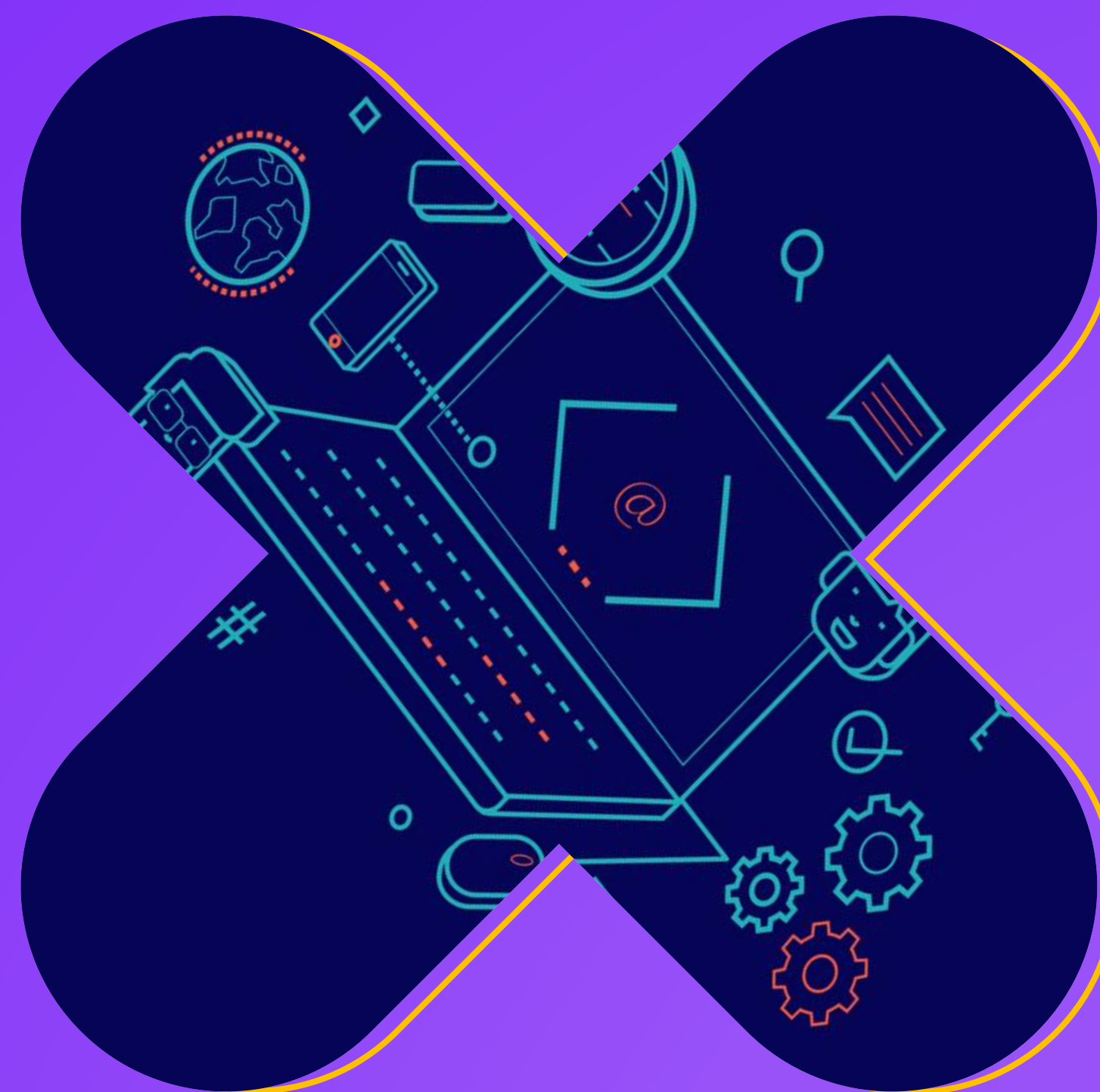




УРОК №1

# Введение в HTML



# Популярные сайты



[YouTube.com](https://www.youtube.com)



[Netflix.com](https://www.netflix.com)



[Playstation.com](https://www.playstation.com)





# Введение

## ЧТО ТАКОЕ HTML

HTML — это язык гипертекстовой разметки, с помощью которого создают структуру веб-страниц. Он позволяет пользователю создавать и структурировать разделы, параграфы, заголовки, ссылки и блоки для веб-страниц и приложений.

