

**HTML**



**CSS**



# HTML

---

- Не является языком программирования.
- Весь синтаксис состоит из тегов (`<..>` и `</..>`).
- Существует две структуры верстки: табличная и блочная.
- Расширение файла с исходным кодом – `html`. (пример `index.html`)

# СТРУКТУРА HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```



<HEAD>

<TITLE>Заголовок документа</TITLE>

</HEAD>

<BODY>

Текст документа

</BODY>

</HTML>

# СТАНДАРТЫ HTML

---

## □ XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

## □ XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

## □ HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
  "http://www.w3.org/TR/html4/strict.dtd">
```

## □ HTML5

```
<!DOCTYPE html>
```

# ШАБЛОН HTML

---

```
<!DOCTYPE html>
<html>
  <head>
    <title>Супер сайт!</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="/css/style.css">
    <script type="text/javascript" src="/js/javascript.js"></script>
  </head>
  <body>
    <h1>Очень полезный ресурс</h1>
    <div>
      <p>Статья N°1</p>
      <p>Статья N°2</p>
    </div>
  </body>
</html>
```

# HTML. РАБОТА С ТЕКСТОМ.

---

- `<p>..</p>` - абзац.
- `<br>` - переход на новую строку.
- `<b>..</b>` - полужирный шрифт.
- `<i>..</i>` - курсив.
- `<u>..</u>` - подчеркнутый текст.
- `<sup>..</sup>` - надстрочный текст.
- `<sub>..</sub>` - подстрочный текст.
- `<h1>..</h1>`, `<h2>..</h2>`, `<h3>..</h3>` - заголовки.

# HTML. СПИСКИ.

---

- Маркированный список

```
<ul>
```

```
<li>Первый пункт списка</li>
```

```
<li>Второй пункт списка</li>
```

```
</ul>
```

- Нумерованный список

```
<ol>
```

```
<li>Первый пункт списка</li>
```

```
<li>Второй пункт списка</li>
```

```
</ol>
```

# HTML. КАРТИНКИ.

---

- `<img>` - тег для создания ссылки на графическое изображение. У него нет закрывающего тега.

Атрибуты тега `<img>`:

- `src` – указывает адрес файла.
- `align` – положение на странице.
- `width` – ширину объекта.
- `height` – высоту объекта.



# HTML. ВИДЕО.

---

□ `<video>`

`<source src="">`

`</video>` - структура позволяющая добавлять видео файл на web страницу.

Атрибуты для `<video>`:

- `autoplay` – видео воспроизводится сразу после загрузки страницы.
- `controls` – добавляет панель управления к ролику.
- `height` – задает высоту области воспроизведения.
- `loop` – повторяет видео сразу после его завершения.
- `poster` – указывает адрес картинки, которая отображается превью.
- `preload` – начинает загружать видео вместе со страницей.
- `src` – указывает путь к воспроизводимому файлу.
- `width` – задает ширину области воспроизведения.

# HTML. МУЗЫКА.

---

□ `<audio>`

`<source src="">`

`</audio>` - структура позволяющая добавлять аудио файл на web страницу.

Атрибуты для `<audio>`:

- `autoplay` – звук воспроизводится сразу после загрузки страницы.
- `controls` – добавляет панель управления к аудио.
- `loop` – повторяет аудио сразу после его завершения.
- `preload` – начинает загружать аудио вместе со страницей.
- `src` – указывает путь к воспроизводимому файлу.

# HTML. ФОРМЫ.

---

- `<form action="">`
  - `<input type="text">`
  - `<input type="submit" value="Отправить">`
- `</form>` - структура, позволяющая создавать форму взаимодействия с пользователем.

Атрибуты для `<form>`:

- `action` – адрес скрипта который получит данные для обработки.
- `method` – метод передачи данных в скрипт.
- `enctype` – способ кодирования данных формы.

Атрибуты для `<input>`:

- `type` – тип формы.
- `name` – имя формы.
- `required` – делает поле обязательным для заполнения.
- `placeholder` – пишет образец текста или название поля в поле ввода.

# HTML. ТАБЛИЦЫ.

---

