="">Select</option>
                            <option value="US">USA</option>
                            <option value="IT">Italy</option>
                        </select>
                    </div>
                </fieldset>
                <div id="shipping-same">Same as Billing <input type="checkbox" id="same-
as-billing" value="" /></div>

                <fieldset id="fieldset-shipping">

                    <legend>Shipping</legend>

                    <div>
                        <label for="sname">Name</label>
                        <input type="text" name="sname" id="sname" data-type="string"
data-message="This field cannot be empty" />
                    </div>
                    <div>
                        <label for="semail">Email</label>
                        <input type="text" name="semail" id="semail" data-
type="expression" data-message="Not a valid email address" />
                    </div>
                    <div>
                        <label for="scity">City</label>
                        <input type="text" name="scity" id="scity" data-type="string"
data-message="This field cannot be empty" />
                    </div>
                    <div>
                        <label for="saddress">Address</label>
                        <input type="text" name="saddress" id="saddress" data-
type="string" data-message="This field cannot be empty" />
                    </div>
                    <div>
                        <label for="szip">ZIP Code</label>
                        <input type="text" name="szip" id="szip" data-type="string" data-
message="This field cannot be empty" />
                    </div>
                    <div>
                        <label for="scountry">Country</label>
                        <select name="scountry" id="scountry" data-type="string" data-
message="This field cannot be empty">
                            <option value="">Select</option>
                            <option value="US">USA</option>
                            <option value="IT">Italy</option>
                        </select>
                    </div>
                </fieldset>
                <p><input type="submit" id="submit-order" value="Submit" class="btn"
/></p>
            </form>
        </div>
    </div>
```

```
    <footer id="site-info">
        Copyright &copy; Winery
    </footer>
</body>
</html>
```

order.html // Сторінка підтвердження замовлення

```
<!DOCTYPE html>
<html>
<head>
    <title>Winery: Your Order</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/style.css" media="screen" type="text/css" />
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
    <script type="text/javascript" src="js/jquery.shop.js"></script>
</head>
<body id="checkout-page">
    <div id="site">
        <header id="masthead">
            <h1>Winery <span class="tagline">Wines for web developers since 1999</h1>
        </header>
        <div id="content">
            <h1>Your Order</h1>
            <table id="checkout-cart" class="shopping-cart">
                <thead>
                    <tr>
                        <th scope="col">Item</th>
                        <th scope="col">Qty</th>
                        <th scope="col">Price</th>
                    </tr>
                </thead>
                <tbody>
                </tbody>
            </table>
            <div id="pricing">
                <p id="shipping">
                    <strong>Shipping</strong>: <span id="sshipping"></span>
                </p>
                <p id="sub-total">
                    <strong>Total</strong>: <span id="stotal"></span>
                </p>
            </div>
            <div id="user-details">
                <h2>Your Data</h2>
                <div id="user-details-content"></div>
            </div>
            <form id="paypal-form" action="" method="post">
                <input type="hidden" name="cmd" value="_cart" />
                <input type="hidden" name="upload" value="1" />
                <input type="hidden" name="business" value="" />
                <input type="hidden" name="currency_code" value="" />
                <input type="submit" id="paypal-btn" class="btn" value="Pay with PayPal"
/>
            </form>
        </div>
    </div>
    <footer id="site-info">
        Copyright &copy; Winery
    </footer>
</body>
</html>
```

2. Створіть файл з назвою *jquery.shop* в папці *js* формату *.js. Всередині цього фалу розмістіть javascript код:

```
(function ($) {
	$.Shop = function (element) {
		this.$element = $(element);
		this.init();
	};

