

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»
МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА КІБЕРНЕТИКИ І ПРИКЛАДНОЇ МАТЕМАТИКИ**

**СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ:
WEB-програмування на боці клієнта. HTML та CSS**

Ужгород-2015

Мулеса О.Ю. Сучасні інформаційні технології. WEB-програмування на боці клієнта. HTML та CSS. Методичні рекомендації для студентів 2-го курсу математичного факультету спеціальності "Прикладна математика"– Ужгород, 2015. – 53 с.

Рекомендовано до друку кафедрою кібернетики і прикладної математики ДВНЗ "Ужгородський національний університет", протокол №6 від 12 лютого 2015 року.

Рекомендовано до друку методичною комісією математичного факультету ДВНЗ "Ужгородський національний університет", протокол №2 від 17 лютого 2015 року.

Рекомендовано до друку вченою радою математичного факультету ДВНЗ "Ужгородський національний університет", протокол №7 від 19 лютого 2015 року.

Рецензенти: **Гече Ф.Е.**, доктор технічних наук, професор, завідувач кафедри кібернетики і прикладної математики (ДВНЗ "Ужгородський національний університет")

Погоріляк О.О., кандидат фізико-математичних наук, доцент, доцент кафедри теорії ймовірностей та математичної статистики (ДВНЗ "Ужгородський національний університет")

ЗМІСТ

1. МОВА РОЗМІТКИ HTML.....	5
1.1. Поняття про HTML.....	5
1.2. Інструменти для роботи з HTML.....	6
1.3. Структура HTML-документа.....	7
1.4. Форматування тексту.....	8
1.5. Форматування абзаців.....	9
1.6. Робота зі списками.....	11
1.7. Форматування таблиць.....	13
1.8. Встановлення кольорів та зображень.....	16
1.9. Гіперпосилання в документі.....	17
2. КАСКАДНІ ТАБЛИЦІ СТИЛІВ CSS.....	18
2.1. Поняття про CSS.....	18
2.2. Синтаксис CSS та його застосування.....	20
2.3. Управління кольором та фоном.....	21
2.4. Форматування шрифтів.....	23
2.5. Форматування тексту.....	25
2.6. Форматування гіперпосилань.....	26
2.7. Селектори атрибутів.....	27
2.8. Класи.....	31

2.9. Ідентифікатори.....	36
2.10. Взаємне розміщення селекторів.....	38
2.11. Універсальний селектор.....	45
2.12. Псевдоелементи.....	46
2.13. Каскадування та специфічність стилів.....	50
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ ТА ЕЛЕКТРОН- НИХ РЕСУРСІВ.....	53

1. МОВА РОЗМІТКИ HTML

1.1. Поняття про HTML

HTML (від англ. HyperText Markup Language – «мова гіпертекстової розмітки») – стандартна мова розмітки документів в мережі Інтернет. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML). Мова HTML інтерпретується браузерами і відображається у вигляді документа в зручній для людини формі.

Мова HTML є додатком SGML (стандартної узагальненої мови розмітки) і відповідає міжнародному стандарту ISO 8879.

У всесвітній павутині HTML-сторінки, як правило, передаються браузерам від сервера по протоколах HTTP або HTTPS, у вигляді простого тексту або з використанням шифрування.

Мова HTML була розроблена британським вченим Тімом Бернерс-Лі приблизно в 1986-1991 роках в стінах Європейського Центру ядерних досліджень в Женеві (Швейцарія). Елементами є структурні і семантичні елементи – дескриптори. Дескриптори також часто називають «тегами». Також, в HTML внесена підтримка гіпертексту.

Таким чином, веб-сторінки створюються за допомогою мови розмітки гіпертексту HTML. Опис веб-сторінок міститься в HTML-програмі, який зберігається у звичайному текстовому файлі з розширенням htm чи html. Програми мовою HTML містять інструкції (коди), що називаються тегами. Всі теги мови HTML виділяються символами-обмежувачами < i >, між якими записується ідентифікатор (ім'я) тега i, можливо, його параметри. Більшість тегів HTML використовується попарно, тобто для окремого тега (відкриваючого) в документі є відповідний закриваючий тег, який записується так само, як і відкриваючий, але з символом / перед іменем тега (закриваючі теги не використовують параметри).

1.2. Інструменти для роботи з HTML

Для ефективної роботи не обійтися без необхідних і звичних інструментів, у тому числі і при написанні коду HTML. Тому для початкової розробки веб-сторінок або навіть невеликого сайту – так називається набір сторінок, пов'язаних між собою посиланнями і єдиним оформленням, можуть знадобитися наступні програми.

- текстовий або WYSIWYG редактор;
- браузер для перегляду результатів;
- валідатор – програма для перевірки синтаксису html і виявлення помилок в коді;
- графічний редактор;
- довідник з тегами html.

Далі розглянемо ці інструменти докладніше.

Текстовий редактор. HTML-документ можна створювати в будь-якому текстовому редакторі, хоч і в Блокноті. Тим не менш, для цієї мети підійде не будь-яка програма. Потрібна така, щоб забезпечувала такі можливості:

- підсвічування синтаксису – виділення тегів, тексту, ключових слів і параметрів різними кольорами. Це полегшує пошук потрібного елемента, прискорює роботу розробника і знижує виникнення помилок;
- робота з вкладками. Сайт являє собою набір файлів, які доводиться правити окремо, для чого потрібен редактор, що вміє одночасно працювати відразу з декількома документами. При цьому файли зручно відкривати в окремих вкладках, щоб швидко переходити до потрібного документа;
- перевірка поточного документа на помилки.

WYSIWYG редактор (від англ. What You See Is What You Get) – прикладна програма, в якій вміст відображується в процесі редагування та виглядає схоже на кінцеву веб-сторінку.

Браузер. Браузер це програма, призначена для перегляду веб-сторінок. На перших порах підійде будь-який браузер, але з підвищенням досвіду і знань потрібно використовувати декілька, щоб перевіряти правильність

відображення сайту в різних браузерах. Справа в тому, що кожен браузер має свої унікальні особливості, тому для перевірки універсальності коду потрібно переглядати і коригувати код з їх урахуванням. На сьогоднішній день найбільшою популярністю користуються три браузера: Firefox, Internet Explorer, Opera, Chrome.

Валідатор. Валідація HTML-документа призначена для виявлення помилок в синтаксисі веб-сторінки і розбіжностей зі специфікацією HTML. Відповідно, програма або система для такої перевірки називається валідатором. Для таких цілей можна використовувати Інтернет ресурс за адресою <http://validator.w3.org> або ж програму Tidy.

Графічний редактор. Графічний редактор необхідний для обробки зображень і їх підготовки для публікації на веб-сторінці. Найпопулярнішою програмою такого роду є Photoshop, що став стандартом для обробки фотографій і створення графічних зображень для сайтів. Але в більшості випадків програма Paint.Net дозволяє зробити всі необхідні маніпуляції із зображеннями.

1.3. Структура HTML-документа

В загальному, структура HTML-документа є такою:

```
<HTML>
<HEAD>
<TITLE> Заголовок документа </TITLE>
</HEAD>
<BODY>
    Тіло документа
</BODY>
</HTML>
```

Парою тегів `<HTML> . . . </HTML>` визначається початок і кінець документа.

Розділ документа **HEAD** визначає його заголовок, розміщений між парою тегів `<TITLE>` і `</TITLE>`, а також мета-теги. Тег `<META>` містить різнорідну інформацію для броузерів, яка не відображається у вікні. Тег

<STYLE> пов'язаний з використанням таблиць стилів. Тег <SCRIPT> використовується для запису сценаріїв.