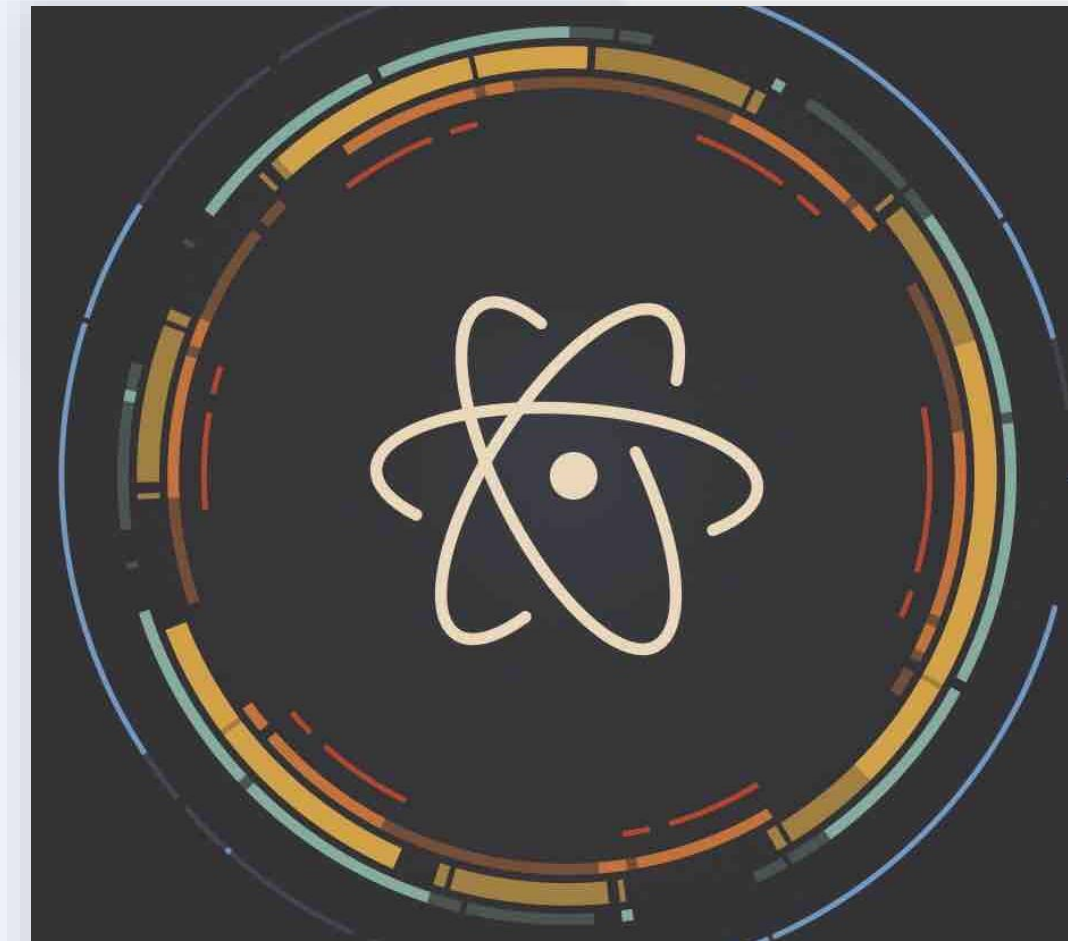
# Программы для создания сайта



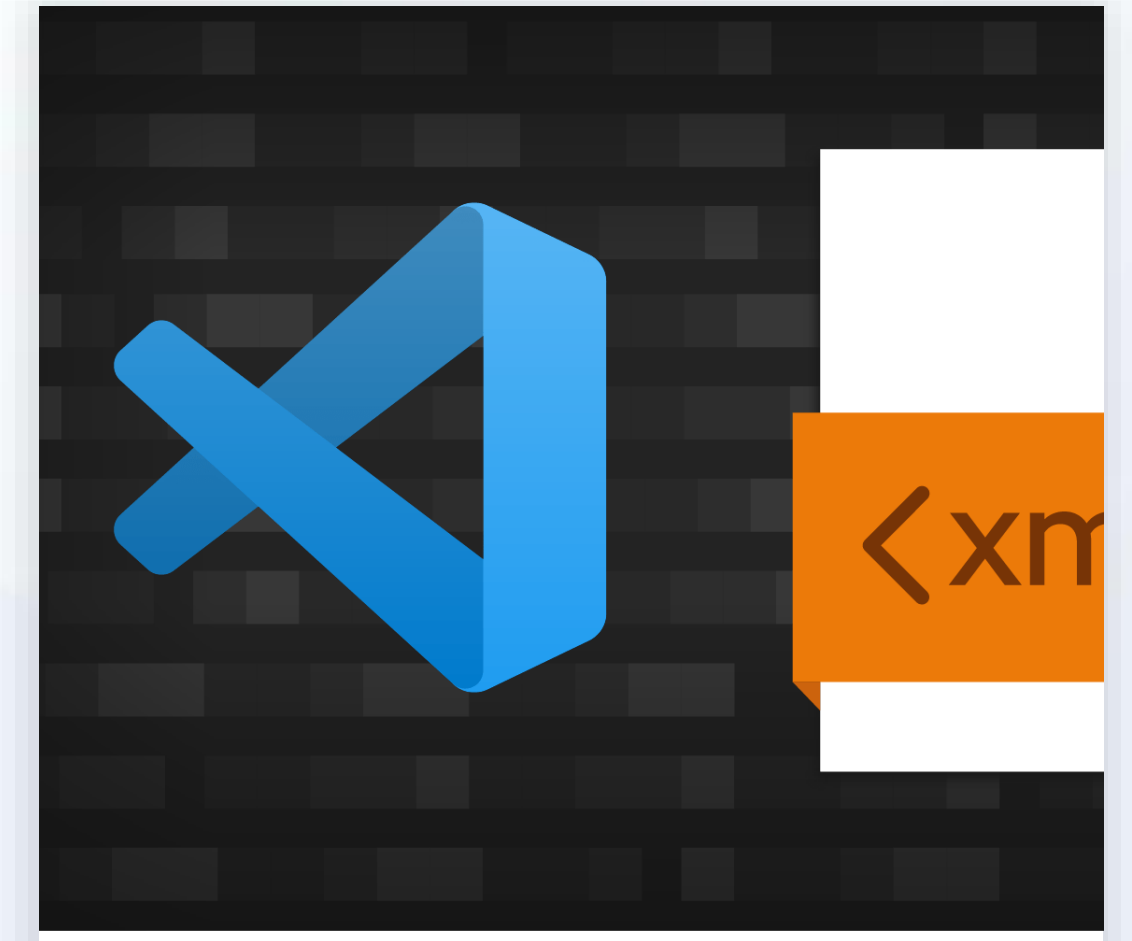
Notepad++



Sublime text 3

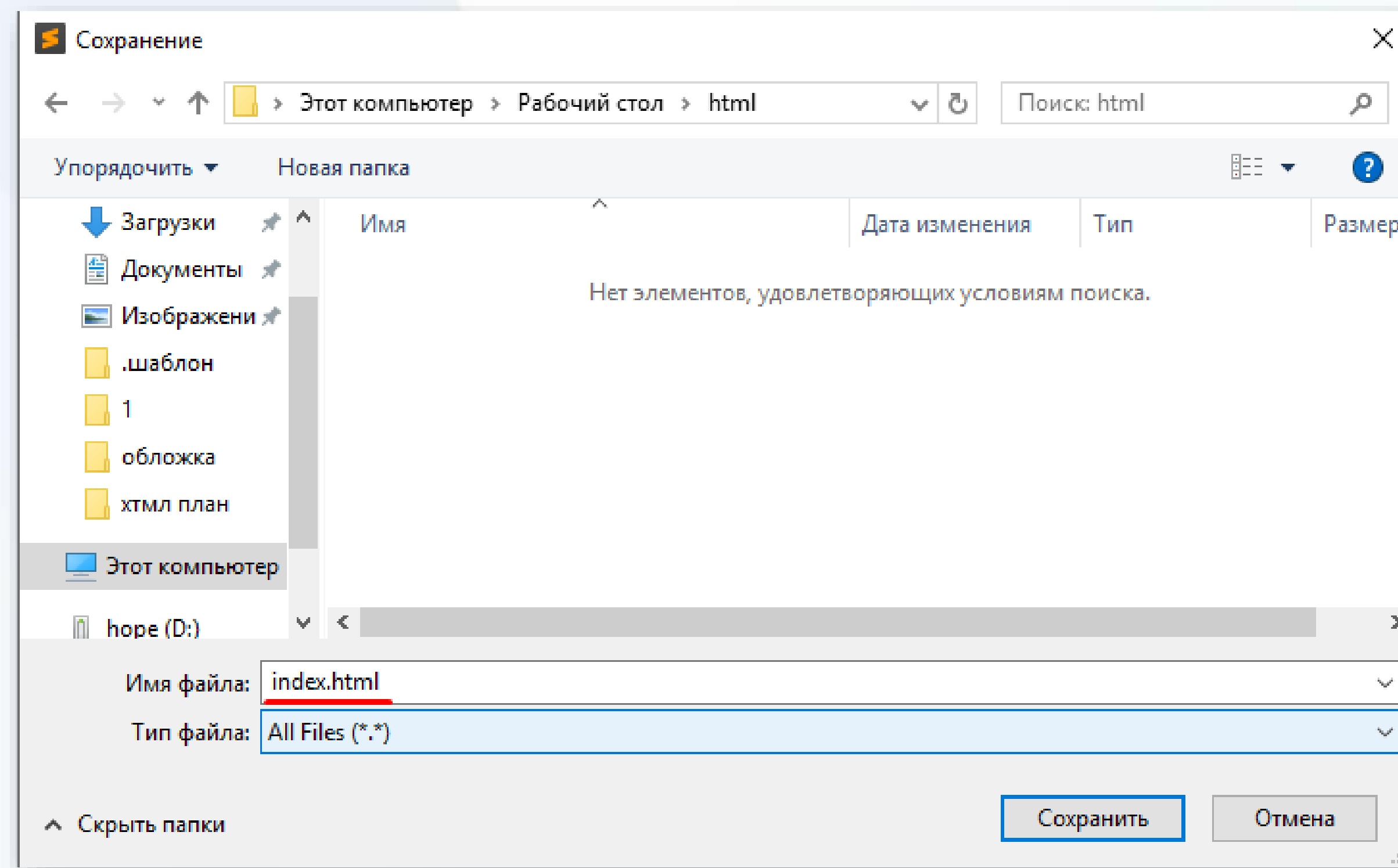
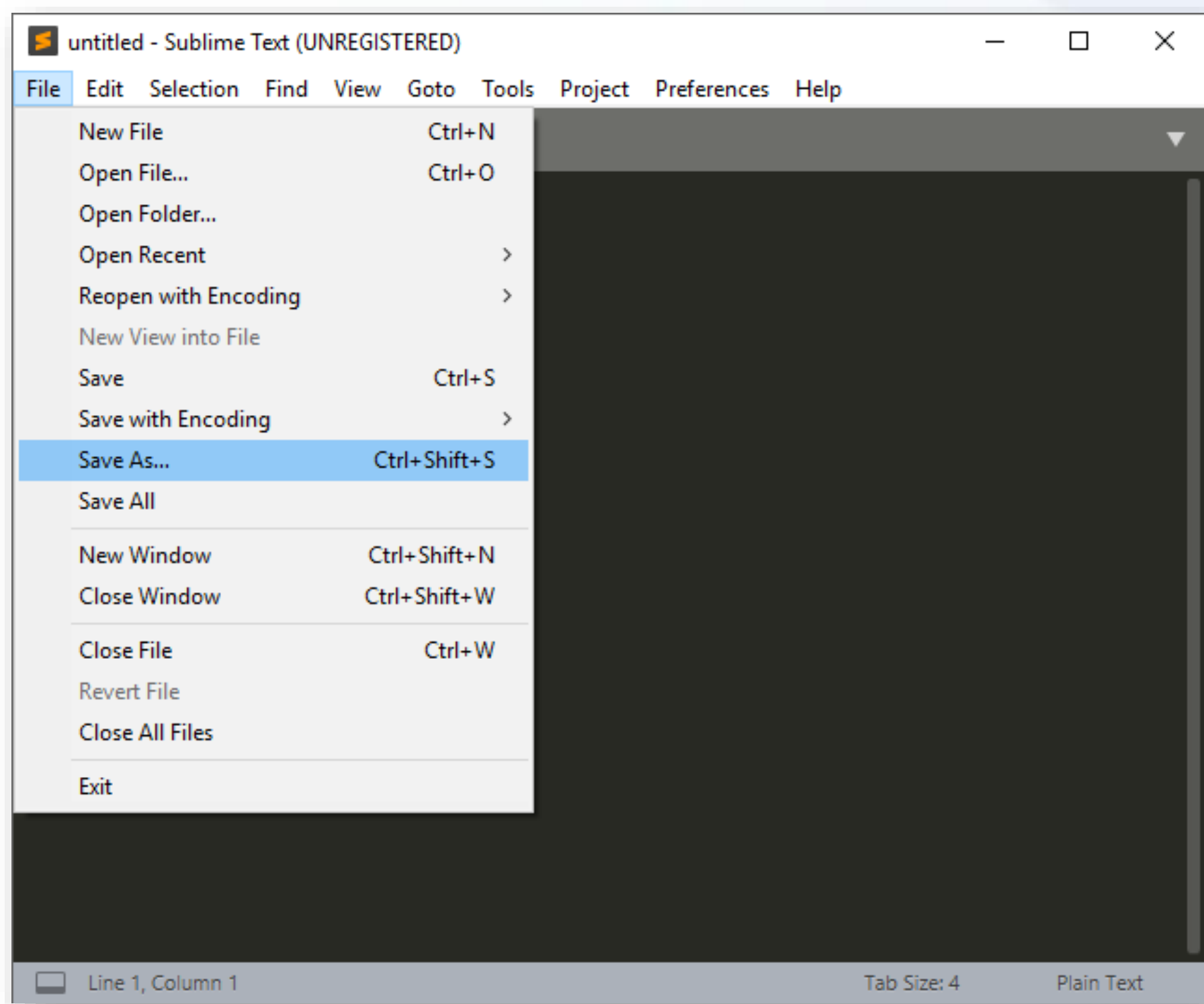


Atom



VS code

# Создание первой страницы

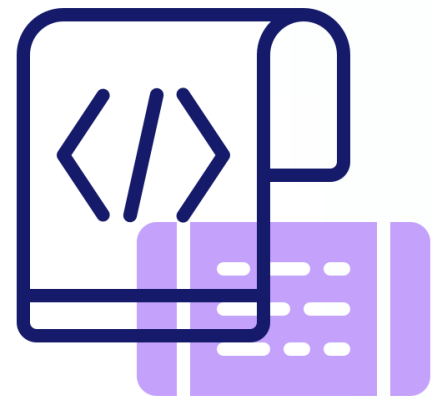


Кликаем по меню "Файл", выбираем "Сохранить как.." в строке "Имя файла" пишем: index.html