	$.Shop.prototype = {
		init: function () {

			// Properties

			this.cartPrefix = "winery-"; // Prefix string to be prepended to the
cart's name in the session storage
			this.cartName = this.cartPrefix + "cart"; // Cart name in the session
storage
			this.shippingRates = this.cartPrefix + "shipping-rates"; // Shipping
rates key in the session storage
			this.total = this.cartPrefix + "total"; // Total key in the session
storage
			this.storage = sessionStorage; // shortcut to the sessionStorage
object


			this.$formAddToCart = this.$element.find("form.add-to-cart"); //
Forms for adding items to the cart
			this.$formCart = this.$element.find("#shopping-cart"); // Shopping
cart form
			this.$checkoutCart = this.$element.find("#checkout-cart"); //
Checkout form cart
			this.$checkoutOrderForm = this.$element.find("#checkout-order-form");
// Checkout user details form
			this.$shipping = this.$element.find("#sshipping"); // Element that
displays the shipping rates
			this.$subTotal = this.$element.find("#stotal"); // Element that
displays the subtotal charges
			this.$shoppingCartActions = this.$element.find("#shopping-cart-
actions"); // Cart actions links
			this.$updateCartBtn = this.$shoppingCartActions.find("#update-cart");
// Update cart button
			this.$emptyCartBtn = this.$shoppingCartActions.find("#empty-cart");
// Empty cart button
			this.$userDetails = this.$element.find("#user-details-content"); //
Element that displays the user information
			this.$paypalForm = this.$element.find("#paypal-form"); // PayPal form


			this.currency = "&euro;"; // HTML entity of the currency to be
displayed in the layout
			this.currencyString = "€"; // Currency symbol as textual string
			this.paypalCurrency = "EUR"; // PayPal's currency code
			this.paypalBusinessEmail = "yourbusiness@email.com"; // Your Business
PayPal's account email address
			this.paypalURL = "https://www.sandbox.paypal.com/cgi-bin/webscr"; //
The URL of the PayPal's form

			// Object containing patterns for form validation
			this.requiredFields = {
				expression: {
					value: /^([\w-\.]+)@((?:[\w]+\.)+)([a-z]){2,4}$/
```

```
                        },

                        str: {
                                value: ""
                        }

                };

                // Method invocation

                this.createCart();
                this.handleAddToCartForm();
                this.handleCheckoutOrderForm();
                this.emptyCart();
                this.updateCart();
                this.displayCart();
                this.deleteProduct();
                this.displayUserDetails();
                this.populatePayPalForm();


        },

        // Public methods

        // Creates the cart keys in the session storage

        createCart: function () {
                if (this.storage.getItem(this.cartName) == null) {

                        var cart = {};
                        cart.items = [];

                        this.storage.setItem(this.cartName, this._toJSONString(cart));
                        this.storage.setItem(this.shippingRates, "0");
                        this.storage.setItem(this.total, "0");
                }
        },

        // Appends the required hidden values to the PayPal's form before
submitting

        populatePayPalForm: function () {
                var self = this;
                if (self.$paypalForm.length) {
                        var $form = self.$paypalForm;
                        var cart =
self._toJSONObject(self.storage.getItem(self.cartName));
                        var shipping = self.storage.getItem(self.shippingRates);
                        var numShipping = self._convertString(shipping);
                        var cartItems = cart.items;
                        var singShipping = Math.floor(numShipping / cartItems.length);

                        $form.attr("action", self.paypalURL);

        $form.find("input[name='business']").val(self.paypalBusinessEmail);

        $form.find("input[name='currency_code']").val(self.paypalCurrency);

                        for (var i = 0; i < cartItems.length; ++i) {
                                var cartItem = cartItems[i];
                                var n = i + 1;
                                var name = cartItem.product;
                                var price = cartItem.price;
                                var qty = cartItem.qty;
```

```
                                    $("<div/>").html("<input type='hidden' name='quantity_"
+ n + "' value='" + qty + "'/>").
                                          insertBefore("#paypal-btn");
                                    $("<div/>").html("<input type='hidden'
name='item_name_" + n + "' value='" + name + "'/>").
                                          insertBefore("#paypal-btn");
                                    $("<div/>").html("<input type='hidden'
name='item_number_" + n + "' value='SKU " + name + "'/>").
                                          insertBefore("#paypal-btn");
                                    $("<div/>").html("<input type='hidden' name='amount_" +
n + "' value='" + self._formatNumber(price, 2) + "'/>").
                                          insertBefore("#paypal-btn");
                                    $("<div/>").html("<input type='hidden' name='shipping_"
+ n + "' value='" + self._formatNumber(singShipping, 2) + "'/>").
                                          insertBefore("#paypal-btn");

