

CSS

Как задать оформление страницы ?

HTML, в отличие от XML, обладает семантикой, т.е. браузер знает как отображать тот или иной HTML тэг. В HTML есть тэги для управления внешним видом (`color` , `font` , `big` , ...), но их возможности явно недостаточны.

Решение - описывать внешний вид отдельно от структуры документа с помощью языка **Cascading Style Sheets**.

СИНТАКСИС CSS

```
.mid-play {  
    padding: 13px 0px 0px 13px;  
}  
p.inner-play a {  
    color: #3c3c3c;  
    text-decoration: underline;  
}  
.big-top {  
    background-image: url(/img/pc/220_130_top.gif);  
}  
  
/* комментарии: селектор { имя_стиля1: значение1; } */
```

Где могут быть заданы стили?

- Встроенные в браузер стили
- Во внешнем файле

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

лучше: reuse, читаются сразу сверху,
можно кешировать

- В коде HTML документа

```
<style>...</style>
```

- Стили могут быть привязаны к конкретному тэгу

```

```

Какие бывают стили ?

- `width`, `height` — размеры элемента
- `margin`, `padding` — границы и отступы
- `display`, `visibility` — режим отображения
- `top`, `left`, `right`, `bottom` — расположение
- `background` — фон элемента
- `font` — управление шрифтом
- `text-align` — выравнивание текста

CSS селекторы

Классы и идентификаторы

```
<div id="userpic"></div>  
<button class="btn btn-main">Одобрить</div>  
<button class="btn">Написать комментарий</div>
```

- `id` - идентификатор элемента, должен быть уникален на странице
- `class` - список классов элемента, классы могут повторяться

если навесить один класс на несколько элементов, они будут отображаться одинаково

Базовые селекторы

- Универсальный селектор

```
* { margin: 0px; padding: 0px; border: 0px; }
```

- Имена тэгов

```
p { margin-top: 10px; }
```

- Имена классов (с точки)

```
.btn { border: solid 1px gray; }
```

- id тэгов (с решетки)

```
#userpic { padding: 10px }
```


Сложные селекторы

- контекстные (вложенные)

`div.article a { text-decoration: underline }` будет применяться только к тем ссылкам, которые вложены в тег `div` класса `article`

- дочерние (вложенность = 1 уровень)

`a > img { border: 2px }` повлияем на все картинки внутри тега `a`

- соседние

`h2.sic + p { margin-left: 30px }` Комбинатор '+' выбирает элемент, который находится непосредственно после указанного элемента, если у них общий родитель.

Синтаксис: A + B

- группировка

`h1, h2 { color: red }`

Пример: селектор `h2 + p` выберет первый элемент `<p>`, который находится непосредственно после элемента `<h2>` (en-US).

Псевдоклассы

- `a:visited` — посещенная ссылка
- `a:link` — непосещенная ссылка
- `div:hover` — элемент при наведении мыши
- `input:focus` — элемент при получении фокуса
- `li:first-child` — выбирает первого потомка среди множества элементов

Псевдоэлементы

- `#el::after` — виртуальный элемент после `#el`
 - `#el::before` — виртуальный элемент перед `#el`
- текст, пункты
etc

```
.jack-sparrow::before {  
    content: "Captain ";  
    display: inline;  
}  
  
.jack-sparrow:hover::before {  
    color: red;  
}
```

Наследование и приоритеты

Наследование стилей

```
<head>
  <style>
    body { color: darkgray; font-family: Arial; }
    p { font-size: 110% }
  </style>
</head>
<body>
  <p> Привет, <a href="/">Мир</a> </p>
</body>
```



Не все стили наследуются.

Приоритеты стилей

В случае, если два разных стиля конфликтуют между собой, применяется тот, что обладает большей **специфичностью**. Если специфичность двух стилей совпадает, применяется тот, что расположен **ниже** в HTML/CSS коде.