# HTML теги

HTML — это язык разметки: именно метки или как их еще называют теги (tags), заключенные в угловые скобки, формируют те элементы, из которых и состоит код HTML.



## ПАРНЫЕ

Парные теги могут содержать текст и другие теги. У парных тегов, в отличие от одиночных, есть вторая половинка — закрывающий тег. В закрывающих тегах перед именем ставится символ / («слэш»):

```
<h1>Текст заголовка </h1>
```



## ОДИНОЧНЫЕ

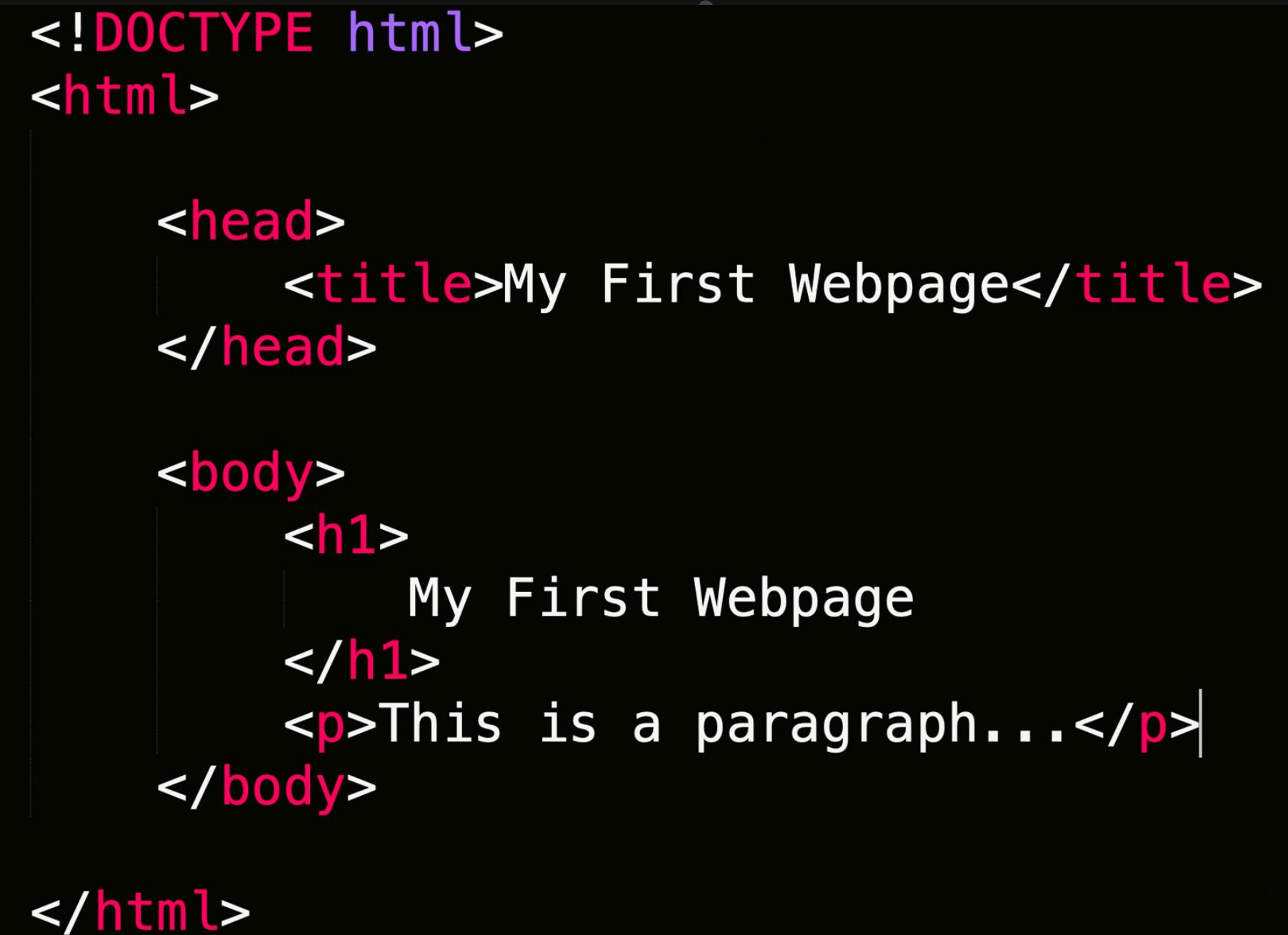
Например, тег <link> — одиночный:

