



# Full-stack

GEN AI





- HTML
- CSS
- Javascript

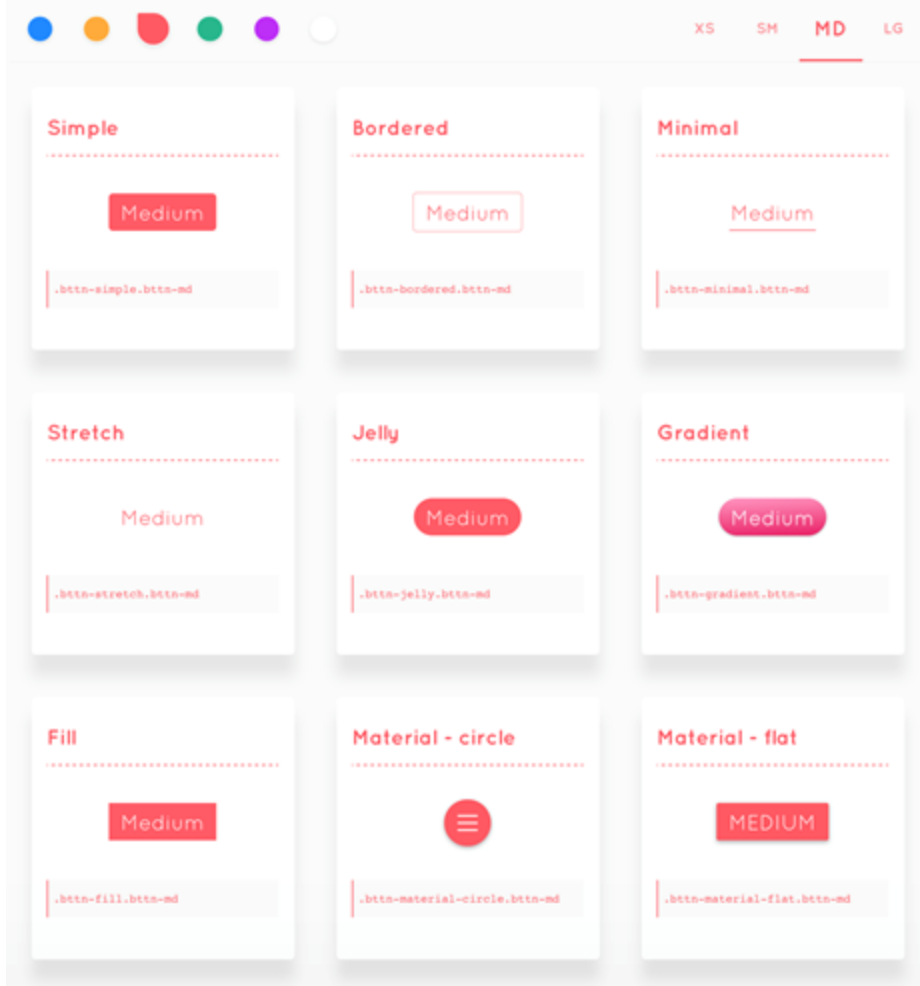


# CSS

CSS позволяет нам определять, как отображать (рендерить) информацию документа, хранящуюся в HTML.

Благодаря CSS мы можем контролировать все аспекты визуализации и некоторые другие функции:

- **Цвета**: содержимое, фон, границы
- **Поля**: внутреннее поле, внешнее поле
- **Положение**: где разместить
- **Размеры**: ширина, высота





## Пример CSS

```
* {  
    цвет: синий; /*комментарий */  
    margin: 10px;  
    font: 14px Tahoma;  
}
```

Это изменит все теги на моем веб-сайте ( «\*» означает все) на **синий** цвет с шрифтом **Tahoma 14px** и оставляя поле **10px** вокруг.

# CSS: как добавить



Существует четыре способа добавить **правила CSS** на ваш веб-сайт:

- Вставка кода в тег style

```
<style>  
    p { color: blue }  
</style>
```

- Ссылка на внешний CSS-файл

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

- Использование атрибута style в теге

```
<p style="color: blue; margin: 10px">
```

- Использование Javascript (об этом мы поговорим позже).