                              }



                        }
                  },

                  // Displays the user's information

                  displayUserDetails: function () {
                        if (this.$userDetails.length) {
                              if (this.storage.getItem("shipping-name") == null) {
                                    var name = this.storage.getItem("billing-name");
                                    var email = this.storage.getItem("billing-email");
                                    var city = this.storage.getItem("billing-city");
                                    var address = this.storage.getItem("billing-address");
                                    var zip = this.storage.getItem("billing-zip");
                                    var country = this.storage.getItem("billing-country");

                                    var html = "<div class='detail'>";
                                    html += "<h2>Billing and Shipping</h2>";
                                    html += "<ul>";
                                    html += "<li>" + name + "</li>";
                                    html += "<li>" + email + "</li>";
                                    html += "<li>" + city + "</li>";
                                    html += "<li>" + address + "</li>";
                                    html += "<li>" + zip + "</li>";
                                    html += "<li>" + country + "</li>";
                                    html += "</ul></div>";

                                    this.$userDetails[0].innerHTML = html;
                              } else {
                                    var name = this.storage.getItem("billing-name");
                                    var email = this.storage.getItem("billing-email");
                                    var city = this.storage.getItem("billing-city");
                                    var address = this.storage.getItem("billing-address");
                                    var zip = this.storage.getItem("billing-zip");
                                    var country = this.storage.getItem("billing-country");

                                    var sName = this.storage.getItem("shipping-name");
                                    var sEmail = this.storage.getItem("shipping-email");
                                    var sCity = this.storage.getItem("shipping-city");
                                    var sAddress = this.storage.getItem("shipping-
address");
                                    var sZip = this.storage.getItem("shipping-zip");
                                    var sCountry = this.storage.getItem("shipping-
country");
```

```
                                var html = "<div class='detail'>";
                                html += "<h2>Billing</h2>";
                                html += "<ul>";
                                html += "<li>" + name + "</li>";
                                html += "<li>" + email + "</li>";
                                html += "<li>" + city + "</li>";
                                html += "<li>" + address + "</li>";
                                html += "<li>" + zip + "</li>";
                                html += "<li>" + country + "</li>";
                                html += "</ul></div>";

                                html += "<div class='detail right'>";
                                html += "<h2>Shipping</h2>";
                                html += "<ul>";
                                html += "<li>" + sName + "</li>";
                                html += "<li>" + sEmail + "</li>";
                                html += "<li>" + sCity + "</li>";
                                html += "<li>" + sAddress + "</li>";
                                html += "<li>" + sZip + "</li>";
                                html += "<li>" + sCountry + "</li>";
                                html += "</ul></div>";

                                this.$userDetails[0].innerHTML = html;

                        }
                }
        },

        // Delete a product from the shopping cart

        deleteProduct: function () {
                var self = this;
                if (self.$formCart.length) {
                        var cart =
this._toJSONObject(this.storage.getItem(this.cartName));
                        var items = cart.items;

                        $(document).on("click", ".pdelete a", function (e) {
                                e.preventDefault();
                                var productName = $(this).data("product");
                                var newItems = [];
                                for (var i = 0; i < items.length; ++i) {
                                        var item = items[i];
                                        var product = item.product;
                                        if (product == productName) {
                                                items.splice(i, 1);
                                        }
                                }
                                newItems = items;
                                var updatedCart = {};
                                updatedCart.items = newItems;

                                var updatedTotal = 0;
                                var totalQty = 0;
                                if (newItems.length == 0) {
                                        updatedTotal = 0;
                                        totalQty = 0;
                                } else {
                                        for (var j = 0; j < newItems.length; ++j) {
                                                var prod = newItems[j];
                                                var sub = prod.price * prod.qty;
                                                updatedTotal += sub;
                                                totalQty += prod.qty;
                                        }
                                }
```

```
                                    self.storage.setItem(self.total,
self._convertNumber(updatedTotal));
                                    self.storage.setItem(self.shippingRates,
self._convertNumber(self._calculateShipping(totalQty)));

                                    self.storage.setItem(self.cartName,
self._toJSONString(updatedCart));
                                    $(this).parents("tr").remove();
                                    self.$subTotal[0].innerHTML = self.currency + " " +
self.storage.getItem(self.total);
                             });
                      }
              },

