

Урок 2

HTML

Теги (элементы)

Тег `` является основным строковым контейнером для фразового контента, который, по существу, ничего не представляет. Он может использоваться для группировки элементов в целях стилизации. Например, внутри абзаца (тега `<p>`) можно изменить цвет и размер слова, если добавить начальный и конечный тег `` и определить для него стиль текста.

HTML

```
<p><span>Lorem</span> ipsum dolor sit, amet consectetur adipisicing elit. Eos  
dolorem ipsum doloremque exercitationem nulla quasi natus! Nobis inventore in at  
asperiores et itaque earum voluptatem, omnis deserunt necessitatibus, molestias  
illo!</p>
```

```
<p><span>L</span>orem ipsum dolor sit, amet consectetur adipisicing elit. Eos  
dolorem ipsum doloremque exercitationem <span>nulla</span> quasi natus! Nobis  
inventore in at asperiores et itaque earum voluptatem, omnis deserunt  
necessitatibus, molestias illo!</p>
```

```
<p><span>L</span>orem ipsum dolor sit, amet consectetur adipisicing elit. Eos  
dolorem ipsum doloremque exercitationem nulla quasi natus! Nobis inventore in at  
asperiores et itaque earum voluptatem, omnis deserunt necessitatibus, molestias  
illo!</p>
```

CSS

```
span{  
  font-size: 20px;  
  color: red;  
}
```

Выделяют две основные категории HTML-элементов, которые соответствуют типам их содержимого и поведению в структуре веб-страницы — блочные и строчные элементы.

Блочные элементы

Блочные элементы являются основой, которая используется для верстки веб-страниц. Такой элемент представляет собой прямоугольник, который по умолчанию занимает всю доступную ширину страницы (если иное значение не указано в CSS), а длина элемента зависит от его содержимого. Такой элемент всегда начинается с новой строки, то есть, располагается под предыдущим элементом. Блочный элемент может содержать в себе другие блочные и строчные элементы.

Примеры блочных элементов: `<p>`, `<h1>`-`<h6>`, `<div>` и т. д.

Строчные элементы

В отличие от блочного, строчный элемент не переносится на новую строку, а располагается на той же строке, что и предыдущий элемент. Такие элементы, как правило, находятся внутри блочных элементов, и их ширина зависит лишь от содержимого и настроек CSS. Еще одно отличие строчного элемента от блочного заключается в том, что в нем может находиться только контент и другие строчные элементы. Блочные элементы в строчные вкладывать нельзя.

Примеры строчных элементов: <a>, , и т. д.

```
<!-- Блочный элемент - занимает всю доступную ширину страницы -->
<p>Hello</p>
<p>Hello</p>
<p>Hello</p>

<!-- Строчный элемент - ширина зависит от содержимого -->
<span>Hello</span>
<span>Hello</span>
<span>Hello</span>

<!-- Строчный элемент - ширина зависит от содержимого -->
<a href="#">Click!</a>
<a href="#">Click!</a>
<a href="#">Click!</a>
```

Div

Тег **<div>** в HTML является блочным элементом верстки. Используется как контейнер для содержимого. С помощью тега **<div>** обычно создается каркас и внутренняя структура страницы. Тег **<div>** может содержать вложенные **<div>** элементы в неограниченном количестве.

HTML элемент **<div>** - основной тег верстки современного сайта.

Как и при использовании других блочных элементов, содержимое тега **<div>** всегда начинается с новой строки. После него также добавляется перенос строки.

HTML

```
<h1>My blog</h1>

<div id="block1">
  <h2>My hobby</h2>
  
  <br>
  <a href="https://en.wikipedia.org/wiki/Piano">Wikipedia</a>
</div>

<div id="block2">
  <h2>My work</h2>
   <br>
  <a href="https://en.wikipedia.org/wiki/Computer_programming"><img src=""
alt=""></a>
</div>
```

CSS

```
#img_piano{
  width: 200px;
  height: 200px;
}

#img_work{
  width: 200px;
  height: 160px;
}

#block1{
  border: 1px solid black;
  background-color: bisque;
}

#block2{
  border: 1px solid brown;
  background-color: lightgrey;
}
```

CSS

Рамка

Универсальное свойство **border** позволяет одновременно установить толщину, стиль и цвет границы вокруг элемента. Значения могут идти в любом порядке, разделяясь пробелом, браузер сам определит, какое из них соответствует нужному свойству. Для установки границы только на определенных сторонах элемента, воспользуйтесь свойствами `border-top`, `border-bottom`, `border-left`, `border-right`.

