



HTML

HTML – мова розмітки сторінки, за допомогою якої браузер відображає інформацію зрозумілим для користувача способом. Головною одиницею HTML являється тег (tag).

Тег – це команда для браузера. Кожен тег містить різний набір початкових значень (жирність, розмір і тд)

OWU

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

  </body>
</html>
```

`<!DOCTYPE html>` - Яка версія html використовується.

`<html>` - вказує межі html документа.

`<head>` - простір для стилів, визначення кодування сторінки (підтримка мови), та іншої метаінформації.

`<body>` - основний тег, контент якого відображається на сторінці.



Типи тегів

Блочні (block)

Блочні теги – теги які (за замовчуванням) займають усю ширину браузерного вікна.

Через цю причину всі інші теги (блочні або строчні), які зазначені в розмітці після блочних будуть відображатись з нового рядка.

Блочні елементи спроможні мати характеристики ширини (width), висоти (height) та інших, яких не мають строчні елементи

Строчні (inline)

Строчні теги – теги які займають простір характеризуємий контентом. Не розривають рядок.

Не мають характеристик width, height за замовчуванням.

Block-level

This is a paragraph.

This is another paragraph.

Paragraphs are block-level elements, so they stack vertically.

Inline

Links are

inline elements,

so they fit side-by-side.



Базові теги

Блочні (block)

h1...h6 - заголовки.

p - параграф.

div - чистий контейнер.

ul - маркований список.

ol - нумерований список.

li - елемент списку

Строчні (inline)

a - посилання.

span - чистий елемент.

b - напівжирний шрифт

strong - напівжирний шрифт (SEO).

i - курсив.

u - підкреслення.

img - картинка. (Має гібридну поведінку)

`<body>`

```
<h1> PuzzleWeb.ru Online Web Tutorial </h1>
```

```
<p>
```

```
.....  
.....  
..<a href="https://puzzleweb.ru">HTML</a>..  
.....
```

```
</p>
```

`</body>`

block elements

inline element



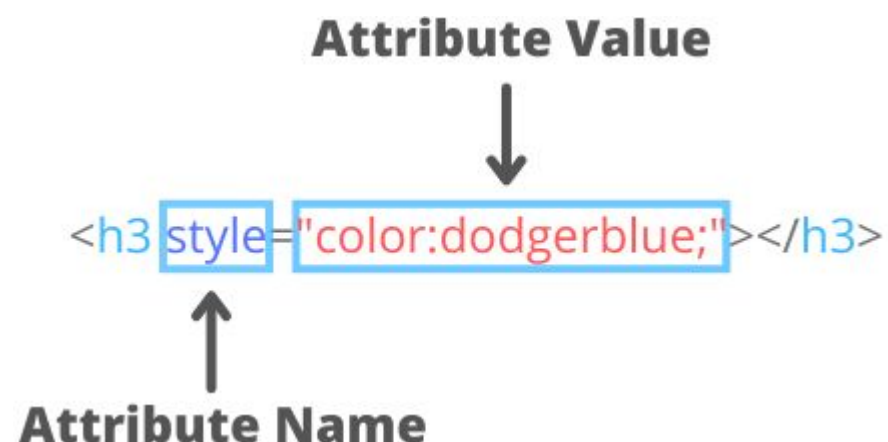
Атрибути

Раніше ми зазначили, що тег – це команда, яку повинен виконати браузер. Атрибут – це додаткова інформація під час виконання команди. До атрибутів відносить можливість визначення стилів, класів, ідентифікаторів, джерел медіаконтенту і т.д..

```
<a href="https://owu.com.ua"> OKTEN SCHOOL</a>
```

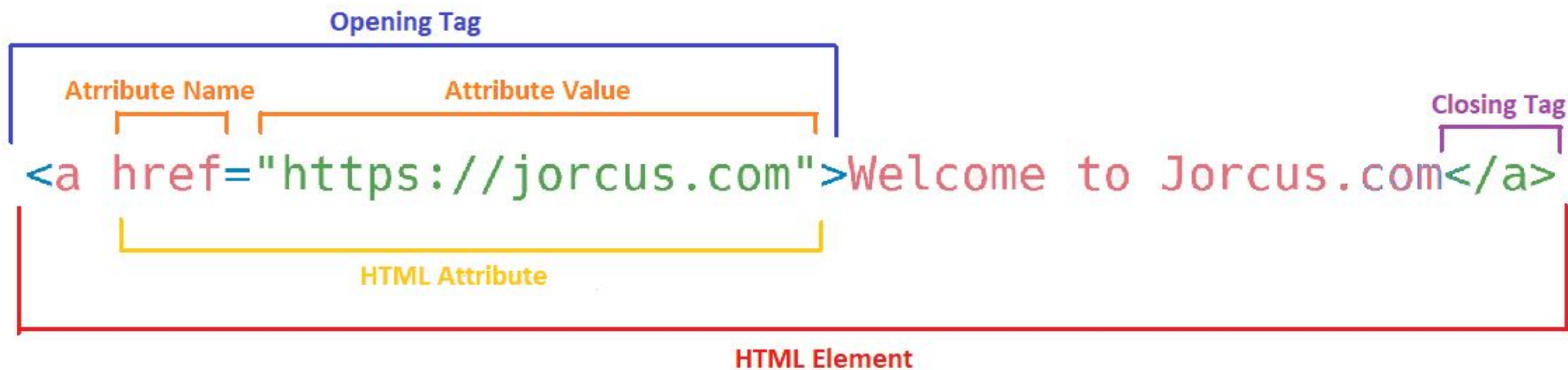
В поточному прикладі тег `<a>` містить атрибут `href`, значення якого посилається на конкретний сайт.

Існують загальні майже для всіх елементів атрибути, такі як : `style`, `class`, `id`, `name`. А деякі атрибути можуть бути визначенні тільки в окремих тегах.





Загальна структура тегу з атрибутами



OWU



Атрибути id та class

Атрибути id та class надають можливість тегам мати додаткові "імена-ідентифікатори". В свою чергу, значення атрибуту id повинно бути унікальним в межах одного html документа (НЕ проекту), а атрибут class може бути не унікальним і фігурувати в багатьох елементах різних типів. Ці атрибути знадобляться нам в майбутньому для стилізації та застосування мови javascript.