              // Displays the shopping cart

              displayCart: function () {
                     if (this.$formCart.length) {
                             var cart =
this._toJSONObject(this.storage.getItem(this.cartName));
                             var items = cart.items;
                             var $tableCart = this.$formCart.find(".shopping-cart");
                             var $tableCartBody = $tableCart.find("tbody");

                             if (items.length == 0) {
                                    $tableCartBody.html("");
                             } else {


                                    for (var i = 0; i < items.length; ++i) {
                                           var item = items[i];
                                           var product = item.product;
                                           var price = this.currency + " " + item.price;
                                           var qty = item.qty;
                                           var html = "<tr><td class='pname'>" + product +
"</td>" + "<td class='pqty'><input type='text' value='" + qty + "' class='qty'/></td>";
                                           html += "<td class='pprice'>" + price +
"</td><td class='pdelete'><a href='' data-product='" + product +
"'>&times;</a></td></tr>";

                                           $tableCartBody.html($tableCartBody.html() +
html);
                                    }

                             }

                             if (items.length == 0) {
                                    this.$subTotal[0].innerHTML = this.currency + " " +
0.00;
                             } else {

                                    var total = this.storage.getItem(this.total);
                                    this.$subTotal[0].innerHTML = this.currency + " " +
total;
                             }
                      } else if (this.$checkoutCart.length) {
                             var checkoutCart =
this._toJSONObject(this.storage.getItem(this.cartName));
                             var cartItems = checkoutCart.items;
                             var $cartBody = this.$checkoutCart.find("tbody");

                             if (cartItems.length > 0) {

                                    for (var j = 0; j < cartItems.length; ++j) {
```

```
                                        var cartItem = cartItems[j];
                                        var cartProduct = cartItem.product;
                                        var cartPrice = this.currency + " " +
cartItem.price;
                                        var cartQty = cartItem.qty;
                                        var cartHTML = "<tr><td class='pname'>" +
cartProduct + "</td>" + "<td class='pqty'>" + cartQty + "</td>" + "<td class='pprice'>" +
cartPrice + "</td></tr>";

                                        $cartBody.html($cartBody.html() + cartHTML);
                                }
                        } else {
                                $cartBody.html("");
                        }

                        if (cartItems.length > 0) {

                                var cartTotal = this.storage.getItem(this.total);
                                var cartShipping =
this.storage.getItem(this.shippingRates);
                                var subTot = this._convertString(cartTotal) +
this._convertString(cartShipping);

                                this.$subTotal[0].innerHTML = this.currency + " " +
this._convertNumber(subTot);
                                this.$shipping[0].innerHTML = this.currency + " " +
cartShipping;
                        } else {
                                this.$subTotal[0].innerHTML = this.currency + " " +
0.00;
                                this.$shipping[0].innerHTML = this.currency + " " +
0.00;
                        }

                }
        },

        // Empties the cart by calling the _emptyCart() method
        // @see $.Shop._emptyCart()

        emptyCart: function () {
                var self = this;
                if (self.$emptyCartBtn.length) {
                        self.$emptyCartBtn.on("click", function () {
                                self._emptyCart();
                        });
                }
        },

        // Updates the cart

        updateCart: function () {
                var self = this;
                if (self.$updateCartBtn.length) {
                        self.$updateCartBtn.on("click", function () {
                                var $rows = self.$formCart.find("tbody tr");
                                var cart = self.storage.getItem(self.cartName);
                                var shippingRates =
self.storage.getItem(self.shippingRates);
                                var total = self.storage.getItem(self.total);

                                var updatedTotal = 0;
                                var totalQty = 0;
                                var updatedCart = {};
                                updatedCart.items = [];
```

```
                                $rows.each(function () {
                                        var $row = $(this);
                                        var pname = $.trim($row.find(".pname").text());
                                        var pqty = self._convertString($row.find(".pqty
> .qty").val());
                                        var pprice =
self._convertString(self._extractPrice($row.find(".pprice")));

                                        var cartObj = {
                                                product: pname,
                                                price: pprice,
                                                qty: pqty
                                        };

                                        updatedCart.items.push(cartObj);