```
<table>
  <tr>
    <td>...</td>
    <td>...</td>
  </tr>
```

</table> - структура для создания таблиц.

Атрибуты для <table>:

align – определяет выравнивание таблицы.

background – задает фоновый рисунок таблице.

bgcolor – цвет фона таблицы.

border – толщина рамки в пикселях.

height – высота таблицы.

width – ширина таблицы.

# HTML. ССЫЛКИ.

---

Адрес ссылки может быть как абсолютным, так и относительным. Абсолютные адреса должны начинаться с указания протокола (обычно `http://`) и содержать имя сайта. Относительные ссылки ведут отсчет от корня сайта или текущего документа.

## □ Относительные ссылки

- Файлы в одной папке

`<a href="Ссылаемый документ.html">Ссылка</a>`

- Файлы в разных папках

`<a href="../Ссылаемый документ.html">Ссылка</a>`

`<a href="../../Ссылаемый документ.html">Ссылка</a>`

`<a href="Папка/Ссылаемый документ.html">Ссылка</a>`

- Ссылка относительно корня сайта

`<a href="/Папка/Ссылаемый документ.html ">Курсы</a>`

## □ Абсолютные ссылки

- `<a href="http://htmlbook.ru">`

# CSS.

---



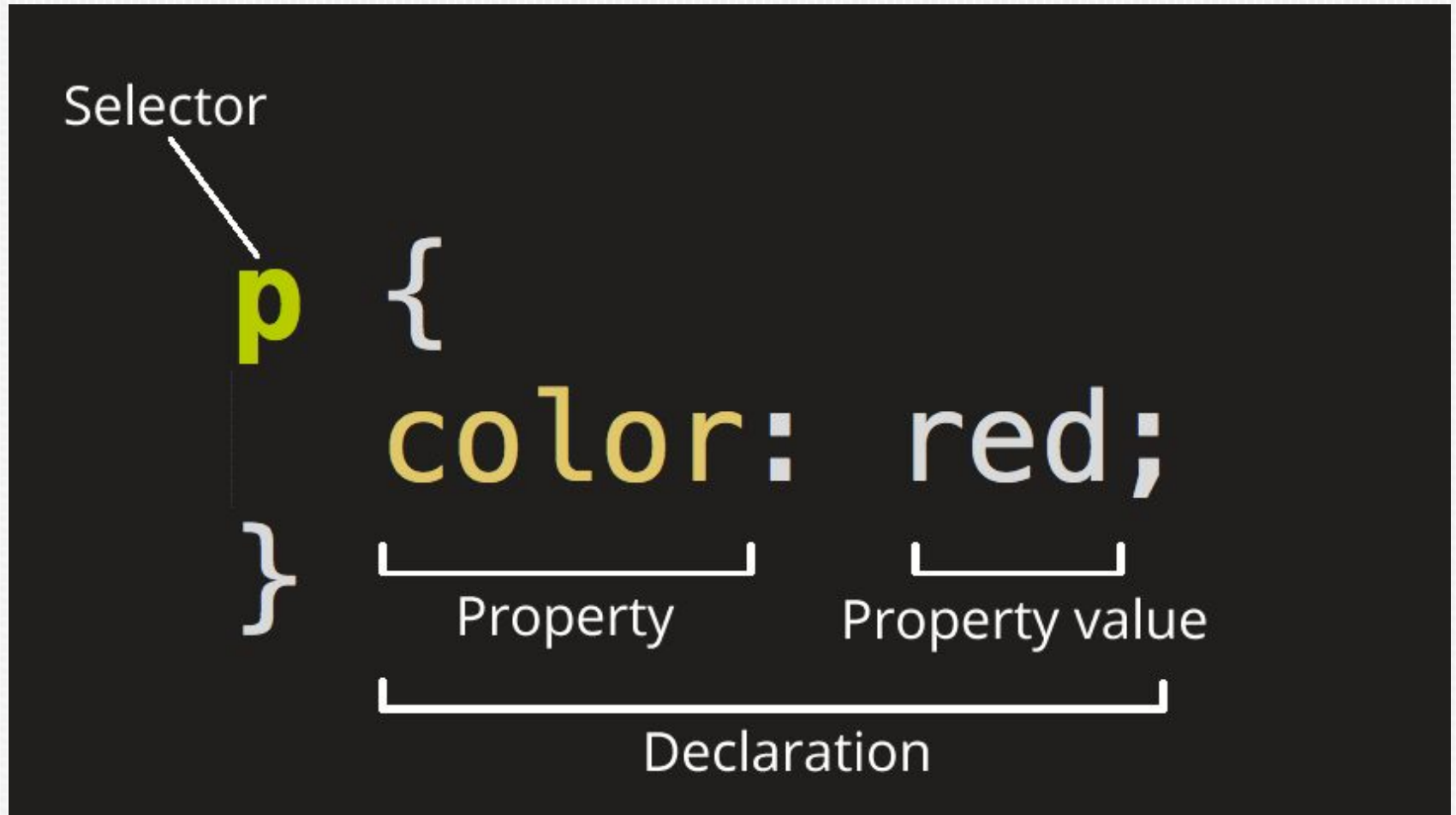
# CSS.

---

- CSS – расшифровывается как каскадная таблица стилей.
- Это правила по которым браузер отображает HTML элементы.
- Стили указываются в head страницы.

# CSS.

---





# CSS. МЕТОД ОБЪЯВЛЕНИЯ №1

---

- Объявляем через атрибут style.

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

# CSS. МЕТОД ОБЪЯВЛЕНИЯ №2

---

- Сводка таблицы стилей внутри основного документа.