Указание в значение стиля флага `!important` позволяет перекрыть проверку специфичности.

Правила расчета специфичности

- id – 100
- классы и псевдоклассы – 10
- тэги и псевдоэлементы – 1

Так, например, селектор `ul.info ol + li` обладает специфичностью 13, а селектор `li.red.level` специфичностью 21 балл

Отображение элементов

Режимы отображения элементов

- `display: none` — элемент невидим, не занимает места
- `display: block` — элемент занимает максимальную ширину, начинается с новой строки, учитывает `width`, `height`
- `display: inline` — элемент занимает минимальную ширину, и не прерывает строку, игнорирует `width`, `height`
- `display: inline-block` — блочный элемент, но не разрывает строку, примерно как `img`

DIV vs. SPAN

`<div class="t">ONE</div>` для задания свойств блокам

`<div class="t">2</div>`

`ONE`

`2` задание свойств тексту

`<style>`

`.t {`

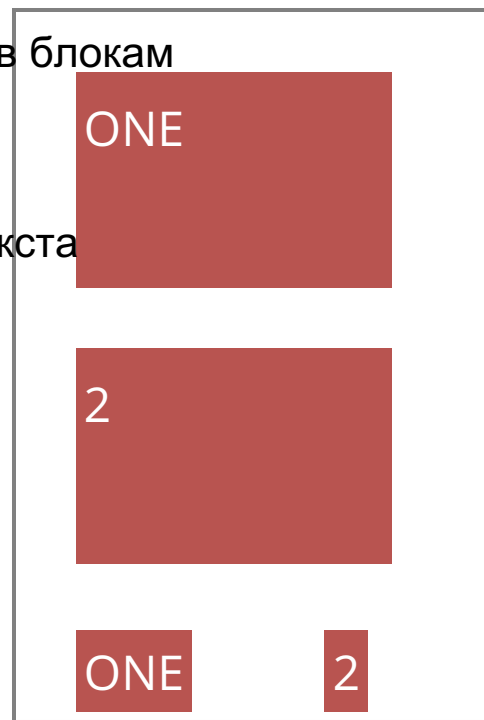
`width: 150px; height: 100px;`

`background: red; color: white;`

`margin: 30px; padding: 4px;`

`}`

`</style>`



float & clear



Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Proin tempor iaculis massa. Fusce sollicitudin purus
viverra erat sollicitudin placerat sit amet ut diam.

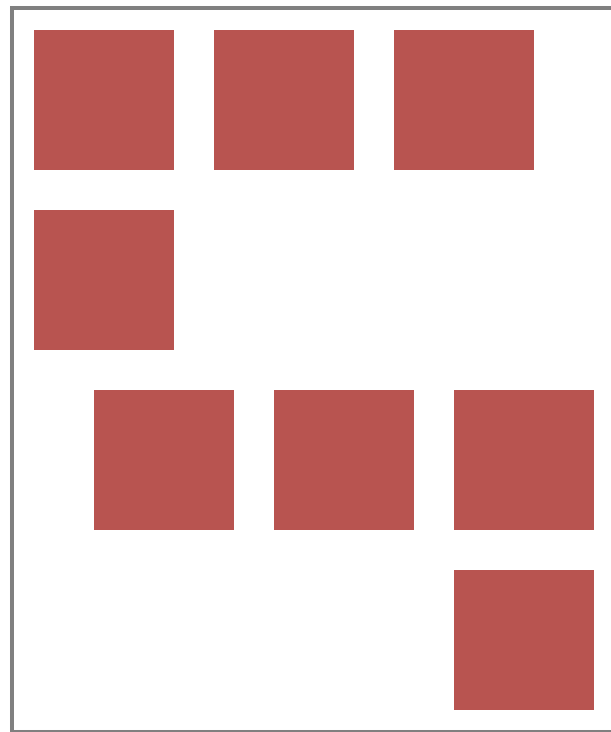
Vivamus malesuada tristique elit. Proin nec eros tempor.