```
<p id="unique_id" class="some_class">okten is cool</p>
```

OWU



Стилі. Вбудований (inline)

Вбудований тип стилізації застосовується для тегу, безпосередньо в тілі його "відкриваючого" тегу, і впливає тільки на один елемент. Цей тип є найбільш пріоритетним в порівнянні з іншими типами стилізації, але його використання вносить надлишковість і через це зменшує продуктивність та робить підтримку проекту дуже не зручною. Намагайтесь використовувати його тільки за крайньої необхідності.

Наступний приклад явно показує приклад надлишковості в коді.

```
<h2>Переваги курсів від OKTEN</h2>
<ul>
  <li style="background: silver">
    Понад 1000 працевлаштованих випускників.
  </li>
  <li style="background: silver">
    Понад 400 відео відгуків.
  </li>
  <li style="background: silver">
    Інтернатура з іт-компаніями.
  </li>
</ul>
```




Глобальний стиль

Тип стилізації який описується глобально для окремого документу. Для визначення елементу стилізації використовуються селектори. В css розрізняють 31-33 різновиди селекторів. Глобальний тип стилізації визначається за допомоги тегу `<style>` в межах `<head>`. Але це не означає, що він не може знаходитись в іншому місці.

```
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
    /*your style here*/
  </style>
</head>
```

owu

В цьому прикладі видно, на скільки менше коду необхідно написати, щоб отримати той самий результат, як і в попередньому прикладі.

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Title</title>
    <style>
      li {
        background: silver;
      }
    </style>
  </head>
  <body>
    <h2>Переваги курсів від OKETN</h2>
    <ul>
      <li>
        В 11 разів більше відгуків від працевлаштованих випускників
      </li>
      <li>
        в 3 рази більший об'єм.
      </li>
      <li>
        Єдині курси зі спільною інтернатурою.
      </li>
    </ul>
  </body>
</html>
```

І саме тут ми можемо побачити перше використання селекторів. Тут li – це селектор по тегу. Іншими словами – звернення до всієї тегів li.

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Title</title>
    <style>
      li {
        background: silver;
      }
    </style>
  </head>
  <body>
    <h2>Переваги курсів від OKETN</h2>
    <ul>
      <li>
        В 11 разів більше відгуків від працевлаштованих випускників
      </li>
      <li>
        в 3 рази більший об'єм.
      </li>
      <li>
        Єдині курси зі спільною інтернатурою.
      </li>
    </ul>
  </body>
</html>
```

А що робити, якщо нам потрібно стилізувати тільки окремі елементи li, а не всі які містить поточний документ?

Для цього використовують різноманітні варіанти селекторів, найпростіші з яких – class та id.

```
1  <!doctype html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>Title</title>
6    <style>
7      .target {background: silver;}
8      #point {background: cornflowerblue;}
9      h2.target {color: red;}
10   </style>
11 </head>
12 <body>
13   <h2 class="target">Переваги курсів від OKETN</h2>
14   <ul>
15     <li class="target">
16       В 11 разів більше відгуків від працевлаштованих випускників
17     </li>
18     <li id="point">
19       в 3 рази більший об'єм.
20     </li>
21     <li class="target">
22       Єдині курси зі спільною інтернатурою.
23     </li>
24   </ul>
25
26 </body>
27 </html>
28
```

Зверніть увагу на 9 строку. Тут використовується композитний селектор, який має одночасне звернення до тегу з певним класом.



Під'єднуємий стиль

Під'єднуємий тип стилізації характеризується окремим файлом з розширенням .css та підключається до багатьох html файлів за допомогою тегу link з двома обов'язковими атрибутами rel та href

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

Вся інша синтаксична конструкція ідентична глобальному типу стилізації.

Найменш пріоритетний тип стилізації, але найзручніший через те, що спроможний бути підключений до багатьох сторінок з однаковими стилями, що дозволяє зменшити кількість css коду.

Послідовність підключення стилів впливає на кінцевий результат.

В поточному прикладі тег а буде мати колір фону cornflowerblue, тому що глобальні стилі декларуються після link, відповідно перевизначають стилі оголошені в your.css файлі.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6     <link rel="stylesheet" href="your.css">
7     <style>
8       a {
9         background: cornflowerblue;
10      }
11    </style>
12  </head>
13  <body>
14    <a href="https://owu.com.ua"> OKTEN SCHOOL</a>
15  </body>
16 </html>
```

html > head > style > a

```
css your.css x
1 a {
2   background: silver;
3 }
```

В поточному прикладі тег а буде мати колір фону silver, тому що глобальні стилі декларуються перед link, відповідно стилі з your.css будуть застосовуватись останніми.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6     <style>
7       a {
8         background: cornflowerblue;
9       }
10    </style>
11    <link rel="stylesheet" href="your.css">
12  </head>
13  <body>
14    <a href="https://owu.com.ua"> OKTEN SCHOOL</a>
15  </body>
16 </html>
```

html > head > style

```
css your.css x
1 a {
2   background: silver;
3 }
```