                                        var subTotal = pqty * pprice;
                                        updatedTotal += subTotal;
                                        totalQty += pqty;
                                });

                                self.storage.setItem(self.total,
self._convertNumber(updatedTotal));
                                self.storage.setItem(self.shippingRates,
self._convertNumber(self._calculateShipping(totalQty)));
                                self.storage.setItem(self.cartName,
self._toJSONString(updatedCart));

                        });
                }
        },

        // Adds items to the shopping cart

        handleAddToCartForm: function () {
                var self = this;
                self.$formAddToCart.each(function () {
                        var $form = $(this);
                        var $product = $form.parent();
                        var price = self._convertString($product.data("price"));
                        var name = $product.data("name");

                        $form.on("submit", function () {
                                var qty =
self._convertString($form.find(".qty").val());
                                var subTotal = qty * price;
                                var total =
self._convertString(self.storage.getItem(self.total));
                                var sTotal = total + subTotal;
                                self.storage.setItem(self.total, sTotal);
                                self._addToCart({
                                        product: name,
                                        price: price,
                                        qty: qty
                                });
                                var shipping =
self._convertString(self.storage.getItem(self.shippingRates));
                                var shippingRates = self._calculateShipping(qty);
                                var totalShipping = shipping + shippingRates;

                                self.storage.setItem(self.shippingRates,
totalShipping);
                        });
                });
```

```
		},

		// Handles the checkout form by adding a validation routine and saving
user's info into the session storage

		handleCheckoutOrderForm: function () {
			var self = this;
			if (self.$checkoutOrderForm.length) {
				var $sameAsBilling = $("#same-as-billing");
				$sameAsBilling.on("change", function () {
					var $check = $(this);
					if ($check.prop("checked")) {
						$("#fieldset-shipping").slideUp("normal");
					} else {
						$("#fieldset-shipping").slideDown("normal");
					}
				});

				self.$checkoutOrderForm.on("submit", function () {
					var $form = $(this);
					var valid = self._validateForm($form);

					if (!valid) {
						return valid;
					} else {
						self._saveFormData($form);
					}
				});
			}
		},

		// Private methods


		// Empties the session storage

		_emptyCart: function () {
			this.storage.clear();
		},

		/* Format a number by decimal places
		 * @param num Number the number to be formatted
		 * @param places Number the decimal places
		 * @returns n Number the formatted number
		 */



		_formatNumber: function (num, places) {
			var n = num.toFixed(places);
			return n;
		},

		/* Extract the numeric portion from a string
		 * @param element Object the jQuery element that contains the relevant
string
		 * @returns price String the numeric string
		 */


		_extractPrice: function (element) {
			var self = this;
			var text = element.text();
			var price = text.replace(self.currencyString, "").replace(" ", "");
			return price;
```

```
},

/* Converts a numeric string into a number
 * @param numStr String the numeric string to be converted
 * @returns num Number the number
 */

_convertString: function (numStr) {
	var num;
	if (/^[-+]?[0-9]+\.[0-9]+$/.test(numStr)) {
		num = parseFloat(numStr);
	} else if (/^\d+$/.test(numStr)) {
		num = parseInt(numStr, 10);
	} else {
		num = Number(numStr);
	}

	if (!isNaN(num)) {
		return num;
	} else {
		console.warn(numStr + " cannot be converted into a number");
		return false;
	}
},

/* Converts a number to a string
 * @param n Number the number to be converted
 * @returns str String the string returned
 */

_convertNumber: function (n) {
	var str = n.toString();
	return str;
},

/* Converts a JSON string to a JavaScript object
 * @param str String the JSON string
 * @returns obj Object the JavaScript object
 */

_toJSONObject: function (str) {
	var obj = JSON.parse(str);
	return obj;
},

/* Converts a JavaScript object to a JSON string
 * @param obj Object the JavaScript object
 * @returns str String the JSON string
 */


_toJSONString: function (obj) {
	var str = JSON.stringify(obj);
	return str;
},


/* Add an object to the cart as a JSON string
 * @param values Object the object to be added to the cart
 * @returns void
 */


_addToCart: function (values) {
	var cart = this.storage.getItem(this.cartName);
```

