

# Manual QA

ОСНОВИ HTML

## План уроку

### 1. HTML

1. Огляд тегу `img` та його атрибутів
2. Тег `table`
3. Теги `ul`, `ol`, `dl`

### 2. CSS

1. Для чого потрібні таблиці стилів. Що таке CSS?
2. Типи стилів.
3. Переваги використання стилів.
4. Семантика CSS.
5. Прості селектори.
6. Одиниці виміру в CSS.

Що було минулого разу

## План уроку

- Концепція тег і атрибут.
- Структура сторінки HTML.
- Поняття блокових та малих тегів.
- Теги логічної та фізичної розмітки.
- Валідація сторінок HTML.
- Службові символи.

# Manual QA

Зображення

# Manual QA

## Зображення



Тег *<img>* призначений для відображення зображення на веб-сторінці.

Тег не має текстового вмісту, тому є самодостатнім.

```

```

Елемент має такі атрибути:

*src* – містить адресу графічного файлу.

*alt* – встановлює альтернативний текст для зображення.

*wtile* – встановлює текст спливаючої підказки.

*width, height* – атрибути задають ширину та висоту зображення.

*usemap* – пов'язує між собою картинку та карту-зображення, що задається за допомогою контейнера *<map>*.



# Manual QA

## Розміри зображень



- Під час додавання зображення на сторінку воно має оригінальний розмір. Але використовуючи атрибути *width* та *height*, можна регулювати висоту та ширину зображення.



- Розміри зображення можна задавати як у відсотках, так і в пікселях. Якщо розмір заданий у відсотках, то розміри зображення будуть обчислюватися щодо батьківського елемента або від вікна браузера у разі відсутності батьківського елемента.



- Якщо задати лише один параметр розміру зображення, інший зміниться пропорційно, у такому випадку, зображення може відображатися неправильно.



# Manual QA

Таблиці



# Manual QA

## Таблиці

Тег `<table></table>` є основним контейнером для елементів таблиці і всі теги, що належать до таблиці, повинні розташовуватися в ньому.



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

`<thead></thead>` – призначений для зберігання одного або декількох рядків заголовків

`<tbody></tbody>` – призначений для зберігання вмісту таблиці. Дозволяє створювати структурні блоки та керувати їх оформленням за допомогою стилів.

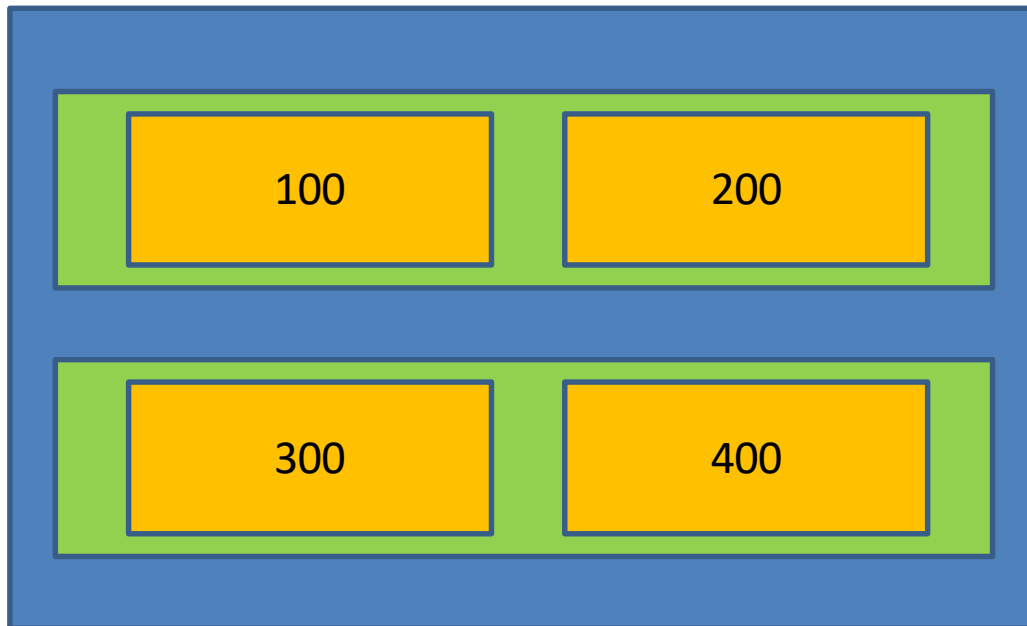
`<tfoot></tfoot>` – призначений для зберігання інформації, яка буде відображена внизу таблиці.