Розділ документа **BODY** може мати такі параметри:

| Параметр | Опис дії параметра |
|---------------------|---|
| ALINK | Визначає колір активного посилання |
| LINK | Визначає колір ще не переглянутого посилання |
| VLINK | Визначає колір уже переглянутого посилання |
| BACKGROUND | Визначає адресу фонового зображення |
| BGCOLOR | Визначає колір фонового зображення |
| TEXT | Визначає колір тексту |
| BGPROPERTIES | Якщо встановлене значення FIXED , фонове зображення не прокручується |
| BOTTOMMARGIN | Встановлює межу нижнього поля в пікселях |
| LEFTMARGIN | Встановлює межу лівого поля в пікселях |
| RIGHTMARGIN | Встановлює межу правого поля в пікселях |
| TOPMARGIN | Встановлює межу верхнього поля в пікселях |

1.4. Форматування тексту

Таким чином, веб-сторінки створюються за допомогою мови розмітки гіпертексту HTML. Опис веб-сторінок міститься в HTML-програмі, який зберігається у звичайному текстовому файлі з розширенням `htm` чи `html`. Програми мовою HTML містять інструкції (коди), що називаються тегами. Всі теги мови HTML виділяються символами-обмежувачами `< i >`, між якими записується ідентифікатор (ім'я) тега `i`, можливо, його параметри. Більшість тегів HTML використовується попарно, тобто для окремого тега (відкриваючого) в документі є відповідний закриваючий тег, який записується так само, як і відкриваючий, але з символом `/` перед ім'ям тега (закриваючі теги не використовують параметри).

Тег **<ACRONYM>** використовується для визначення абревіатур в поєднанні з параметром **TITLE**, в якості значення якого можна використовувати повну форму запису абревіатури, яка буде з'являтися у вигляді підказки.

Наприклад:

```
<ACRONYM TITLE="HyperText Markup Language"> HTML
</ACRONYM>
```


Тег **<CITE>** використовується для позначення цитат, назв статей або книг, посилань на інші джерела (виводиться курсивом).

Наприклад:

```
<CITE>HTML</CITE> - мова розмітки веб-сторінок.
```

Тег **** *текст* **** відображає текст напівжирним шрифтом.

Тег **<I>** *текст* **</I>** відображає текст курсивом.

Тег **<TT>** *текст* **</TT>** відображає текст моноширинним шрифтом.

Тег **<U>** *текст* **</U>** відображає текст підкресленим.

Тег **<S>** *текст* **</S>** відображає текст перекресленим горизонтальною лінією.

Тег **_{** *текст* **}** зміщує текст у нижній індекс.

Тег **^{** *текст* **}** зміщує текст у верхній індекс.

Тег **** вказує параметри шрифту. Параметри тега:

| Назва параметра | Опис дії параметра |
|-----------------|--|
| FACE | Вказує тип шрифту |
| SIZE | Вказує розмір шрифту в умовних одиницях від 1 до 7 |
| COLOR | Вказує колір шрифту. Може задаватися як за допомогою стандартних імен, так і в форматі #RRGGBB |

Наприклад:

```
<FONT SIZE=10 COLOR=red> текст </FONT>
```

1.5. Форматування абзаців

Тег **<P>***текст абзацу***</P>** використовується для створення абзацу (**</P>** не обов'язковий).

Тег **
** використовується для переведення, розриву рядка. Кінцевого тега немає.

Тег **<NOBR>***текст***<NOBR>** використовується для заборони розриву рядка.

Для оформлення заголовків всередині документа використовуються теги :

<H1>Заголовок розміру 1**</H1>**,

<H2>Заголовок розміру 2**</H2>**,

<H3>Заголовок розміру 3**</H3>**,

<H4>Заголовок розміру 4**</H4>**,

<H5>Заголовок розміру 5**</H5>**,

<H6> Заголовок розміру 6**</H6>**

(H1 – найбільший розмір, H6 – найменший).

Для горизонтального вирівнювання тексту використовується параметр

ALIGN в тегах **<P>** і **<H1>...<H6>**:

| Значення параметру ALIGN | Дія параметру |
|---------------------------------|--|
| LEFT | Вирівнювання тексту по лівій межі вікна |
| CENTER | Вирівнювання тексту по центру вікна |
| RIGHT | Вирівнювання тексту по правій межі вікна |
| JUSTIFY | Вирівнювання тексту по ширині вікна |

Для створення горизонтальної лінії використовується тег **<HR>**. Параметри:

| Параметри тега | Призначення |
|----------------|--|
| ALIGN | Вирівнює по краю чи центру (можливі значення: LEFT , CENTER , RIGHT) |
| WIDTH | Встановлює довжину лінії в пікселях чи відсотках від довжини вікна |
| SIZE | Встановлює товщину лінії в пікселях |
| NOSHADE | Відмінює рельєфність лінії |
| COLOR | Вказує колір лінії |

Для створення рухомого рядка використовується тег **<MARQUEE параметри>текст</MARQUEE>**. Параметри:

| Параметр | Призначення |
|--------------------------------|---|
| WIDTH | Ширина поля рухомого рядка в пікселях або відсотках від ширини вікна |
| HEIGHT | Висота поля рухомого рядка в пікселях |
| HSPACE
VSPACE | Інтервали по горизонталі і вертикалі між текстом рядка і краями його поля в пікселях |
| ALIGN | Положення тексту рухомого рядка в його полі. Можливі аргументи: TOP (зверху), BOTTOM (знизу), MIDDLE (посередині) |
| DIRECTION | Визначає напрямок руху рядка. Можливі аргументи: LEFT (справа наліво), RIGHT (зліва направо) |
| BEHAVIOR | Характер руху рядка. Можливі аргументи: SCROLL (текст з'являється з одного кінця і зникає за іншим), SLIDE (рядок витягується з одного кінця і зупиняється біля іншого), ALTERNATE (перемінний напрямок руху з одного кінця в інший) |
| LOOP | Кількість повторень тексту (число), для нескінченної кількості повторень LOOP=INFINITY |
| SCROOLAMOUNT | Встановлює довжину (в пікселях) „стрибка” тексту за один такт |
| SCROOLDELAY | Визначає величину паузи між тактами переміщення тексту в мілісекундах |
| BGCOLOR | Колір фону рухомого рядка |

1.6. Робота зі списками

Маркований список. Для створення маркованого списку необхідно використовувати тег-контейнер ` `, в середині якого розміщуються всі елементи списку. Кожен елемент списку повинен починатися тегом ``, який не потребує відповідного закриваючого тега. Для зміни виду маркерів в тезі `` можна використовувати параметр `type`, при чому:

type = disc – маркери відображаються зафарбованими кругами;

type = circle – маркери відображаються незафарбованими кругами;

type = square – маркери відображаються зафарбованими квадратами.

Наприклад:

```
<html>
<head>
<title>Маркований список</title>
</head>
```

```

<body>
  <ul type=circle>
    <b>Пункти:</b>
    <li>Перший пункт
    <li>Другий пункт
    <li>Третій пункт
  </ul>
</body>
</html>

```

Також, в якості маркерів списку можна використовувати графічні зображення. В такому випадку можна обійтися без тегів ``. Достатньо лише перед кожним елементом списку вставити бажане графічне зображення. Єдиною задачею, яку необхідно буде розв'язати, буде відділення елементів списку один від одного. Для цього можна використати теги абзацу `<p>` або вимушеного переведення рядка `
`.