# Поля CSS



Ниже приведен список наиболее распространенных полей CSS и примеры их использования:

- `color: #FF0000; red; rgba(255,00,100,1.0);` //различные способы указания цветов
- `background-color: red;`
- `background-image: url('file.png');`
- `font: 18px 'Tahoma';`
- `border: 2px solid black;`
- `border-top: 2px solid red;`
- `border-radius: 2px;` //для сглаживания углов и придания им более округлой формы
- `margin: 10px;` //расстояние от границы до внешних элементов
- `padding: 2px;` //расстояние от границы до внутренних элементов
- `width: 100%; 300px; 1.3em;` //множество различных способов указания расстояний
- `height: 200px;`
- `text-align: center;`
- `box-shadow: 3px 3px 5px black;`
- `cursor: pointer;`
- `display: inline-block;`
- `overflow: hidden;`

# CSS-селекторы



Начнем с изменения цвета фона одного тега нашего веб-сайта:

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

Это правило CSS означает, что каждый тег DIV, найденный на нашем веб-сайте, должен иметь красный цвет фона. Помните, что DIV в основном используются для представления областей нашего веб-сайта.

Мы также можем изменить фон всего веб-сайта, воздействуя на тег body:

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

# CSS-селекторы



Что делать, если мы хотим изменить один конкретный тег (а не все теги одного типа)?

Мы можем указать более точные селекторы помимо имени тега. Например, по классу или идентификатору. Чтобы указать тег с заданным именем класса, мы используем точку:

```
p.intro {  
    color: red;  
}
```

Это повлияет только на теги p с именем класса intro:

```
<p class="intro">
```



# CSS-селекторы



Существует несколько селекторов, которые мы можем использовать для сужения наших правил до очень конкретных тегов нашего веб-сайта.

Основные селекторы:

- **имя тега**: просто имя тега
  - `p { ... } //влияет на все теги <p>`
- **точка (.)**: влияет на теги с ЭТИМ КЛАССОМ
  - `p.highlight { ... } //влияет на все теги <p> с class="highlight"`
- **знак решетки (#)**: указывает теги с ЭТИМ id
  - `p#intro { ... } //влияет на тег <p> с id="intro"`
- **две точки (:)**: состояния поведения (мышь наверху)
  - `p:hover { ... } //влияет на теги <p> с наведенной мышью`
- **скобки ([attr='value'])**: теги с атрибутом attr со значением 'value'
  - `input[type="text"] {...} // влияет на теги ввода типа text`

# CSS-селекторы



Вы также можете указать теги по их контексту, например: теги, которые находятся внутри тегов, соответствующих селектору. Просто разделите селекторы пробелом:

```
div#main p.intro { ... }
```

Это повлияет на теги `p` класса `intro`, которые находятся внутри тега `div` с id `main`

```
<div id="main">  
  <p class="intro">...</p> ← Влияет на этот  
</div>
```

```
<p class="intro">...</p> ← но не на этот
```

# CSS-селекторы



Вы можете комбинировать селекторы, чтобы еще больше сузить поиск.

```
div#main.intro:hover { ... }
```

применит CSS к любому тегу `div` с id `main` и классом `intro`, если курсор мыши находится **над ним**.

И вам не нужно указывать тег, вы можете использовать селекторы класса или id без тега, это означает, что он будет влиять на любой узел с id `main`

```
#main { ... }
```

# CSS-селекторы



Если вы хотите выбрать только элементы, которые являются прямыми потомками одного элемента (а не имеют предка с этим правилом), используйте символ `>`:

```
ul.menu > li { ... }
```

Наконец, если вы хотите применить одни и те же CSS-действия к нескольким селекторам, вы можете использовать символ запятой `,`:

```
div, p { ... } ← это будет применяться ко всем тегам div и p
```

# Модель блока



Важно отметить, что по умолчанию любая ширина и высота, указанные для элемента, не учитывают его отступы, поэтому `div` с шириной 100 пикселей и отступом 10 пикселей будет отображаться на экране размером 120 пикселей, а не 100 пикселей.

Это может стать проблемой, нарушающей ваш макет.

Вы можете изменить это поведение, изменив модель блока элемента, чтобы ширина использовала самую

## Box-Sizing