# Manual QA

## Рядки та осередки

Таблиця містить рядки, подані тегом `<tr></tr>`, які у свою чергу містять осередки, представлені тегом `<td></td>`, або `<th></th>` – якщо використовуються великі комірки.



|     |     |
|-----|-----|
| 100 | 200 |
| 300 | 400 |

`<table>`

`<tr>`

`<td>100</td>`

`<td>200</td>`

`</tr>`

`<tr>`

`<td>300</td>`

`<td>400</td>`

`</tr>`

`</table>`



# Manual QA

## Назва таблиці

Використовуючи тег `<caption></caption>` таблиці можна встановити назву, яка буде розміщена над таблицею. Сам тег слід помістити відразу після відкриваючого тега `<table>`.

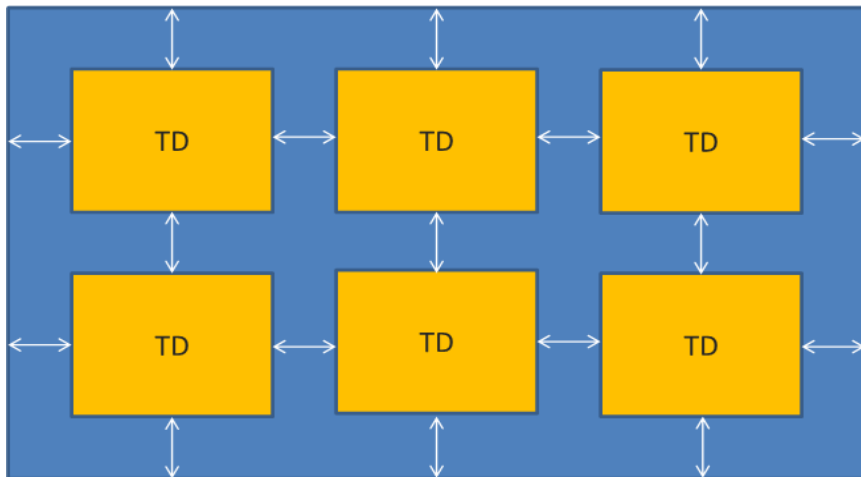
Дані користувачів

| Ім'я | Прізвище | Телефон | E-mail |
|------|----------|---------|--------|
|      |          |         |        |
|      |          |         |        |
|      |          |         |        |

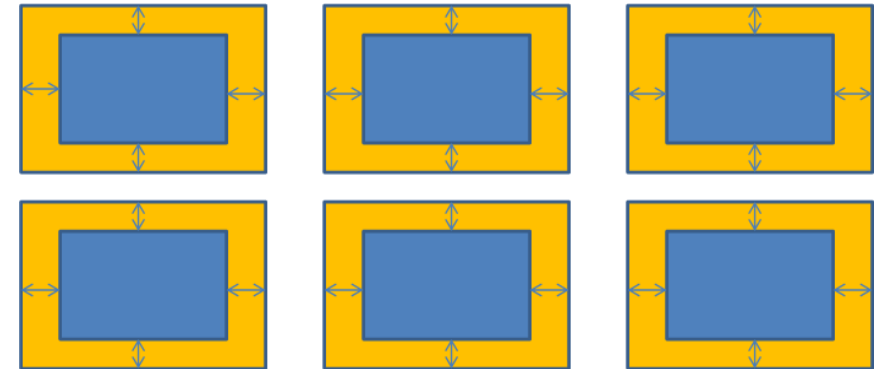


## Cellspacing та cellpadding

*Cellspacing* — це атрибут таблиці, який встановлює відступи між осередками таблиці та від межі таблиці.