```
<html>
  <head>
    <title>Example</title>
    <style type="text/css">
      p {
        color: #FF0000;
      }
    </style>
  </head>
  <body>
    <p>This is a red page</p>
  </body>
</html>
```

# CSS. МЕТОД ОБЪЯВЛЕНИЯ №3

---

- Ссылка на таблицу стилей.

```
<html>
  <head>
    <title>Example</title>
    <link rel="stylesheet" type="text/css" href="style/style.css" />
  </head>
  <body>
    <p>This is a red page</p>
  </body>
</html>
```

# CSS.

---

Позволяет работать с:

- Шрифтами
- Цветом
- Полями
- Строками
- Высотой
- Шириной
- Фоновыми изображениями
- Позиционированием элементов
- И другими вещами...

# CSS. ЦВЕТА.

---

- Возможно указывание цветов по имени (white, black, blue...)
- В шестнадцатеричном формате палитры RGB (#732acb, #3c3).
- В десятичном формате палитры RGB ( rgb(0,255,126) ).
- В формате RGB с указанием величин в процентах ( rgb(30%, 10%, 90%) ).

# CSS. РАБОТА С ТЕКСТОМ.

---

- `text-indent` – устанавливает отступы для первой строки абзаца.
- `text-align` – выравнивание текста.
- `text-decoration` – позволяет добавлять различные декоративные эффекты.
- `letter-spacing` интервал между буквами в пикселях, пунктах, в относительных единицах `em` – относительная единица длины, равная размеру текущего шрифта.
- `text-transform` – трансформация текста.

# CSS. ПСКВДОКЛАСС.

---

Псевдокласс позволяет учитывать различные условия или события при определении свойств HTML тэгов.

Пример: `a:link { color: #023fcb; }`

Псевдоклассы для ссылок:

`:link` – используется для ссылок на страницы, которые пользователь еще не посещал.

`:visited` – используется для ссылок на страницы, которые пользователь посетил.

`:hover` – используется для ссылок над которыми находится указатель мыши.

`:active` – используется для активных ссылок.

# CSS. ИДЕНТИФИКАЦИЯ ЭЛЕМЕНТОВ.

- class – задает стилевой класс, который позволяет связать определенный тег со стилевым оформлением. В значении допускается указывать сразу несколько классов, разделяя их между собой пробелом.
- id – задает стилевой идентификатор, уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты. Идентификатор в коде документа должен быть в единственном экземпляре – должен встречаться один раз.

```
<ul>
  <li class="class1">CLASS1</li>
  <li class="class1">CLASS1</li>
  <li class="class2">CLASS2</li>
  <li id="id1" class="class2">CLASS3 + ID</li>
</ul>
```



```
.class1 {
  color: red;
}
.class2 {
  color: yellow;
}
#id1 {
  color: blue;
}
```



- CLASS1
- CLASS1
- CLASS2
- CLASS3 + ID



# CSS. ПРИОРИТЕТНОСТЬ.

---

- Самый высокий приоритет имеет атрибут **style**. Это правило перекрывает все селекторы описанные в стилях.
- Второе место занимает присутствие **ID** в селекторе (`#some-id`).
- Далее идут все атрибуты (в том числе и атрибут **class**) и псевдоклассы (`pseudo-classes`) в селекторе.
- Самый низкий приоритет у селекторов с именами элементов.
- Если к паре свойство – значение приписать **!important**, тогда это значение свойства получит максимальный приоритет в каскаде стилей. Это объявление сработает независимо от наличия других правил CSS для этого элемента.

# HTML + CSS

---