```
                var cartObject = this._toJSONObject(cart);
                var cartCopy = cartObject;
                var items = cartCopy.items;
                items.push(values);

                this.storage.setItem(this.cartName, this._toJSONString(cartCopy));
        },

        /* Custom shipping rates calculation based on the total quantity of items
in the cart
         * @param qty Number the total quantity of items
         * @returns shipping Number the shipping rates
         */

        _calculateShipping: function (qty) {
                var shipping = 0;
                if (qty >= 6) {
                        shipping = 10;
                }
                if (qty >= 12 && qty <= 30) {
                        shipping = 20;
                }

                if (qty >= 30 && qty <= 60) {
                        shipping = 30;
                }

                if (qty > 60) {
                        shipping = 0;
                }

                return shipping;

        },

        /* Validates the checkout form
         * @param form Object the jQuery element of the checkout form
         * @returns valid Boolean true for success, false for failure
         */



        _validateForm: function (form) {
                var self = this;
                var fields = self.requiredFields;
                var $visibleSet = form.find("fieldset:visible");
                var valid = true;

                form.find(".message").remove();

                $visibleSet.each(function () {

                        $(this).find(":input").each(function () {
                                var $input = $(this);
                                var type = $input.data("type");
                                var msg = $input.data("message");

                                if (type == "string") {
                                        if ($input.val() == fields.str.value) {
                                                $("<span class='message'/>").text(msg).
                                                        insertBefore($input);

                                                valid = false;
                                        }
```

```
                                } else {
                                        if (!fields.expression.value.test($input.val()))
{
                                                $("<span class='message'/>").text(msg).
                                                        insertBefore($input);

                                                valid = false;
                                        }
                                }

                        });
                });

                return valid;

        },

        /* Save the data entered by the user in the ckeckout form
         * @param form Object the jQuery element of the checkout form
         * @returns void
         */


        _saveFormData: function (form) {
                var self = this;
                var $visibleSet = form.find("fieldset:visible");

                $visibleSet.each(function () {
                        var $set = $(this);
                        if ($set.is("#fieldset-billing")) {
                                var name = $("#name", $set).val();
                                var email = $("#email", $set).val();
                                var city = $("#city", $set).val();
                                var address = $("#address", $set).val();
                                var zip = $("#zip", $set).val();
                                var country = $("#country", $set).val();

                                self.storage.setItem("billing-name", name);
                                self.storage.setItem("billing-email", email);
                                self.storage.setItem("billing-city", city);
                                self.storage.setItem("billing-address", address);
                                self.storage.setItem("billing-zip", zip);
                                self.storage.setItem("billing-country", country);
                        } else {
                                var sName = $("#sname", $set).val();
                                var sEmail = $("#semail", $set).val();
                                var sCity = $("#scity", $set).val();
                                var sAddress = $("#saddress", $set).val();
                                var sZip = $("#szip", $set).val();
                                var sCountry = $("#scountry", $set).val();

                                self.storage.setItem("shipping-name", sName);
                                self.storage.setItem("shipping-email", sEmail);
                                self.storage.setItem("shipping-city", sCity);
                                self.storage.setItem("shipping-address", sAddress);
                                self.storage.setItem("shipping-zip", sZip);
                                self.storage.setItem("shipping-country", sCountry);

                        }
                });
        }
};

$(function () {
        var shop = new $.Shop("#site");
```

```
    });
})(jQuery);
```

3. Даний приклад демонструє інтерактивне переключення між сторінками інтернет магазину з вибором кількості товарів, заповненням і очищенням корзини, а також заповненням форми замовлення.

3.3. Питання для самоконтролю

1. Можливості JavaScript, де застосовується.
2. Робота JavaScript з DOM сторінки.
3. JavaScript команди і коментарі.
4. Змінні в JavaScript і операції над ними.
5. Арифметичні оператори в JavaScript.
6. Логічні оператори та умовні конструкції в JavaScript.
7. Вікна оповіщення і підтвердження в JavaScript.
8. JavaScript функції.
9. Локальні і глобальні змінні.
10. Цикли JavaScript.
11. Події та їх обробка.
12. Перевірка форм в JavaScript.
13. Спеціальні символи в JavaScript.
14. Методи об'єктів в JavaScript.
15. Масиви в JavaScript.