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

# Основные функциональные понятия HTML

## HTML-структура

- Тег `<!DOCTYPE html>` — предназначен для указания типа текущего документа.
- Тег `<html> </html>` — это элемент самого высокого уровня, который охватывает каждую HTML-страницу.
- Тег `<head> </head>` — содержит метаинформацию, такую как заголовок страницы и кодировка.
- Тег `<body> </body>` — содержит всё содержимое, отображаемое на странице.



```
<!DOCTYPE html>
<html>

  <head>
    <title>My First Webpage</title>
  </head>

  <body>
    <h1>
      My First Webpage
    </h1>
    <p>This is a paragraph...</p>
  </body>

</html>
```



### Тег `<meta charset="utf-8">`

Кодировку HTML-страницы нужно указывать для того, чтобы веб-браузер мог правильно отображать текст на странице. Если браузер неправильно угадает кодировку, то вместо текста будут отображаться иероглифы.



### Тег `<title> ... </title>`

Если вы хотите дать название документу, чтобы оно появлялось в верхней строке браузера, это нужно сделать в разделе "head", используя тег `<title>`



### Тег `<p> ... </p>`

Одним из основных элементов страницы являются абзацы. Их можно сравнить с абзацами в книге - каждый абзац начинается с новой строки и имеет так называемую красную строку



# ОСНОВЫ viewport

Мобильные браузеры отображают страницы в виртуальном «окне» (viewport), которое, как правило, шире экрана устройства. Поэтому им не нужно сжимать макет каждой страницы в крошечное окно.



Свойство width определяет размер окна просмотра. Он может быть установлен на определенное количество пикселей, скажем, width=600 или на специальное значение device-width, которое означает ширину экрана в пикселях CSS в масштабе 100%. Свойство initial-scale контролирует уровень масштабирования при первой загрузке страницы. Свойства maximum-scale, minimum-scale и user-scalable определяют, как пользователям разрешено увеличивать или уменьшать страницу.