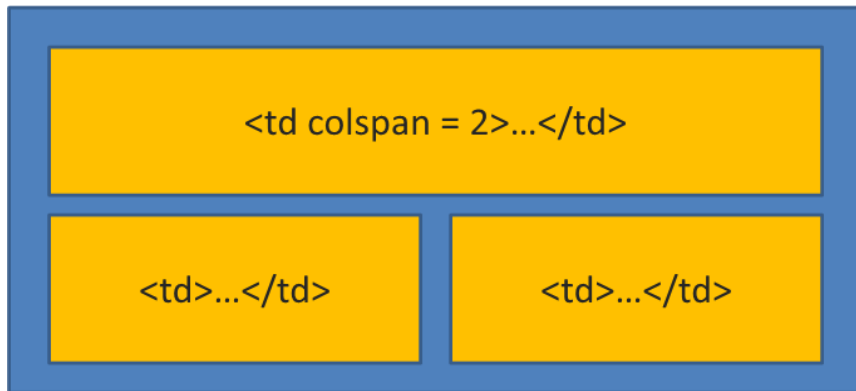
*Cellpadding* — це атрибут таблиці, який встановлює відступи усередині комірки, тобто відступи від межі комірки до її вмісту.



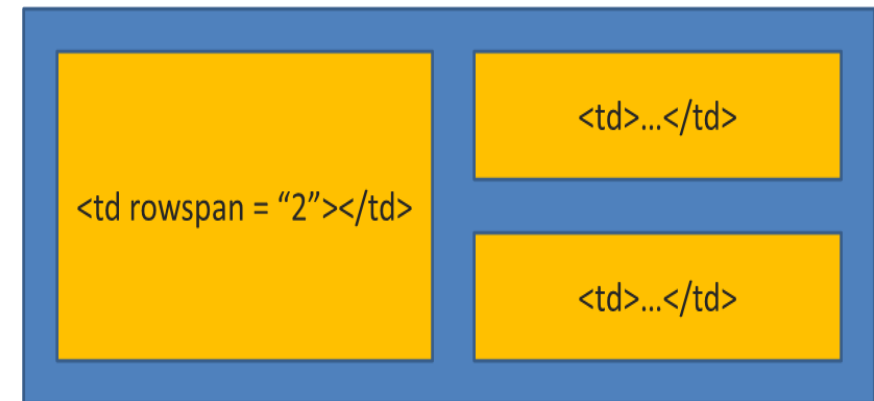
# Manual QA

## Colspan та rowspan

*Colspan* – це атрибут, що встановлює число осередків, які повинні бути об'єднані по горизонталі.



*Rowspan* – це атрибут, що встановлює кількість осередків, які мають бути об'єднані за вертикаллю.



# Manual QA

Списки

# Manual QA

## Списки

Списки призначені для угруповання пов'язаних між собою фрагментів інформації.

Існує три види списків:

`<ul></ul>` – маркований список. Кожен елемент списку `<li></li>` відзначається маркером.

`<ol></ol>` – нумерований список. Кожен елемент списку `<li></li>` відзначається цифрою.

`<dl></dl>` – список визначень. Складається з пар термін (тег `<dt></dt>`) – визначення (тег `<dd></dd>`).



# Manual QA

## Маркований список <ul>

Тег `<ul></ul>` (unordered list) є контейнером для невпорядкованих списків.