Наприклад:

```

<html>
<head>
<title>Маркований список</title>
</head>

<body>
  <ul> <b>Пункти:</b>
    Пункт 1<br>
    Пункт 2<br>
    Пункт 3<br>
  </ul>
</body>
</html>

```

Нумерований список. Для створення нумерованого списку необхідно використовувати тег-контейнер ` `, в середині якого розміщені всі елементи списку. В нумерованому списку перед кожним його елементом автоматично проставляється порядковий номер. Кожен елемент нумерованого списку повинен починатися з тега ``.

В тезі `` можуть бути вказані параметри `type` і `start`. Параметр `type` використовується для означення виду нумерації списку:

type = A – задає маркери у вигляді великих латинських літер;

type = a – задає маркери у вигляді малих латинських літер;

type = I – задає маркери у вигляді великих римських літер;

type = 1 – задає маркери у вигляді арабських цифр.

За замовчуванням завжди використовується значення `type = 1`.

Параметр `start` тега `` дозволяє почати нумерацію списку не з одиниці.

Наприклад:

```
<html>
<head>
<title>Нумерований список</title>
</head>

<body>
  <ol start=3>
    <b>Пункти:</b>
    <li>Пункт 3
    <li>Пункт 4
    <li>Пункт 5
  </ol>
</body>
</html>
```

1.7. Форматування таблиць

Елементи таблиці. Таблиці будуються за принципом вкладення. Кожна таблиця починається тегом **<TABLE>** і закінчується тегом **</TABLE>**. Таблицю будують по рядках, а рядки заповнюються комірками за допомогою таких елементів:

<TR> </TR> – рядок таблиці;

<TD> </TD> – комірки даних;

<TH> </TH> – комірка заголовка.

Таблиця може мати і заголовок, обмежений парою тегів **<CAPTION>** і **</CAPTION>**, які мають єдиний допустимий параметр **align**, що визначає розміщення заголовку, при чому:

ALIGN=TOP – заголовок над таблицею (за замовчуванням);

ALIGN=BOTTOM – заголовок під таблицею.

Параметри тега <TABLE>:

BORDER – вивід зовнішньої рамки таблиці в пікселях (за замовчуванням **BORDER=1**).

CELLSPACING – визначає відстань в пікселях між рамками сусідніх комірок як по горизонталі, так і по вертикалі (за замовчуванням **CELLSPACING =2**).

CELLPADDING – визначає розмір вільного місця (відступу) в пікселях між рамкою комірки і даними всередині комірки (за замовчуванням **CELLPADDING=1**).

ALIGN – задає вирівнювання таблиці у документі. Допустимі значання:

ALIGN = LEFT – розміщення таблиці уздовж лівого краю документа;

ALIGN = RIGHT – уздовж правого краю документа;

ALIGN = CENTER – по центру документа.

HEIGHT – задає висоту таблиці в пікселях.

WIDTH – задає ширину таблиці в пікселях чи процентах (від повної ширини вікна).

Атрибути елементів рядків і стовпців:

WIDTH і **HEIGHT** – встановлюють розміри комірок рядка: ширину і висоту (можуть використовуватися в тегах <TD>, <TH>, <TR>).

ALIGN – вирівнює вміст в комірках. Атрибут може набувати значень **LEFT**, **RIGHT**, **CENTER**, **JUSTIFY**; (можуть використовуватися в тегах <TD>, <TH>, <TR>).

VALIGN – вирівнює вміст по вертикалі. Можливі значення: **TOP** (вирівнювання по верхньому краю комірок), **BOTTOM** (вирівнювання по нижньому краю), **MIDDLE** (центрування по вертикалі).

Колір у таблицях. Атрибути, що визначають колір таблиць:

BGCOLOR – колір фону в таблиці (може використовуватися в тегах <TABLE>, <TD>, <TH>, <TR>).

BORDERCOLOR – колір рамок таблиці (може використовуватися в тегах <TABLE>, <TD>, <TH>, <TR>).

Об'єднання комірок таблиці. Для об'єднання комірок в початкових тегах <TD> або <TH> застосовуються такі атрибути:

ROWSPAN – об'єднує комірки суміжних рядків. Значення атрибута задає кількість об'єднаних комірок;

COLSPAN – об'єднує комірки суміжних стовпців.

Приклад HTML-коду, що реалізує об'єднання комірок в такій таблиці:

| | | |
|--------------------------------|----------------------------------|-----------|
| Комірка, що об'єднує два рядка | Комірка, що об'єднує два стовпця | |
| | Комірка 3 | Комірка 4 |
| Комірка 5 | Комірка 6 | Комірка 7 |

```
<HTML>
<HEAD>
<TITLE> Об'єднання комірок </TITLE>
</HEAD>

<BODY>
  <TABLE>
    <TR>
      <TD ROWSPAN=2>Комірка, що об'єднує два рядки </TD>
      <TD COLSPAN=2>Комірка, що об'єднує два стовпці
    </TD>
    </TR>
    <TR>
      <TD> Комірка 3</TD>
      <TD>Комірка 4</TD>
    </TR>
    <TR>
      <TD> Комірка 5</TD>
      <TD>Комірка 6</TD>
      <TD>Комірка 7</TD>
    </TR>
  </TABLE>
</BODY>
```

</HTML>

1.8. Встановлення кольорів та зображень

Колір фону сторінки і фонове зображення. Для визначення кольору фону чи встановлення фонового зображення, необхідно в тегу <BODY> визначити такі атрибути: **BGCOLOR** (колір фону) і **BACKGROUNDIMAGE** (фонове зображення). Атрибут **BGPROPERTIES = FIXED** забороняє прокручування фонового зображення.

Зображення. Для додавання зображень слід використовувати тег , всередині якого обов'язково записується атрибут **SRC**, що вміщує адресу зображення. Даний тег може мати ряд параметрів:

ALIGN – призначений для вирівнювання зображення на сторінці. Значення параметра є такими:

| Значення | Дія параметра |
|-----------------------------------|--|
| TOP | Верхня межа зображення вирівнюється по найвищому елементу поточного рядка |
| TEXTTOP | Верхня межа зображення вирівнюється по найвищому текстовому елементу поточного рядка |
| MIDDLE | Вирівнювання середини зображення по базовій лінії поточного рядка |
| ABSMIDDLE | Вирівнювання середини зображення посередині поточного рядка |
| BASELINE або BOTTOM | Вирівнювання нижньої межі зображення по базовій лінії поточного рядка |
| ABSBOTTOM | Вирівнювання нижньої межі зображення по нижній межі поточного рядка |
| LEFT | Зображення притискається до лівого поля вікна. Текст обтікає зображення з правого боку |
| RIGHT | Зображення притискається до правого поля вікна. Текст обтікає зображення з лівого боку |

Параметри **WIDTH** (ширина) **HEIGHT** (висота) задають розміри зображення в пікселях або відсотках.

Параметри **HSPACE** (відступ по горизонталі) **VSPACE** (відступ по вертикалі) задають відступ від тексту до зображення в пікселях.

BORDER – визначає товщину рамки в пікселях, в яку буде поміщене зображення.

1.9. Гіперпосилання в документі

Всі гіперпосилання задаються тегом `<A HREF>...`. Розрізняють такі види гіперпосилань:

– текстові посилання: ` текст посилання `

– графічні посилання: ` `

– посилання в межах одного документа (закладки): для закладок потрібні якір і власне посилання. Якір визначає місце в документі, на яке відбувається перехід через посилання. Посилання використовує ім'я якоря замість імені (адреси) файлу.