`float: left` - всплывание влево, `float: right` - всплывание
вправо, `clear: both` - отменяет всплывание, «проводит черту»

без clear: both квадраты, ориентированные по правому краю,
могли бы следовать сразу за верхними квадратами

float & clear

```
<div class="outer">
  <div class="sqr fl"></div> ...
  <div class="clr"></div>
  <div class="sqr fr"></div> ...
</div>
<style>
.outer { float: left; width: 300px }
.sqr { width: 70px; height: 70px }
.fl { float: left; }
.fr { float: right; }
.clr { clear: both; }
</style>
```



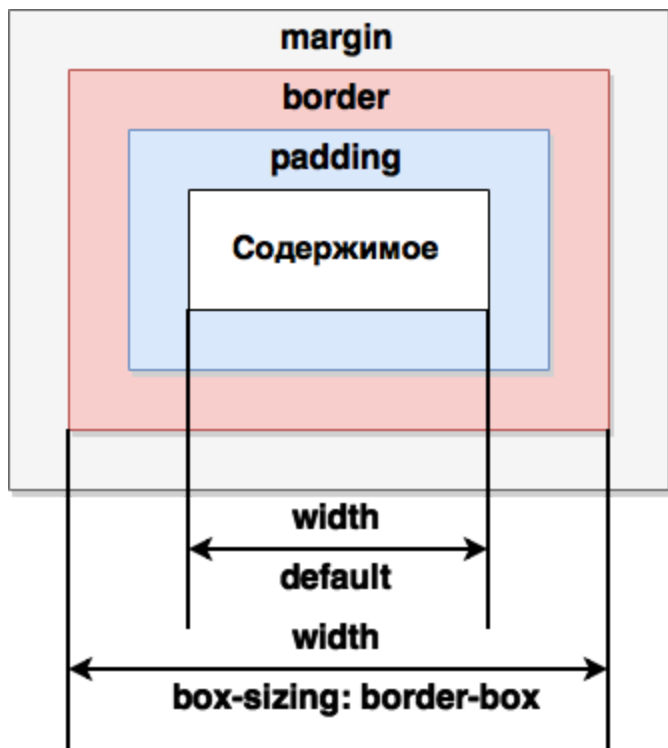
Позиционирование

- `position: static` — обычное расположение
- `position: relative` — смещение относительно начального местоположения на странице
- `position: absolute` — если родитель `relative`, `absolute` или `fixed` — относительно родителя, иначе - относительно начала документа
- `position: fixed` — относительно окна браузера
- `top/right/bottom/left` - отступы, могут быть отрицательными

Отступы и box-
model

Отступы

width: включает в себя padding и пр



Способы задания отступов:

```
margin: 10px;  
margin: 10px 5px;  
margin: 10px 2px 5px;  
margin: 1px 2px 3px 4px;  
margin-left: 10px;
```

Варианты box-sizing:

- content-box (default)
- border-box настройка границ блока

23

проблема наложения отступов(margin) - если блоки идут друг за другом, между ними будет общее значение margin-top, margin-bottom etc

Bootstrap

Что такое Bootstrap?

Bootstrap - это готовая библиотека стилей (CSS-фреймворк) от Twitter. Bootstrap позволяет быстро разработать приемлемый дизайн даже при базовых знаниях CSS.

Что включает в себя Bootstrap?

- Шаблон страниц
- Сетка
- Современные «стили по умолчанию»
- Верстка: таблицы, формы, списки, кнопки, ...
- Компоненты: навигация, меню, пагинатор, ...
- JavaScript плагины

Сетка Bootstrap

```
<div class="row">  
<div class="col-md-4 col-lg-2">LEFT</div>  
<div class="col-md-8 col-lg-10">CONTENT</div>  
</div>
```

md - medium, lg - малый
размер, адаптивные размеры

LEFT	CONTENT
------	---------