Має атрибут `type`:

- `disc` – маркер у формі чорного кола,
- `circle` – маркер у формі порожнього кола,
- `square` – маркер у формі чорного зафарбованого кола,
- `none` – без маркера

- Пункт №1
- Пункт №2
- Пункт №3

- Пункт №1
- Пункт №2
- Пункт №3

`<ul type="circle">...</ul>`

`<ul type="square">...</ul>`





# Manual QA

## Нумерований список

Тег `<ol></ol>` (ordered list) є контейнером для упорядкованих списків.

Має атрибут *type*:

- *A* – нумерація великими латинськими літерами
- *a* – маленькі латинські літери
- *I* – великі римські цифри
- *i* – маленькі римські цифри
- по умовчанию – арабські цифри.

Додаткові атрибути:

*reversed* – відображає список у зворотному порядку,

*start* – вказує початкове значення нумерації.

*value* – вказує номер конкретного елемента

A. Пункт №1

B. Пункт №2

C. Пункт №3

`<ol type="A">...</ol>`

a) Пункт №1

b) Пункт №2

c) Пункт №3

`<ol type="a">...</ol>`

I. Пункт №1

II. Пункт №2

III. Пункт №3

`<ol type="I">...</ol>`

i. Пункт №1

ii. Пункт №2

iii. Пункт №3

`<ol type="i">...</ol>`



# Manual QA

## Вкладені списки

<ul>

<li>Coffee</li>

<li>Tea

<ul>

<li>Black tea</li>

<li>Green tea</li>

</ul>

</li>

<li>Milk</li>

</ul>

В елемент списку можна вкладати не лише текстовий вміст, але й інші списки.

Таким чином, буде реалізовано вкладені списки, а точніше підписки.

За замовчанням у вкладених списках маркер буде відмінним від маркера батьківського списку.



# Manual QA

## Списки визначень

<dl>

<dt>Кава</dt>

<dd>Чорний гарячий напій</dd>

<dt>Молоко</dt>

<dd>Білий холодний напій</dd>

</dl>

Елементи таких списків складаються із двох складових:

- ключ <dt>...</dt>
- значення <dd>...</dd>



# Manual QA

## ОСНОВИ CSS

## Для чого потрібні таблиці стилів. Що таке CSS?



*CSS (Cascading Style Sheets)* – каскадні таблиці стилів, що використовуються для опису зовнішнього вигляду документа, написаного мовою розмітки.

Каскадні таблиці стилів описують правила форматування елементів за допомогою властивостей та допустимих для них значень. Для кожного елемента можна використовувати обмежений набір властивостей, решта не чинитиме на нього жодного впливу.

Розробкою стандартів займається організація W3C.

|                   |                                   |
|-------------------|-----------------------------------|
| 17 грудня 1996 г  | - вийшла версія CSS1              |
| 12 травня 1998 г  | - вийшла версія CSS2              |
| 7 червня 2011 г   | - вийшла версія CSS2.1            |
| 2006 – теп. час   | - розробляється версія CSS3       |
| 29 вересня 2011 г | - почала розроблятися версія CSS4 |

## Переваги застосування стилів

1. Розмежування коду та оформлення
2. Різне оформлення для різних пристроїв
3. Розширені способи форматування
4. Прискорення завантаження сайту
5. Єдине стильове оформлення безлічі документів
6. Централізоване зберігання

# Manual QA

## Типи стилів

Стилі представляють набір параметрів, що управляють виглядом та положенням елементів веб-сторінки.

Існує кілька типів стилів:

- *Стиль браузера* – оформлення, яке за замовчуванням застосовується до елементів веб-сторінки браузером.
- *Стиль автора* – стиль, який додає до документа його розробник.
- *Стиль користувача* – стиль, який може увімкнути користувач сайту через налаштування браузера. Такий стиль має вищий пріоритет і перевизначає вихідне оформлення документа.