**Типичный сайт, оптимизированный для мобильных устройств, содержит следующий мета-тег**

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

# Заголовки в HTML

h1, h2, h3, h4, h5, h6

Заголовки создаются с помощью тегов `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`. Они имеют разную степень важности. В заголовке **h1** следует располагать название всей HTML страницы, в **h2** - название блоков страницы, в **h3** - название подблоков и так далее.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Это заголовок тайтл</title>
  </head>
  <body>
    <h1>Заголовок h1</h1>
    <h2>Заголовок h2</h2>
    <h3>Заголовок h3</h3>
    <h4>Заголовок h4</h4>
    <h5>Заголовок h5</h5>
    <h6>Заголовок h6</h6>
    <p>Это первый абзац.</p>
    <p>Это второй абзац.</p>
    <p>Это третий абзац.</p>
  </body>
</html>
```



## Тег `<b>...</b>`

- ✓ Тег `<b>` информирует браузер, что весь текст между тегом `<b>` и `</b>` должен быть напечатан жирным шрифтом.
- ✓ "b" это сокращение для «**bold**».
- ✓ `<strong>текст</strong>` (указывает, что текст надо выделить).



## Тег `<i>...</i>`

- ✓ Тег `<i>` информирует браузер, что весь текст между `<i>` и `</i>` должен быть напечатан курсивом.
- ✓ "i" это сокращение для «**italic**».



## Тег `<u>...</u>`

- ✓ Тег `<u>` информирует браузер, что весь текст между `<u>` и `</u>` должен быть подчеркнутым.
- ✓ "u" это сокращение от слова «**underlined**».





# Одиночные теги

Элемент состоит только из открывающего тега. Тег для вставки изображения или тег для вставки разделительной полосы добавляют на страницу одиночный объект, и им не нужно для этого заключать в себя какой-то текст. Поэтому их называют одиночными.

`<!doctype>`

`<meta>`

`<hr>`

`<link>`

`<br>`

`<img>`

# Одиночные теги



## Новая строка

Тег `<br>` устанавливает перевод строки в том месте, где этот тег находится.



## Горизонтальная линия

Тег `<hr>` рисует горизонтальную линию, которая по своему виду зависит от используемых параметров. Он относится к блочным элементам, линия всегда начинается с новой строки, а после нее все элементы отображаются на следующей строке.



# Изображения

Тег <img>

- Вы можете вставить любое изображение в веб-страницу с помощью **тега <img>**
- С помощью **атрибута "src"** - можно указать адрес картинки, которую Вы хотите поставить на сайт.
- Делается это так: 
- С помощью **атрибутов height и width** Вы можете изменять размер картинки.
- Пример 

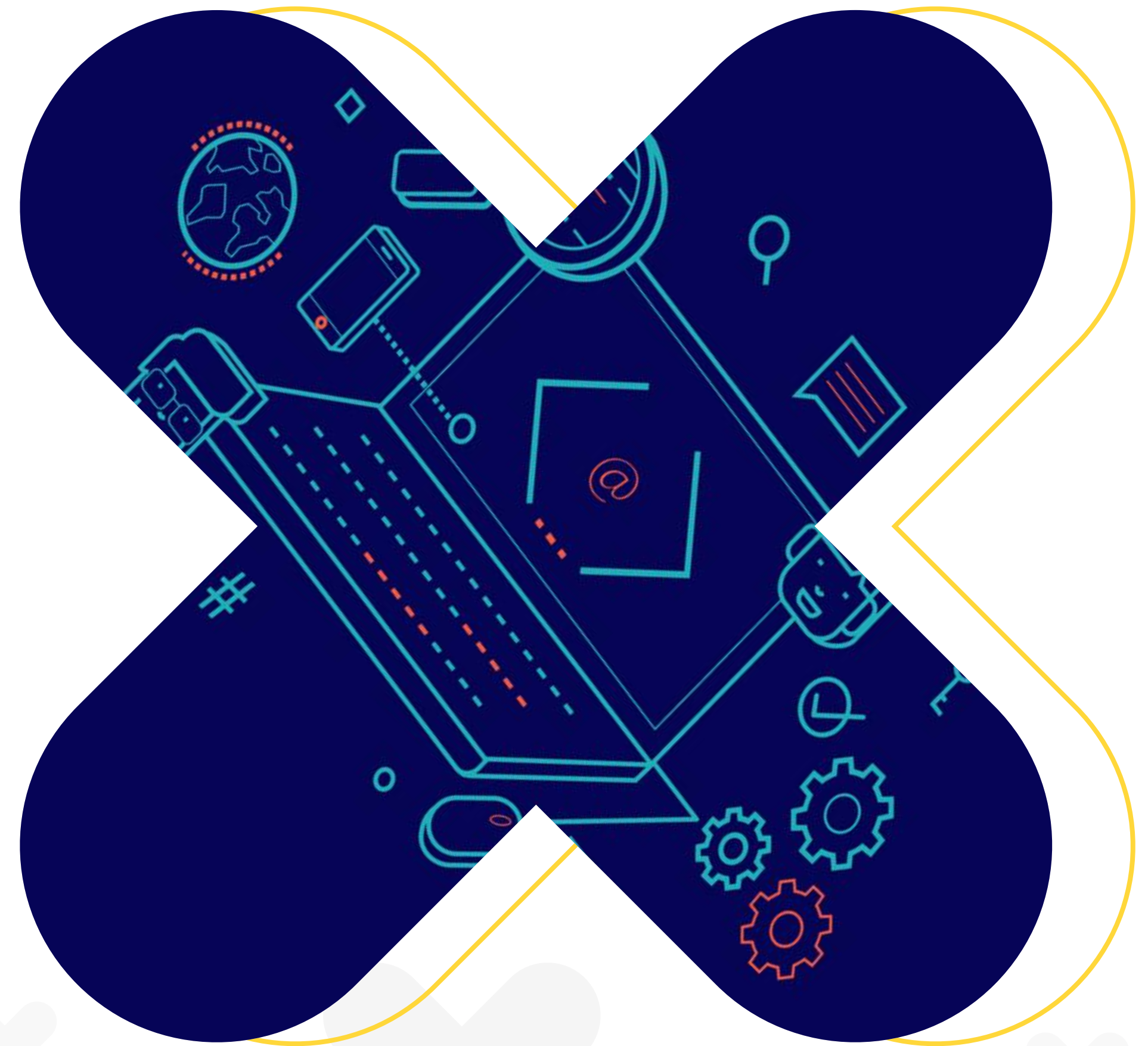




# Комментарий

---

- Тег `<!-- -->` добавляет комментарий в код документа.
- Текст комментария не отображается на странице.
- Разрешается внутри комментария добавлять другие теги.
- Пример: `<!-- текст -->`





# Введение в CSS3



# Каскадные таблицы стилей

---

- CSS расшифровывается как «каскадные таблицы стилей». Этот язык отвечает за внешний вид HTML-страницы.
- Синтаксис языка достаточно прост: он состоит из селекторов и свойств.
- С помощью селекторов можно сказать браузеру какие именно элементы мы хотим оформить. Свойства описывают как именно мы хотим оформить эти элементы.

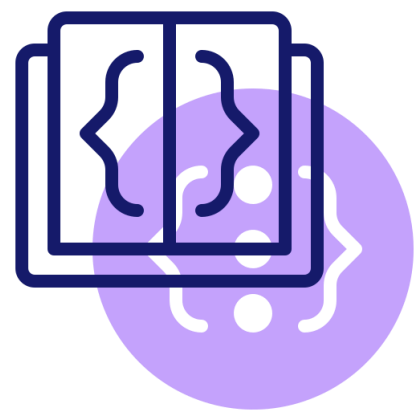


# Свойства

В CSS существует огромное количество свойств. Их можно разбить на следующие группы:

- ✓ оформление текста;
- ✓ работа с размерами и отступами;
- ✓ позиционирование элементов;
- ✓ создание сеток;
- ✓ декоративные: цвета, фон, тени;
- ✓ другие.





## Строчное подключение

---

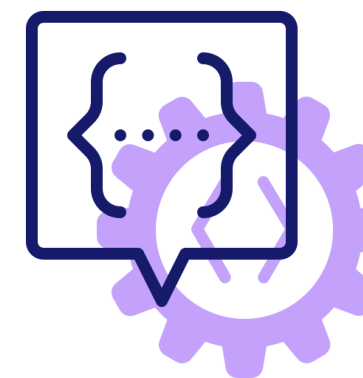
- ✓ При внутриэлементном или строчном подключении стиля, он определяется непосредственно внутри тега HTML при помощи атрибута `style`.



## Встроенное подключение

---

- ✓ Встроенные или внутридокументные стили используются для определения стилей элементов целой веб-страницы.
- ✓ В теге `<head>` определяется тег `<style>`, в котором задаются все стили для веб-страницы.



## Внешние подключение

---

- ✓ Внешние стили используются для подключения одновременно ко многим страницам сайта.
- ✓ Обычно это отдельный CSS файл. CSS файл имеет расширение `.css`

**Селектор** {  
    свойство1: значение;  
    свойство2: значение;  
}



## CSS-правила

- Селекторы – это имена стилей. У каждого селектора есть "свойства", определяемые внутри фигурных скобок.
- Свойства – это ключевые слова, которым присвоены определенные значения.



# Простейшие селекторы

**Простейшие селекторы - селекторы по именам тегов.**

Селекторы по тегам содержат имя тега без символов < и > и применяются ко всем подходящим тегам.

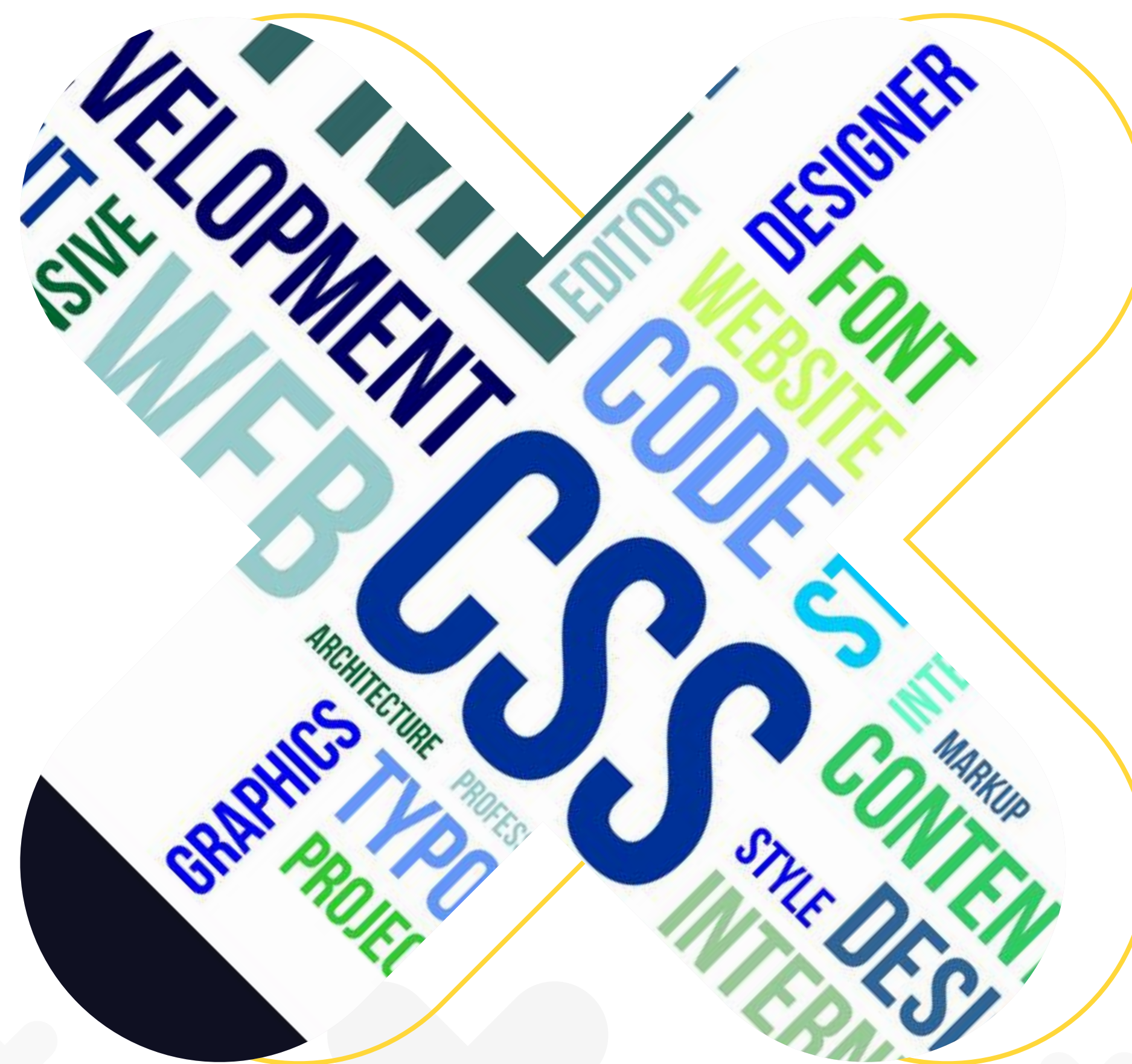


# Селекторы классов

Класс — это один из атрибутов тегов.

Запомните несколько правил написания классов:

- В CSS перед названием селектора класса обязательно ставится точка (но при присвоении класса в HTML-документе эта точка не нужна);
- В названии классов можно использовать только буквы латинского алфавита, дефис, символ подчеркивания, цифры;
- Название класса всегда должно начинаться с буквы (правильные варианты названий: .intro, .img-border, .nav\_menu\_01; неправильные: .2color, .-link, .\_divider);

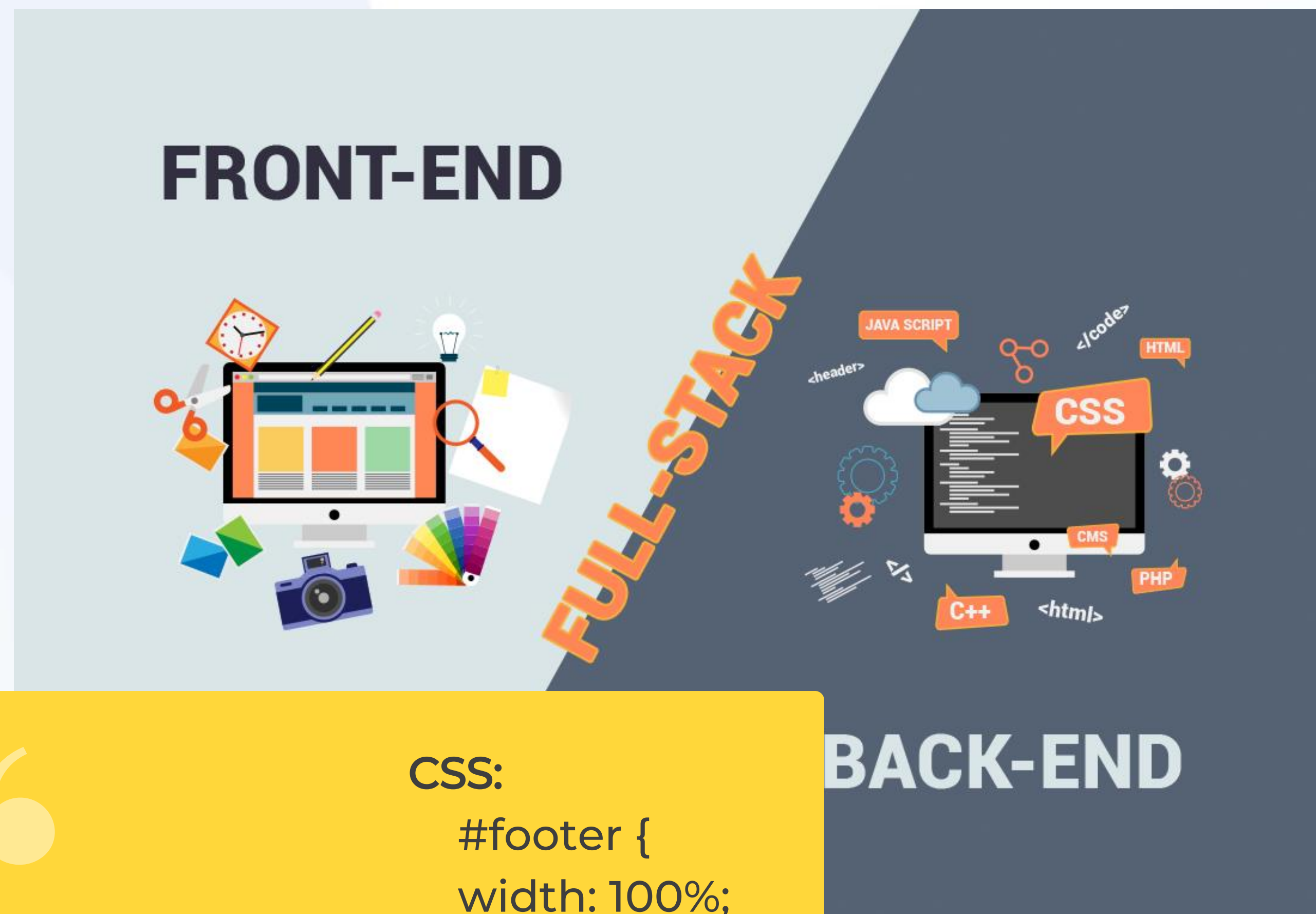




# Селекторы ID

## Идентификатор

- ✓ Идентификатор определяет уникальное название элемента.
- ✓ ID – это уникальное название элемента на веб-странице, которое должно встречаться на ней только один раз.
- ✓ Идентификаторы позволяют получить быстрый доступ к элементу DOM из скриптов.
- ✓ Идентификатор имеет больший вес, чем класс, поэтому, если элементу присвоен и ID, и класс, предпочтение отдается ID.
- ✓ С помощью идентификаторов можно ставить якорные ссылки на определенные элементы веб-страницы.



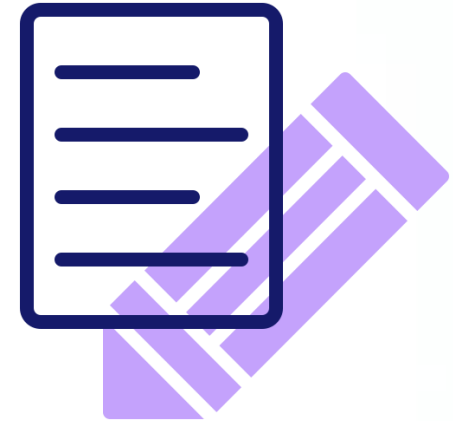
“

HTML:

```
<div id="footer">  
</div>
```

CSS:

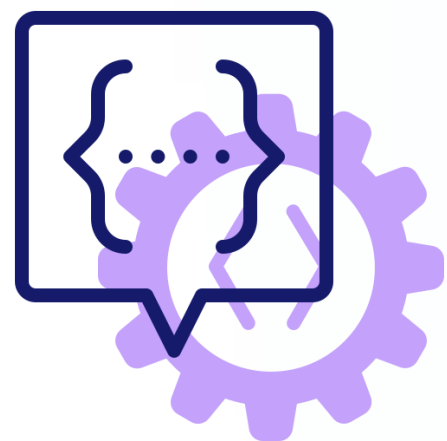
```
#footer {  
  width: 100%;  
}
```



## Шрифт

- Семейство шрифтов **font-family**  
Свойство используется для выбора начертания шрифта.

**font-family:** Arial;



## Размер шрифта

- Размер шрифта **font-size**  
Свойство определяет размер шрифта.

**font-size:** 12px;



## Цвет шрифта

- Цвет шрифта **color**  
Свойство задаёт цвет шрифта с помощью различных систем цветопередачи.

**color:** red;



# Цвет фона

- Цвет фона background-color.
- Свойство устанавливает цветную заливку для любого HTML элемента.
- Для блочных элементов цвет фона распространяется на всю ширину и высоту блока элемента, для строчных —только на область их содержимого.



Пример 1:

```
p{  
background-color: red;  
}
```



Пример 2:

```
h1{  
background-color: #ccc;  
}
```

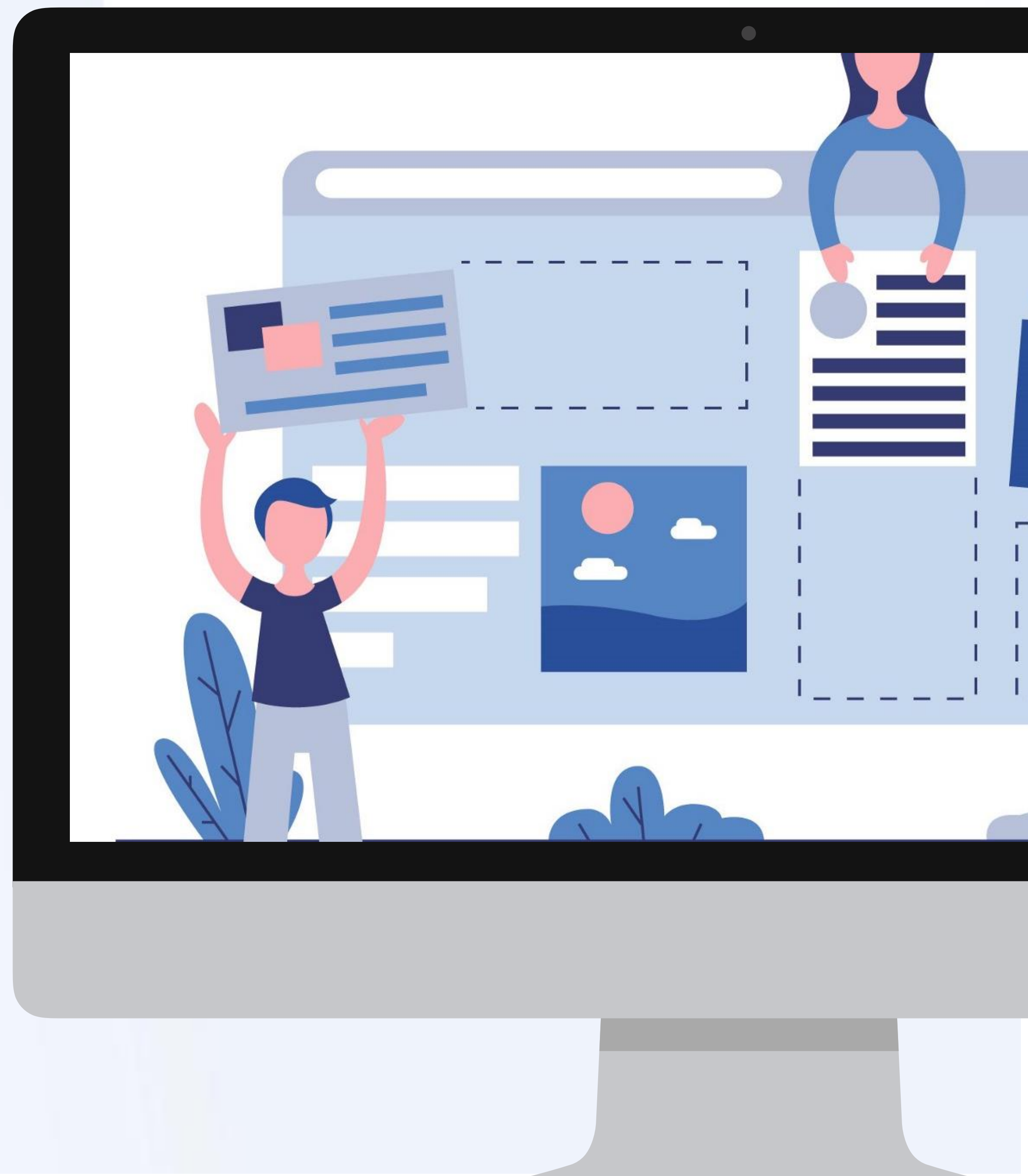
# Фоновое изображение

- ❑ Для создания заднего фона вашей HTML страницы может послужить любая картинка!

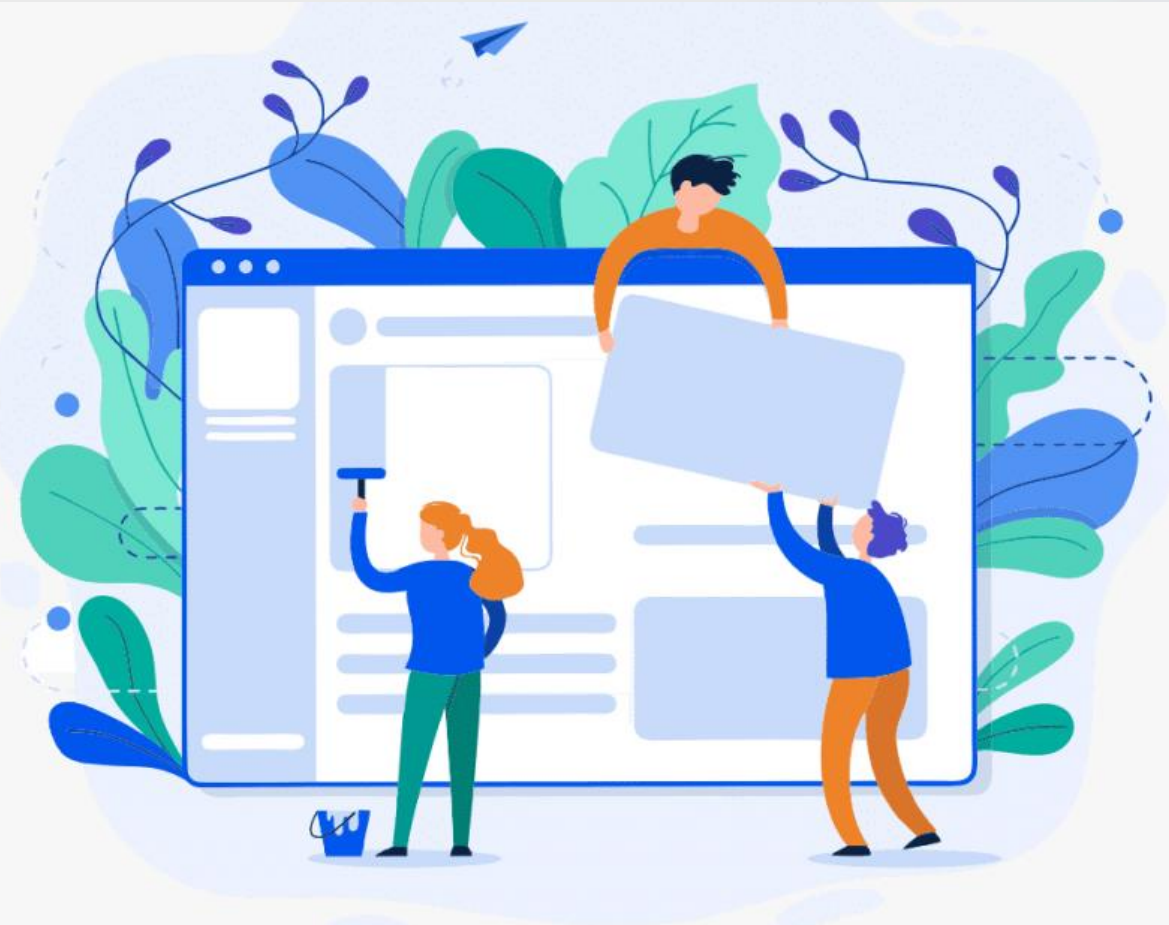
background-image

- ❑ Свойство устанавливает изображение в качестве фона для элемента .