Якір задається так: ` текст на екрані `

Посилання: ` текст на якому клацати `

2. КАСКАДНІ ТАБЛИЦІ СТИЛІВ CSS

2.1. Поняття про CSS

Каскадні таблиці стилів (англ. *Cascading Style Sheets* або скорочено **CSS**) — спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки – розділення змісту сторінки (даних) та їхньої візуальної презентації.

CSS використовується авторами та відвідувачами веб-сторінок для того, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг – можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документа (що описується в CSS).

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою (КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерів та ін.)

Один і той самий HTML або XML документ може бути відображений по-різному залежно від використаного CSS. Стилі для відображення сторінки можуть бути такими:

– стилі автора (інформація надана автором сторінки):

зовнішні таблиці стилів (англ. *stylesheet*), найчастіше окремий файл або файли з розширенням.css;

внутрішні таблиці стилів, включені як частина документа або блоку;
стили для окремого елемента;

– стилі користувача – локальний .css-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера;

– стилі переглядача (браузера) – стандартний стиль переглядача, наприклад стандартні стилі для елементів, визначені браузером, використовуються тоді, коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок та діапазон застосування стилів, те, в якій послідовності і для яких елементів застосовуються стилі. Таким чином, використовується принцип *каскадності*, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями.

Переваги CSS:

– інформація про стиль для усього сайту або його частин може міститися в одному .css-файлі, що дозволяє швидко робити зміни в дизайні та презентації сторінок;

– різна інформація про стилі для різних типів користувачів: наприклад великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на принтер, стиль для мобільних пристроїв;

– сторінки зменшуються в об'ємі та стають більш структурованими, оскільки інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з їх урахуванням;

– прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузери здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної.

2.2. Синтаксис CSS та його застосування

Фундаментальна модель CSS є такою:

```
selector {
    property1: value;
    property2: value;
    ...
}
```

Тут:

`selector` – спеціальне позначення, що визначає діапазон застосування правила;

`property` – назва властивості елемента;

`value` – значення властивості.

Треба відмітити, що властивість елемента та значення властивості має розділяти двокрапка, правила для різних властивостей мають бути розділені крапкою з комою.

Методи застосування CSS в HTML- документі. Для того, щоб включити таблицю стилів CSS у HTML документ можливим є використання таких способів:

– *in-line метод (атрибут style):*

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body style="background-color: #FF0000;">
    <p>This is a red page</p>
  </body>
</html>
```

– *внутрішній (meg style):*

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

```

</head>
<body>
  <p>This is a red page</p>
</body>
</html>

```

– зовнішній (посилання на таблицю стилів – файл *.css*):

```

<html>
  <head>
    <title>My document</title>
    <link rel="stylesheet" type="text/css"
href="style/style.css" />
  </head>
  <body>
    ...

```

2.3. Управління кольором та фоном

Властивості, що використовуються для управління кольором та фоном є такими:

1. Колір переднього плану – **'color'**
2. Колір фону елемента – **'background-color'**
3. Фонове зображення – **'background-image'**

Наприклад:

```

body {
  background-color: #FFCC66;
  background-image: url("butterfly.gif");
                   або url("../images/butterfly.gif")
}

h1 {
  color: #990000;
  background-color: #FC9804;
}

```

4. Повторення фонового зображення на сторінці – **'background-repeat'**. Властивість має такі значення:

| Значення | Опис |
|------------------|---|
| repeat-x | Рисунок повторюється по горизонталі |
| repeat-y | Рисунок повторюється по вертикалі |
| repeat | Рисунок повторюється і по горизонталі, і по вертикалі |
| no-repeat | Рисунок не повторюється |

Наприклад:

```
body {
  background-color: #FFCC66;
  background-image: url("butterfly.gif");
  background-repeat: no-repeat;
}

h1 {
  color: #990000;
  background-color: #FC9804;
}
```

5. Блокування фонового зображення – '**background-attachment**' (визначає, чи рисунок на сторінці є зафіксованим, чи має прокручуватися разом із сторінкою). Значення властивості можуть бути такими:

| Значення | Опис |
|---------------|---|
| scroll | Зображення прокручується разом зі сторінкою |
| fixed | Зображення заблоковано |

Наприклад:

```
body {
  background-color: #FFCC66;
  background-image: url("butterfly.gif");
  background-repeat: no-repeat;
  background-attachment: fixed;
}

h1 {
  color: #990000;
  background-color: #FC9804;
}
```

6. Положення фонового рисунку – '**background-position**': (дозволяє вказувати координати рисунку на сторінці у відсотках ширини екрану,

в фіксованих одиницях (пікселях, сантиметрах, тощо) або ж "прив'язувати" рисунок до конкретної частини сторінки (*top, bottom, center, left, right*). Задавати значення властивості можна такими способами:

| Значення | Опис |
|---|---|
| <code>background-position: 2cm 2cm</code> | Рисунок розміщений на 2 см зліва та на 2 см зверху |
| <code>background-position: 50% 25%</code> | Рисунок розміщений по центру, на четверть екрану зверху |
| <code>background-position: top right</code> | Рисунок розміщений в правому верхньому куточку сторінки |

2.4. Форматування шрифтів

Для форматування шрифтів доступними є такі властивості%

1. Сімейство шрифту – '**font-family**' (вказує пріоритетний список шрифтів, що використовується при відображенні елемента чи сторінки. Шрифт може відображатися тільки за умови, якщо він встановлений на комп'ютері клієнта)

Деякі типи шрифтів:

Times New Roman

Garamond

Georgia

шрифти сімейства serif – шрифти з "засічками"

Trebuchet

Arial

Verdana

шрифти сімейства sans-serif (без "засічок")

Courier

Courier New

Andale Mono

шрифти сімейства monospace – шрифти з однаковою шириною символів

Наприклад:

```
h1 {
  font-family: arial, verdana, sans-serif;
}
h2 {
  font-family: "Times New Roman", serif;
}
```

2. Стиль шрифту – '**font-style**' (визначає `normal`, `italic` або `oblique`)

Наприклад:

```
h1 {
  font-family: arial, verdana, sans-serif;
}
h2 {
  font-family: "Times New Roman", serif;
  font-style: italic;
}
```

3. Варіант шрифту – '**font-variant**' (вибирає між `normal` та `small-caps`)

Наприклад:

```
h1 {
  font-variant: small-caps;
}
h2 {
  font-variant: normal;
}
```

4. Вага шрифту – '**font-weight**' (описує на скільки товстим має бути шрифт; шрифт може бути `normal` або `bold`)

Наприклад:

```
p {
  font-family: arial, verdana, sans-serif;
}
td {
  font-family: arial, verdana, sans-serif;
  font-weight: bold;
}
```


5. Розмір шрифту – 'font-size'

Наприклад:

```
h1 {font-size: 30px;}
h2 {font-size: 12pt;}
h3 {font-size: 120%;}
p {font-size: 1em;}
```

6. Скорочений запис – 'font' (порядок властивостей: Font-style | font-variant | font-weight | font-size | font-family)

Наприклад:

```
p {
  font-style: italic;
  font-weight: bold;
  font-size: 30px;
  font-family: arial, sans-serif;
}
або
p { font: italic bold 30px arial, sans-serif; }
```

2.5. Форматування тексту

Властивості для форматування тексту є такими:

1. Відступи – 'text-indent'

Наприклад:

```
p { text-indent: 30px; }
```

2. Вирівнювання тексту – 'text-align' (left, right, centered, justify)

Наприклад:

```
th { text-align: right; }
td { text-align: center; }
p { text-align: justify; }
```

3. Декор – 'text-decoration'

Наприклад:

```
h1 {text-decoration: underline;}
h2 {text-decoration: overline;}
h3 {text-decoration: line-through;}
```

4. Інтервал між буквами – '**letter-spacing**'

Наприклад:

```
h1 { letter-spacing: 6px;}
p { letter-spacing: 3px;}
```

5. Трансформація регістру – '**text-transform**' (capitalize, uppercase, lowercase, none)

Наприклад:

```
h1 {text-transform: uppercase;}
li {text-transform: capitalize;}
```

2.6. Форматування гіперпосилань

Для форматування лінків використовують псевдокласи, які дозволяють враховувати різні події чи умови при визначенні властивостей тега.

Використовуються такі псевдокласи:

```
a:link | a:visited | a:active | a:hover
```

1. Псевдоклас **a:link** використовується для лінків, які користувач ще не відвідував.

2. Псевдоклас **a:visited** використовується для посилань на сторінки, які користувач вже відвідав.

3. Псевдоклас **a:active** використовується для активних посилань.

4. Псевдоклас **a:hover** використовується для посилань, на які наведений вказівник мишки.

Наприклад:

```
a:hover {
  text-transform: uppercase;
  font-weight:bold;
  color:blue;
  background-color:yellow;
}
```

2.7. Селектори атрибутів

Застосовуються такі селектори атрибутів:

1. *Простий селектор атрибуту* – встановлює стиль для елемента, якщо заданий специфічний атрибут тега. Значення атрибуту неважливе.

Синтаксис:

```
[атрибут] {
    властивість1: значення;
    властивість2:значення;...
}
Селектор[атрибут]{
    властивість1: значення;
    властивість2:значення;...
}
```

Пробіл між іменем селектора та квадратними дужками **не допускається**.

Приклад:

```
<html>
<head>
<title>Поезія</title>
<style>
  q {
    font-style:italic;
    font-size:14px;
    color:#3333CC;
  }
  q[title]{
    font-style:oblique;
    font-size:16px;
    color:#CC0033;
    text-align:left;
  }
</style>
</head>
<body>
<p>
<q> Бувають хвили – серце мліє <br />
  І скорбних мислей рій летить, <br />
  Мов чорна хмара небо криє <br />
```

```

    І грім у хмарі гуркотить. <br />
</q>
<q title="Бувають хвилі..."> Іван Франко </q>
</p>
</body>
</html>

```

2. *Селектор атрибуту зі значенням* – встановлює стиль для елемента, у якого задане конкретне значення специфічного атрибуту.

Синтаксис:

[атрибут="значення">{опис стилю}

Селектор[атрибут="значення">{опис стилю}

Приклад:

```

<html>
<head>
<title>Поезія</title>
<style>
  td[class="name"] {
    font-family:Geneva,Arial,Helvetica, sans-serif;
    font-style:oblique;
    font-size:16px;
    color:#FF9933;
  }
</style>
</head>

<body>
<table border="2">
<tr>
<th>Прізвище автора</th>
<th>Назва твору </th>
</tr>
<tr>
<td class="name"> Шевченко Т.Г.</td>
<td> Заповіт </td>
</tr>
<tr>
<td class="name"> Франко І.</td>
<td> Каменярі </td>
</tr>
</table>
</body>

```

```
</html>
```

3. Селектор атрибуту, значення якого починається з деякого тексту – встановлює стиль для елемента в тому випадку, коли значення атрибуту починається з певного тексту.

Синтаксис:

```
[атрибут^="значення">{опис стилю}
```

```
Селектор[атрибут^="значення">{опис стилю}
```

Приклад:

```
<html>
  <head>
    <title>Гіперпосилання</title>
    <style>
      A[href^="http://"] {
        font-weight: bold;
      }
    </style>
  </head>

  <body>
    <p><a href="1.html">Звичайне посилання</a> <br>
    <a href="http://htmlbook.ru">Зовнішнє посилання
</a>
  </p>
</body>
</html>
```

4. Селектор атрибуту, значення якого закінчується певним текстом.

Синтаксис:

```
[атрибут$="значення"]{
```

```
  опис стилю
```

```
}
```

```
Селектор[атрибут$="значення"]{
```

```
  опис стилю
```

```
}
```

Приклад:

```
<html>
<head>
<title>Селектори атрибутів </title>
  <style>
    A[href$=".ua"] {
      color:#33CC33;
    }
    A[href$=".com"] {
      color:#FF99FF;
    }
  </style>
</head>

<body>
<p>
  <a href="http://www.gmail.com">Gmail.com</a> |
  <a href="http://www.i.ua">I.ua</a></p>
</body>
</html>
```

5. Селектор атрибуту, значення якого містить певний текст.

Синтаксис:

[атрибут="значення"]{опис стилю}*

Селектор[атрибут="значення"]{опис стилю}*

Приклад:

```
<html>
<head>
  <title>Селектори атрибутів</title>
  <style>
    [href*="google"] {
      background: yellow;
    }
  </style>
</head>

<body>
<p>
  <a href="http://www.google.com.ua">
    Пошукова система Google
  </a> |
  <a href="http://www.i.ua/">
```

```

        Інформаційний ресурс I.ua
    </a>
</p>
</body>
</html>

```

6. Селектор атрибуту з одним із декількох значень

Синтаксис:

```

    [атрибут~="значення"]{опис стилю}
    Селектор[атрибут~="значення"]{опис стилю}

```

Приклад:

```

<html>
<head>
<title>Блок</title>
  <style>
    [class~="block"]{
      color: green;
    }
  </style>
</head>

<body>
  <div class="block tag">
    <h3>Заголовок</h3>
  </div>
</body>
</html>

```

Всі перераховані методи можна комбінувати між собою, визначаючи стилі для елементів, які містять два і більше атрибутів. В таких випадках квадратні дужки ідуть підряд.

Синтаксис:

```

    [атрибут1="значення1"][атрибут2="значення2"]{опис стилю}
    Селектор[атрибут1="значення1"][атрибут2="значення2"]{опис стилю}

```

2.8. Класи

Класи застосовуються у випадку, коли необхідно визначити стиль для індивідуального елемента веб-сторінки або задати різні стилі для одного тега.

Синтаксис:

```

Тег.ім'я класу {
    властивість1: значення;
    властивість2: значення;...
}

```

Для того, щоб вказати в HTML кодї те, що тег використовується з відповідним класом, необхідно до тега додати атрибут `class="Ім'я класу"`.

Приклад:

```

<html>
<head>
  <title>Класи</title>
  <style>
    P {
      color:#009933;
    }
    P.name {
      color: #CC0066;
      font-family:Arial, Helvetica, sans-serif;
      border-left: 1px solid navy;
    }
  </style>
</head>

<body>
  <p>
Красива осінь вишиває клени <br />
Червоним, жовтим, срібним, золотим. <br />
А листя просить: - Виший нас зеленим! <br />
Ми ще побудем, ще не облетим. <br />
<br />
А листя просить: - Дай нам тої втіхи! <br />
Сади прекрасні, роси - як вино. <br />
Ворони п'ють надкльовані горіхи. <br />
А що ім, чорним? Чорним все одно. <br />
  <p class="name"> Ліна Костенко</p>
  </body>
</html>

```

Можливим є також використання імені класу без зазначення тега.

Синтаксис:

```
.ім'я класу {
    властивість1: значення;
    властивість2: значення; ...
}
```

В цьому випадку клас можна застосовувати для будь-якого тега.

Приклад 1:

```
<html>
<head>
  <title>Класи</title>
  <style>
    P {
      color:#009933;
    }
    .title {
      font-style:italic;
      color:#FF0000;
    }
    .name {
      color: #CC0066;
      font-family:Arial, Helvetica, sans-serif;
      border-left: 1px solid navy;
    }
  </style>
</head>

<body>
  <p>
    <b class="title">
Місто, премісто, прамісто моє! </b> <br />
Стійбище людське з асфальту й бетону.<br />
Як там не буде, а все-таки є<br />
той силует у вікні золотому!<br />
<br />
Ось я проходжу, і ось я пройду.<br />
Може, й навік, але справа не в тому -<br />
Тільки б не танув, як тінь на льоду, <br />
той силует у вікні золотому!<br />
<br />
Скільки душа прориває тенет!<br />
З моря і суші - додому, додому!<br />
```

```

Там, де у мене є той силует,<br />
той силует у вікні золотому! <br />
<br />
  <p class="name"> Ліна Костенко</p>
</body>
</html>

```

Приклад 2:

```

<html>
<head>
  <title>Міста «Мільйонери» </title>
  <style>
    table.country {
      width: auto;
      border: thick;
    }
    th {
      background: yellow;
      color: blue;
      text-align: center;
    }
    .odd {
      background: #FFFF99;
    }
  </style>
</head>

<body>
  <table class="country">
    <tr>
      <th>Місто</th>
      <th>Область</th>
      <th>Чисельність населення, <br /> 2014</th>
    </tr>
    <tr class="odd">
      <td>Київ</td>
      <td>м.Київ</td>
      <td>2 868 702</td>
    </tr>
    <tr>
      <td>Харків</td>
      <td>Харківська</td>
      <td>1 470 902</td>
    </tr>
    <tr class="odd">

```

```

        <td>Одеса</td>
        <td>Одеська</td>
        <td>1 029 049</td>
    </tr>
    <tr>
    </table>
</body>
</html>

```

До будь-якого тега можна застосувати декілька класів, перераховуючи їх в атрибуті `class` через пробіл. В такому випадку до елемента застосовується стиль, описаний в правилах для кожного класу. Пріоритет мають значення класу, який описаний в коді найнижче.

Приклад:

```

<html>
<head>
  <title>Degrees of comparasion</title>
  <style type="text/css">
    .level1 {
      font-size: 1em;
      color:#000000;
    }
    .level2 {
      font-size: 1.2em;
      color:#0000CC;
    }
    .level3 {
      font-size: 1.4em;
      color:#006666;
    }
    .level4 {
      font-size: 1.6em;
      color:#996600;
    }
    .level5 {
      font-size: 1.8em;
      color:#CC9900;
    }
    .level6 {
      font-size: 2em;
      color:#FF0000;
    }
    .tag {

```

```

        font-family:Geneva, Arial, Helvetica, sans-
serif;
    }
</style>
</head>

<body>
<div>
    <p class="tag level1">the worst</p>
    <p class="tag level2">worse</p>
    <p class="tag level3">bed</p>
    <p class="tag level4">good</p>
    <p class="tag level5">better</p>
    <p class="tag level6">the best </p>
</div>
</body>
</html>

```

2.9. Ідентифікатори

Ідентифікатор визначає унікальне ім'я елемента, яке використовується для зміни його стиля та звернення до нього через скрипт.

Синтаксис:

```

#Імя ідентифікатора {
    властивість1: значення;
    властивість2:значення;...
}

```

Важливим є те, що ідентифікатори, на відміну від класів, мають бути унікальними. Вони вказуються в атрибуті тегів `id`.

Приклад:

```

<html>
<head>
    <title>Ідентифікатори</title>
    <style>
        #poem {
            position: absolute;
            left: 160px;
            top: 50px;
            width: 230px;

```

```

        padding: 10px;
        background: yellow;
        color: blue;
    }
</style>
</head>

<body>
  <div id="поем">
    Життя пройде, немов вода, <br />
    і відцвіте, немов вишнева гілка... <br />
    в житті одна помилка - не біда, <br />
    біда, коли усе життя - помилка.
  </div>
</body>
</html>

```

Допустимим також є застосування ідентифікатора до конкретного тега.

Синтаксис:

```

Тег#Імя ідентифікатора {
    властивість1: значення;
    властивість2: значення;...
}

```

Приклад:

```

<html>
<head>
  <title>Ідентифікатори</title>
  <style>
    P {
      color: red;
      font-style: oblique;
      border: double;
      width: 600px;
    }
    P#opa {
      color: green;
      border: hidden;
      background: #eee;
      padding: 5px;
      font-style: normal;
      width: 150px;
    }
  </style>
</head>
<body>
  <p>Привіт!</p>
  <p id="opa">Привіт!</p>
</body>
</html>

```

```

        position:absolute;
        left:450px;
    }
</style>
</head>

<body>
    <p>Неважливо як ти троянду назвеш, Адже пахнути
вона буде тільки трояндою.
    </p>
    <p id="ора"> "Ромео і Джульєтта"</p>
</body>
</html>

```

2.10. Взаємне розміщення селекторів

Контекстні селектори. У випадку, коли одні теги вкладені в інші, можливим є застосування контекстних селекторів, які дозволяють встановлювати стилі для тегів, які містяться в середині інших тегів.

Синтаксис:

```

Тег1 Тег2 {
    властивість1: значення;
    властивість2:значення;...
}

```

В такому випадку стиль буде застосований до Тега2, що знаходиться в Тегу1. Наприклад:

```

<Тег1>
    <Тег2> ... </Тег2>
</Тег1>

```

Приклад:

```

<html>
<head>
    <title>Контекстні селектори</title>
    <style>
        P B {
            font-family: Times, serif;
            color: red;

```

```

    }
  </style>
</head>
<body>
  <div><b>It does not matter how slowly you go
    so long as you do not stop
  </b></div>
  <p><b>Не так важливо, з якою швидкістю
    рухатися в потрібному напрямку,
    набагато важливіше не робити зупинок.
  </b></p>
</body>
</html>

```

Допустимим є застосування двох чи більше послідовних вкладень тегів.

Іншим способом використання контекстних селекторів є зазначення ідентифікаторів та класів. Це дозволяє встановлювати стиль для того елемента, який розташований всередині класу.

Приклад:

```

<html>
<head>
  <title>Контекстні селектори</title>
  <style>
    A {
      color: #000066;
    }
    .menu {
      padding: 7px;
      border: 1px solid #333;
      background: #fc0;
      width:500px;
    }
    .menu A {
      color: #FF0000;
    }
  </style>
</head>

<body>
  <div class="menu">
    Кафедри математичного факультету: <br />
    <a href="http://">Кафедра алгебри</a> <br />

```

```

<a href="http://">Кафедра диференціальних
    рівнянь та математичної фізики</a> <br />
<a href="http://">Кафедра кібернетики і
    прикладної математики</a> <br />
<a href="http://">Кафедра системного аналізу
    та теорії оптимізації</a> <br />
<a href="http://">Кафедра теорії ймовірностей
    і математичного аналізу</a>
</div>
<p><a href="http://">Інші факультети</a></p>
</body>
</html>

```

Сусідні селектори. Сусідніми називаються елементи сторінки, які слі-
дують безпосередньо один за одним в коді документа.

Синтаксис:

```

Селектор1 + Селектор2 {
    властивість1:значення;
    властивість2:значення; ...
}

```

При цьому стиль застосовується до Селектора2, але тільки в тому ви-
падку, коли він слідує безпосередньо після Селектора1.

Приклад 1:

```

<html>
<head>
  <title>Сусідні селектори</title>
  <style>
    B + I {
      color: red;
      font-family:Geneva, Arial, Helvetica, sans-
serif;
      text-decoration:underline;
    }
  </style>
</head>

<body>
  <p><b>Життя</b> - це не пошук себе. <i>Життя</i>
- це створення себе.</p>

```



```

    <p>Життя - це не пошук себе. <i>Життя</i> - це
створення себе.</p>
  </body>
</html>

```

Приклад 2:

```

<html>
<head>
  <title>Стиль абзацу</title>
  <style>
    h1 {
      color:#FF0000;
      text-align:center;
    }
    H2.sic {
      font-size: 140%;
      color: maroon;
      font-weight: normal;
      margin-left: 30px;
      margin-bottom: 0px;
    }
    p {
      font-style:italic;
    }
    H2.sic + P {
      background: #ddd;
      margin-left: 30px;
      margin-top: 0.5em;
      padding: 7px;
      text-align:justify;
    }
  </style>
</head>

<body>
  <h1>Конституція України</h1>
  <h2>Преамбула</h2>
  <p>
    Верховна Рада України від імені Українського на-
    роду - громадян України всіх національностей, <br />
    виражаючи суверенну волю народу, <br />
    спираючись на багатовікову історію українського
    державотворення і на основі здійсненого українською на-
    цією, усім Українським народом права на самовизначення,
    <br />

```

```

    дбаючи про забезпечення прав і свобод людини та
    гідних умов її життя, <br />
    піклуючись про зміцнення громадянської злагоди на
    землі України, <br />
    прагнучи розвивати і зміцнювати демократичну, со-
    ціальну, правову державу, <br />
    усвідомлюючи відповідальність перед Богом, влас-
    ною совістю, попередніми, нинішнім та майбутніми поко-
    ліннями, <br />
    керуючись Актом проголошення незалежності України
    від 24 серпня 1991 року, схваленим 1 грудня 1991 року
    всенародним голосуванням, <br />
    приймає цю Конституцію - Основний Закон України.
<br />
</p>
<h2 class="sic">Стаття 53. Кожен має право на
освіту</h2>
<p>
    Повна загальна середня освіта є обов'язковою.
<br />
    Держава забезпечує доступність і безоплатність
    дошкільної, повної загальної середньої, професійно-
    технічної, вищої освіти в державних і комунальних на-
    вчальних закладах; розвиток дошкільної, повної загаль-
    ної середньої, позашкільної, професійно-технічної, ви-
    щої і післядипломної освіти, різних форм навчання; на-
    дання державних стипендій та пільг учням і студентам.
<br />
    Громадяни мають право безоплатно здобути вищу осві-
    ту в державних і комунальних навчальних закладах на
    конкурсній основі.<br />
    Громадянам, які належать до національних меншин,
    відповідно до закону гарантується право на навчання рід-
    ною мовою чи на вивчення рідної мови у державних і ко-
    мунальних навчальних закладах або через національні ку-
    льтурні товариства.</p>
</body>
</html>

```

Дочірні селектори. Дочірнім є елемент, який **безпосередньо** розміщується всередині батьківського елемента.

Синтаксис:

```

Селектор1 > Селектор2 {
    властивість1:значення;
    властивість2:значення;...
}

```

Стиль застосовується для Селектора2, але лише в тому випадку, коли він є дочірнім для Селектора1.

Дочірні селектори схожі на контекстні з тою лише різницею, що стиль до дочірнього селектора застосовується лише в тому випадку, коли він є прямим нащадком батьківського селектора, а для контекстних селекторів допускається будь-який рівень вкладення.

Приклад 1:

```

<html>
<head>
  <title>Дочірні селектори</title>
  <style>
    DIV I {
      color: green;
    }
    P {
      background-color:#FFFF33;
      text-align:center;
      font-size:18px;
    }
    P > I {
      color: red;
    }
  </style>
</head>

<body>
  <div>
    <p>Don't trouble
      <i> trouble </i>
      until
      <i>trouble</i>
      troubles you.
    </p>
  </div>

```

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

Приклад 2:

```
<html>  
<head>  
  <title>Дочірні селектори</title>  
  <style>  
    UL#menu {  
      margin: 0;  
      padding: 0;  
    }  
    UL#menu > LI {  
      list-style: none;  
      width: 100px;  
      background: #b3d9d2;  
      color: #333;  
      padding: 5px;  
      font-family: Arial, sans-serif;  
      font-size: 90%;  
      font-weight: bold;  
      float: left;  
    }  
    LI > UL {  
      list-style: none;  
      margin: 0;  
      padding: 0;  
      border-bottom: 1px solid #666;  
      padding-top: 5px;  
    }  
    LI > A {  
      display: block;  
      font-weight: normal;  
      font-size: 90%;  
      background: #fff;  
      border: 1px solid #666;  
      border-bottom: none;  
      padding: 5px;  
    }  
  </style>  
</head>  
  
<body>  
  <ul id="menu">  
    <li>Правка
```

```

    <ul>
      <li><a href="#">Отменить</a></li>
      <li><a href="#">Вырезать</a></li>
      <li><a href="#">Копировать</a></li>
      <li><a href="#">Вставить</a></li>
    </ul>
  </li>
  <li>Начертание
    <ul>
      <li><a href="#">Жирное</a></li>
      <li><a href="#">Курсивное</a></li>
      <li><a href="#">Подчеркнутое</a></li>
    </ul>
  </li>
  <li>Размер
    <ul>
      <li><a href="#">Маленький</a></li>
      <li><a href="#">Нормальный</a></li>
      <li><a href="#">Средний</a></li>
      <li><a href="#">Большой</a></li>
    </ul>
  </li>
</ul>
</body>
</html>

```

2.11. Універсальний селектор

У випадку, коли необхідно встановити одночасно стиль для всіх елементів веб-сторінки, необхідно використовувати універсальний селектор, який відповідає будь-якому елементу сторінки.

Синтаксис:

```

* {
  опис правил стилю
}

```

Приклад:

```

<html>
<head>
  <title>Універсальний селектор</title>
  <style>

```

```

* {
  font-family: "Courier New", Courier, monospace;
  font-size: 24px;
  color:#CC0033;
}
</style>
</head>

<body>
  <p>We live on the planet that is called the
Earth. <br />
  The greatest part of it is taken by the waters of
the World Ocean. Large masses of land surrounded by
oceans and seas are called mainlands or continents.
They are like big islands. Looking at the map of the
world we can see that there are six continents. They
are: Eurasia (Europe and Asia combined) which is the
largest continent, Africa, North America, South Ameri-
ca, Australia and Antarctica, in which the South Pole
is situated.</p>
</body>
</html>

```

2.12. Псевдоелементи

Псевдоелементи дозволяють задавати стилі елементів, не визначених в дереві елементів сторінки, а також генерувати вміст, якого немає в початковому коді тексту.

Синтаксис:

```

Селектор:псевдоелемент {
  опис правил стилю
}

```

Кожен псевдоелемент може бути застосованим тільки до одного селектора.

1. :after – застосовується для вставки контенту після вмісту елемента. Цей псевдоелемент працює разом з властивістю стиля `content`, яка визначає вміст для вставки.

Приклад:

```

<!DOCTYPE html>
<html>
<head>
  <title>Псевдоелементи</title>
  <style>
    P.new:after {
      content: " - Mark Twain ";
      color:#CC3366;
      font-family:Geneva, Arial, Helvetica, sans-
serif;
      font-style:italic;
    }
  </style>
</head>

<body>
  <p class="new"> Give every day the chance to be-
come the most beautiful day of your life.</p>
</body>
</html>

```

2. :before – застосовується для вставки контенту до вмісту елемента.

Приклад:

```

<!DOCTYPE HTML>
<html>
<head>
  <title>Псевдоелементи</title>
  <style>
    UL {
      padding-left: 0;
      list-style-type: none;
    }
    LI:before {
      content: "\20aa ";
      color: blue;
    }
  </style>
</head>

<body>
  <ul>
    <li>Чебурашка</li>

```

```

    <li>Крокодил Гена</li>
    <li>Шапокляк</li>
    <li>Крыса Лариса</li>
  </ul>
</body>
</html>

```

При використанні зазначених псевдоелементів необхідно враховувати такі особливості:

- `:after` і `:before` не працюють у браузері ІЕ до 7-ої версії;
- `:after` і `:before` дають результат тільки для тих елементів, у яких є вміст, тому їх додавання їх, наприклад, до селектора `img` не дає результату.

3. `:first-letter` – визначає стиль першого символу в тексті елемента.

Приклад:

```

<!DOCTYPE html>
<html>
<head>
  <title>Псевдоелементи</title>
  <style>
    P {
      font-family: "Times New Roman", Times, serif ;
      font-size: 16px;
      color: green;
    }
    P:first-letter {
      font-family: Arial, Helvetica, sans-serif;
      font-size: 36px;
      color: red;
      font-style:italic;
      font-weight:bolder;
    }
  </style>
</head>

<body>
  <p>Усі люди є вільні і рівні у своїй гідності
    та правах. Права і свободи людини
    є невідчужуваними та непорушними. </p>
  <p>Кожна людина має право на вільний

```



```
розвиток своєї особистості, якщо при
цьому не порушуються права і свободи
інших людей, та має обов'язки перед
суспільством, в якому забезпечується
вільний і всебічний розвиток її
особистості.</p>
```

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

4. :first-line – визначає стиль першого рядка блочного тексту.

Для нього допустимо застосування властивостей, які відносяться до шрифту, зміни кольору тексту та фону, а також `clear`, `line-height`, `letter-spacing`, `text-decoration`, `text-transform`, `vertical-align` та `word-spacing`.

Приклад:

```
<!DOCTYPE html>
<html>
<head>
  <title>Псевдоелементи</title>
  <style>
    P {
      font:Georgia, "Times New Roman", Times, serif;
      font-style:oblique;
    }
    P:first-line {
      color: red;
      font-style:normal;
      font-weight:bolder;
      text-transform:lowercase;
    }
  </style>
</head>

<body>
  <p>Рівність прав жінки і чоловіка забезпечується:
  <br />
  наданням жінкам рівних з чоловіками можливостей
  у громадсько-політичній і культурній
  діяльності, у здобутті освіти і професійній
  підготовці, у праці та винагороді за неї;
  <br />
  спеціальними заходами щодо охорони праці і
```

```

здоров'я жінок, встановленням пенсійних пільг;
<br />
створенням умов, які дають жінкам можливість
поєднувати працю з материнством;
<br />
правовим захистом, матеріальною і моральною
підтримкою материнства і дитинства, включаючи
надання оплачуваних відпусток та інших пільг
вагітним жінкам і матерям.
</p>
</body>
</html>

```

2.13. Каскадування та специфічність стилів

Каскад в CSS – одночасне застосування різних стильових правил до елементів документа за допомогою декількох стильових файлів, наслідування властивостей та інших методів. Для того, щоб браузер розумів, яке саме правило застосовувати до елемента введені деякі пріоритети (в порядку зростання пріоритету правила):

1. Стиль браузера.
2. Стиль автора.
3. Стиль користувача.
4. Стиль автора з додаванням `!important`.
5. стиль користувача з додавання `!important`.

!important використовується у випадку, коли користувачі підключають свої власні таблиці стилів, для підвищення пріоритету стилю.

Синтаксис:

Властивість: значення !important

Якщо до одного і того ж елемента застосовуються стильові правила з протиріччями, то більш високий пріоритет має те правило, у якого значення специфічності селектора більше. Специфічність – умовна величина, яка обчислюється за таким правилом:

за кожен ідентифікатор (кількість =a) нараховується 100, за кожен клас та псевдо клас (b) – 10, за кожен селектор тега та псевдоелемент (c) – 1.

Приклад:

```
* {} /* a=0 b=0 c=0 специфічність = 0 */
Li {} /* a=0 b=0 c=1 специфічність = 1 */
li:first-line {} /* a=0 b=0 c=2 специфічність = 2 */
ul li {} /* a=0 b=0 c=2 специфічність = 2 */
ul ol+li {} /* a=0 b=0 c=3 специфічність = 3 */
ul li.red {} /* a=0 b=1 c=2 специфічність = 12 */
li.red.level {} /* a=0 b=2 c=1 специфічність = 21 */
#t34 {} /* a=1 b=0 c=0 специфічність = 100 */
#content #wrap {} /* a=2 b=0 c=0 специфічність = 200 */
```

Діють також такі правила:

- вбудований стиль, який додається до тега через атрибут `style`, має специфічність 1000;
- проте, додавання `!important` перебиває в тому числі вбудовані стилі;
- якщо два селектори мають однакову специфічність, то застосовується той стиль, який в коді іде нижче.

Приклад:

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Список</title>
    <style>
      #menu ul li {
        color: green;
      }
      .two {
        color: red;
      }
    </style>
```

```
</head>

<body>
  <div id="menu">
    <ul>
      <li>Перший</li>
      <li class="two">Другий</li>
      <li>Третій</li>
    </ul>
  </div>
</body>
</html>
```

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ ТА ЕЛЕКТРОННИХ РЕСУРСІВ

Навчальна література

1. Э. Фримен, Э. Фримен Изучаем HTML, XHTML и CSS. Head First HTML with CSS & XHTML. — П.: «Питер», 2010. — 656 с. — ISBN 978-5-49807-113-8.
2. Эд Титтел, Джефф Ноубл. HTML, XHTML и CSS для чайников, 7-е издание. HTML, XHTML & CSS For Dummies, 7th Edition. — М.: «Диалектика», 2011. — 400 с. — ISBN 978-5-8459-1752-2.
3. Питер Лабберс, Брайан Олберс, Фрэнк Салим. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений. Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development. — М.: «Вильямс», 2011. — 272 с. — ISBN 978-5-8459-1715-7.
4. Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание. HTML, XHTML, and CSS Bible, 5th Edition. — М.: «Диалектика», 2010. — 656 с. — ISBN 978-5-8459-1676-1.

Електронні ресурси

1. http://uk.wikipedia.org/wiki/Головна_сторінка
2. <http://htmlbook.ru/>
3. <http://html.manual.ru/>
4. <http://www.codecademy.com/en/tracks/web>
5. <http://css.manual.ru/>