# Manual QA

## Способи додавання стилів на сторінку

### 1. Inline або внутрішні стилі

Підключаються з використанням атрибуту *style*

```
<p style="...">Some text</p>
```

### 2. Глобальні стилі

Встановлюються тегом *<style>* тега *<head>*

```
<head>  
  <style>  
    ...  
  </style>  
</head>
```

### 3. Зовнішні стилі

Вказуються шляхи до .css файлу в атрибуті *href* тега *<link>* в *<head>*.

```
<link href="url" rel="stylesheet" />
```






# Manual QA

## Семантика CSS

Для стилів inline використовується наступний формат запису CSS стилів:


`<p style="background-color: red; font-size: 10px;">Some text</p>`



В теґі `<style>` і у зовнішніх стилях використовується наступний синтаксис:

селектор

```
{  
    властивість: значення;  
    властивість: значення;  
    властивість: значення;  
    ...  
}
```



Як селектор може виступати ім'я теґа, клас, ідентифікатор, також селектори можуть бути складні.

# Manual QA

## Селектор елемент

Якщо як селектор виступає ім'я елемента (тега), то в цьому випадку всі елементи даного тега будуть відповідати заданим стилям.

Якщо всередині цих тегів будуть текстові елементи, вони будуть успадковувати правила.

```
p
{
    font-family: Arial;
    font-size: 18:pt;
    color: green;
}
```



# Manual QA

## Селектор клас

Якщо необхідно встановити стилі для індивідуального елемента або встановити різні стилі для одного тега слід використовувати селектор клас.

Імена класів повинні починатися з латинського символу та можуть містити символ дефісу й підкреслення.

Щоб прив'язати до елемента клас, необхідно у тезі задати атрибут *class* та присвоїти йому ім'я класу, описаного в стилях.

```
<p class="testClass">Content</p>
```

```
.testClass
```

```
{  
    font-family: Arial;  
    font-size: 18:pt;  
    color: green;  
}
```



# Manual QA

## Селектор ідентифікатор

**Ідентифікатор** – особливий вид селектора, яким позначається один елемент на сторінці. Щоб вказати селектор ідентифікатор, в CSS необхідно використовувати конструкцію #ім'я\_ідентифікатора.

Щоб прив'язати до елемента ідентифікатор, необхідно у тезі, що відкриває, задати атрибут *id* та присвоїти йому ім'я ідентифікатора, описаного в стилях.

```
<p id="id1">Content</p>
```

```
#id1
{
    font-family: Arial;
    font-size: 18:pt;
    color: green;
}
```



# Manual QA

## Властивості width та height

Властивість *width* встановлює ширину блокових елементів.

Властивість *height* встановлює висоту блокових елементів.

Властивості не включають товщину меж навколо елемента, значення відступів та полів елемента.  
Як значення можна задавати будь-які допустимі в CSS одиниці виміру.



# Manual QA

## Одиниці виміру

Щоб задавати розміри різних елементів HTML розмітки, CSS використовуються абсолютні та відносні одиниці вимірювання.

*Відносні одиниці*

| Одиниця | Опис                             |
|---------|----------------------------------|
| em      | Висота шрифту поточного елемента |
| ex      | Висота символу X                 |
| px      | Піксель                          |
| %       | Відсоток                         |

*Абсолютні одиниці*

| Одиниця | Опис               |
|---------|--------------------|
| in      | Дюйм (2,54 см)     |
| cm      | Сантиметр          |
| mm      | Міліметр           |
| pt      | Пункт (1/72 дюйма) |
| pc      | Піка (12 пунктів)  |



Що ми сьогодні вивчили

## План уроку

### 1. HTML

1. Огляд тегу `img` та його атрибутів
2. Тег `table`
3. Мітки `ul`, `ol`, `dl`

### 2. CSS

1. Для чого потрібні таблиці стилів. Що таке CSS?
2. Типи стилів.
3. Переваги використання стилів.
4. Семантика CSS.
5. Прості селектори.
6. Одиниці вимірювання у CSS.



# Manual QA

## Підсумки

Що одне, найголовніше, ви дізнались сьогодні?

# Manual QA

Дякую за увагу! До нових зустрічей!

# Інформаційний відеосервіс для розробників програмного забезпечення

