



Full-stack

GEN AI





- HTML
- CSS
- Javascript



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



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
```

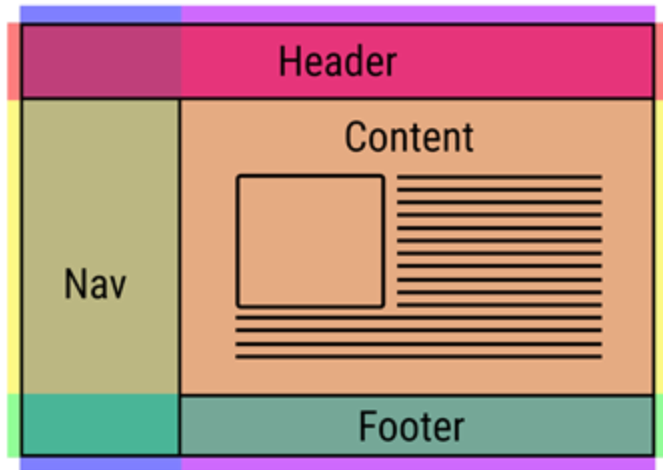
Макет



Одной из самых сложных частей CSS является интерпретация макета вашего веб-сайта (структуры внутри окна).

По умолчанию HTML имеет тенденцию размещать все в одной колонке, что не является идеальным решением.

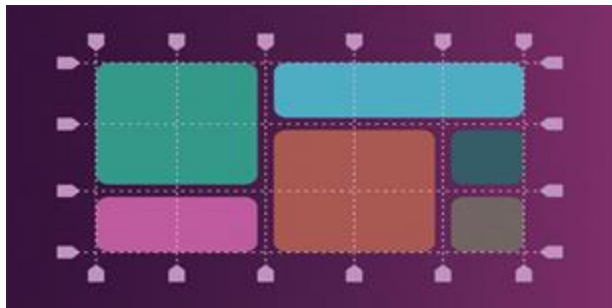
В CSS было много предложений по решению этой проблемы (таблицы, фиксированные div, flex, grid и т. д.).



Система сетки

Поскольку большинство сайтов структурированы в виде сетки, я рекомендую использовать систему CSS Grid.

Мы просто указываем, сколько строк/столбцов должен использовать div из основной сетки, и он будет автоматически распределяться.



HTML

```
<div class="grid-container">  
  <div class="grid-item1">1</div>  
  <div class="grid-item2">2</div>  
</div>
```

CSS

```
.grid-container {  
  display: grid;  
  grid-template-rows: 100px 100px;  
  grid-template-columns: 100px 100px 100px;  
  grid-gap: 5px;  
}  
  
.grid-item1 {  
  background: blue;  
  border: black 5px solid;  
  grid-column-start: 1;  
  grid-column-end: 5;  
  grid-row-start: 1;  
  конец строки сетки: 3;  
}
```

Полноэкранные div

Иногда нам нужно, чтобы div занимал весь экран (для создания веб-приложения), а не прокручивался как обычный документ.

В этом случае не забудьте использовать проценты для определения размера элементов, но имейте в виду, что проценты относятся к размеру родительского элемента, поэтому вы должны установить размер элементов `<body>` и `<html>` на 100%.

CSS

```
html, body {  
    width: 100%;  
    height: 100%;  
}  
  
div {  
    margin: 0;  
    отступ: 0;  
}  
  
#main {  
    width: 100%;  
    высота: 100%;  
}
```