Синтаксис

```
border: [border-width || border-style || border-color]
```

Значения

Значение **border-width** определяет толщину границы. Для управления ее видом предоставляется несколько значений **border-style**. Их названия и результат действия представлен на Табл. 1.

Табл. 1. Вид рамки в зависимости от стиля и толщины границы элемента

1 пиксел	3 пиксела	5 пикселей	7 пикселей
			
			
			
			
			
			
			
			

border-color устанавливает цвет границы, значение может быть в любом допустимом для CSS формате.

CSS-селекторы

Селектор определяет, к какому элементу применять то или иное CSS-правило.

- Селекторы по типу элемента (название тега)

Этот базовый селектор выбирает тип элементов, к которым будет применяться правило.

Синтаксис: **название_тега**

Пример: селектор `img` выберет все элементы ``.

```
img{
  width: 200px;
  height: 150px;
}
```

- Селекторы по классу

Этот базовый селектор выбирает элементы, основываясь на значении их атрибута `class`.

Синтаксис: **.имяКласса**

Пример: селектор `.logo` выберет все элементы с соответствующим классом (который был определён в атрибуте `class="logo"`).

HTML

```
<a href="https://www.facebook.com/" target="_blank">  </a>
<br>
<a href="https://www.linkedin.com/" target="_blank"> </a>
```

CSS

```
.logo{
  width: 40px;
  height: 40px;
}
```

- Селекторы по идентификатору

Этот базовый селектор выбирает элементы, основываясь на значении их id атрибута. Не забывайте, что идентификатор должен быть уникальным, т. е. использоваться только для одного элемента в HTML-документе.

Синтаксис: **#имяИдентификатора**

Пример: селектор #instagramlogo выберет элемент с идентификатором instagramlogo (который был определён в атрибуте id="instagramlogo").

HTML

```
<a href="https://www.instagram.com/" target="_blank">
  
</a>
```

CSS

```
#instagramlogo{
  width: 20px;
  height: 20px;
}
```

Отступы

Внешние

Свойство **margin** CSS определяет внешний отступ на всех четырёх сторонах элемента. Это сокращение, которое устанавливает все отдельные поля одновременно: margin-top, margin-right, margin-bottom и margin-left. Если у элемента нет родителя, отступом будет расстояние от края элемента до края окна браузера с учетом того, что у самого окна по умолчанию тоже установлены отступы.

Свойство margin позволяет задать величину отступа сразу для всех сторон элемента или определить ее только для указанных сторон.

Внутренние

Свойство **padding** устанавливает внутренние отступы/поля со всех сторон элемента. Область отступов — это пространство между содержанием элемента и его границей. Отрицательные значения не допускаются. Свойство padding краткая форма записи свойств, чтобы не писать отдельное правило для каждой стороны (padding-top, padding-right, padding-bottom, padding-left).

И для margin и для padding разрешается использовать одно, два, три или четыре значения, разделяя их между собой пробелом. Эффект зависит от количества значений:

- 1 - Отступы будут установлены одновременно от каждого края элемента.
- 2 - Первое значение устанавливает отступ от верхнего и нижнего края, второе — от левого и правого.
- 3 - Первое значение задает отступ от верхнего края, второе — одновременно от левого и правого края, а третье — от нижнего края.
- 4 - Поочередно устанавливается отступ от верхнего, правого, нижнего и левого края.

HTML

```
<div id="block1">
    <h2>My hobby</h2>
    
    <br>
    <a href="https://en.wikipedia.org/wiki/Piano">Wikipedia</a>
</div>
```

CSS

```
#block1{
    border: 1px solid black;
    background-color: bisque;
    margin: 40px 0 100px 150px;
    /* либо так
    margin-top: 40px;
    margin-right: 0px;
    margin-bottom: 100px;
    margin-left: 150px;
    */

    padding: 20px 40px 60px 80px;
    /* либо так
    padding-top: 20px;
    padding-right: 40px;
    padding-bottom: 60px;
    padding-left: 80px;
    */
}
```