```
body{  
    background-image: url(flower.png);  
}
```



# Фоновое изображение



Повтор фоновых  
изображений

**Background-repeat**



Позиционирование  
фоновых изображений

**Background-position**



Размер фоновых  
изображений

**Background-size**



# Background-repeat

Определяет, как будет повторяться фоновое изображение. Можно установить повторение рисунка только по горизонтали, по вертикали или в обе стороны.

**repeat** Весь фон страницы будет заполнен фоновым рисунком.

**no-repeat** Фоновое изображение не будет повторяться.

**repeat-x** Фоновый рисунок повторяется по Оси x.

**repeat-y** Фоновый рисунок повторяется по Оси y.

**space** Между картинками добавляется пустое пространство.

**round** Изображение повторяется так, чтобы в области поместилось целое число рисунков.



repeat-x



repeat-y



repeat



round



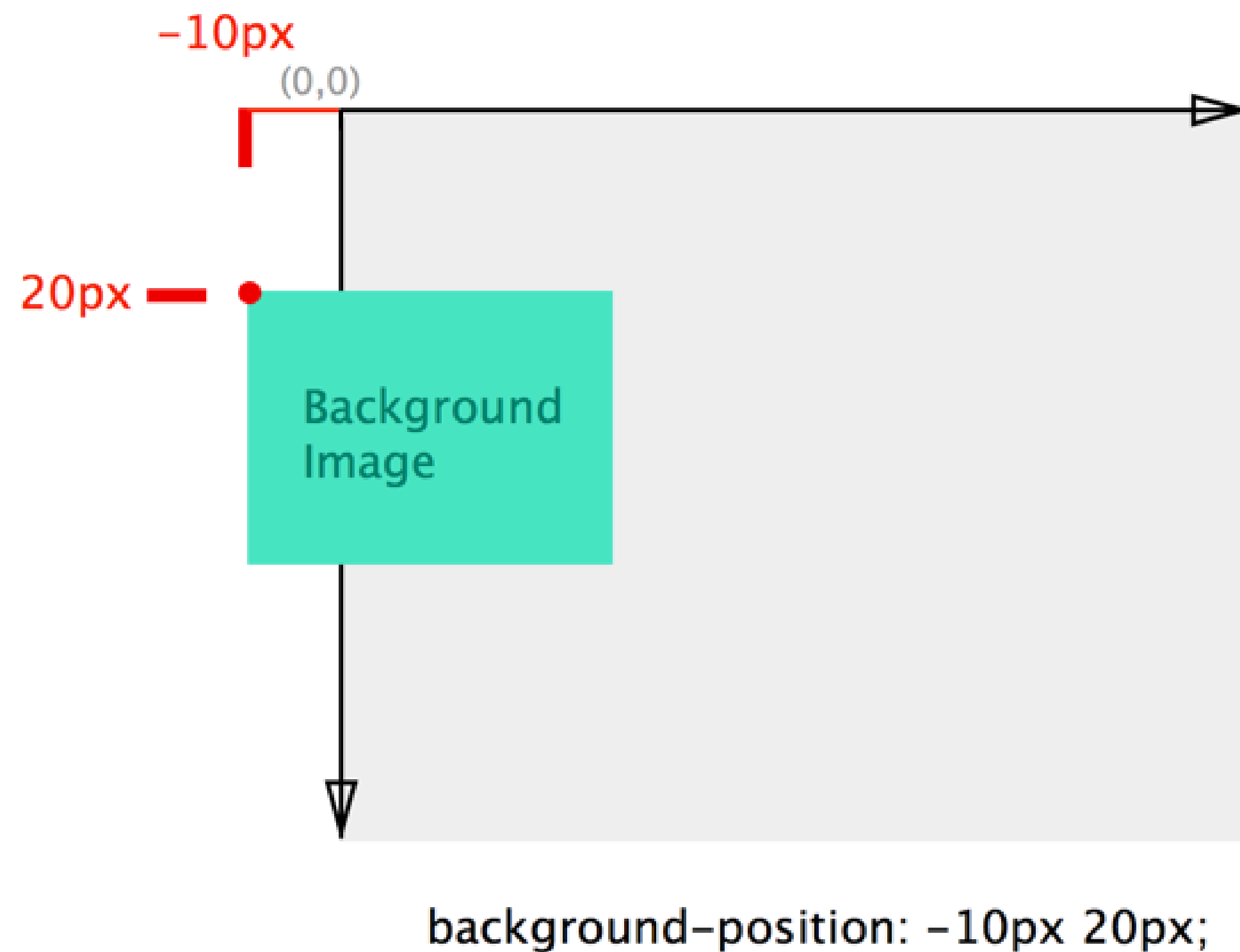
no-repeat



space



# Background-position



- По умолчанию фоновое изображение будет повторяться до бесконечности. Вы можете указать исходное положение, выбрав горизонтальное значение x и вертикальное y.

Для каждой координаты вы можете использовать:

- ✓ значения в пикселях;
- ✓ проценты, относительно размеров элемента HTML;
- ✓ ключевые слова, такие как center, left, bottom и др.

# Background-size

**Cover** Масштабирует изображение с сохранением пропорций так, чтобы его ширина или высота равнялась ширине или высоте блока.

**Contain** Масштабирует изображение с сохранением пропорций таким образом, чтобы оно целиком поместилось внутри блока.

