

# Занятие 2

Введение в CSS.  
Блочная верстка.

# Подключаем CSS стили

```
<head>
  <meta charset="UTF-8">
  <link href="style.css" rel="stylesheet">
  <title>My first page</title>
</head>
```

```
<body>
  <style>
    body{
      background: red;
    }
  </style>
</body>
```

## Структура файлов

```
/project_name
├── index.html
├── /assets
│   ├── /images
│   │   ├── logo.png
│   │   └── sprite.png
│   └── /styles
│       └── style.css
└── /fonts
    ├── openSans.ttf
    ├── openSans.eot
    ├── openSans.woff
    └── openSans.svg
```

# CSS селекторы

```
//По названию тега  
body{...}  
li{...}  
div ul li{...} // контекстный  
  
//По id  
#elem {...}  
div#elem {...}  
  
//По классу  
.element {...}  
div.element {...}  
.element .inner-part {...}  
  
//По атрибуту  
.element[href="#" ] {...}  
div.element[src="image.jpg"] {...}
```

```
//Для всех элементов  
* {...}  
li * {...}  
  
//Для всех на том же уровне  
.element ~ div {...}  
.element + a.link {...}  
  
//Только дочерние элементы  
div.element > ul {...}  
div.element > ul > li {...}  
  
//Перечисление свойств  
div.element > ul,  
div.element > ol {...}
```

# Псевдоклассы в CSS

```
// Стили для непосещенных ссылок
a:link {...}
a.logo-link:link {...}

// Стили для посещенных ссылок
a:visited {...}
a.logo-link:visited {...}

// Стили для активного элемента
.element:active {...}
a.logo-link:active {...}

// Стили для элемента, получающего фокус
.element:focus {...}
a.logo-link:focus {...}
```

```
// Реакция элемента по наведению
a:hover {...}
.element:hover {...}

// Отмена ховера для модификатора
.element {
  text-decoration: none;
  color: gray;
  cursor: pointer;
}
.element._active {
  text-decoration: underline;
  cursor: default;
}
.element:hover {
  text-decoration: underline;
  color: red;
}
.element._active:hover {
  color: gray;
}
```

# Каскадность

*// Механизм комбинирования стилей из разных источников  
в итоговый набор свойств и значений для каждого тега*

```
<p class="beloved-color">Зелёный - мой любимый цвет</p>
```

```
.beloved-color { color: green; }
```

*// Браузерные стили:*

```
p {margin: 1em 0; }
```

*// Итоговые стили для элемента:*

```
color: green; margin: 1em 0;
```

Специфичность селектора разбивается на 4 группы — a, b, c, d:

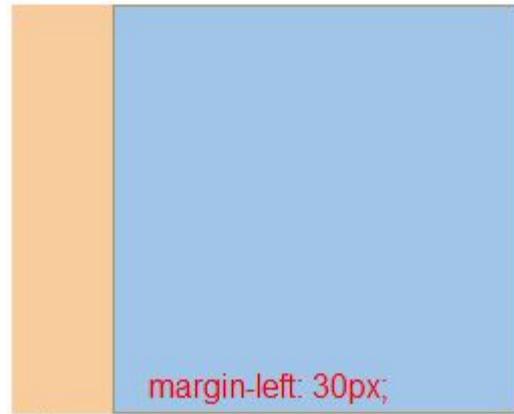
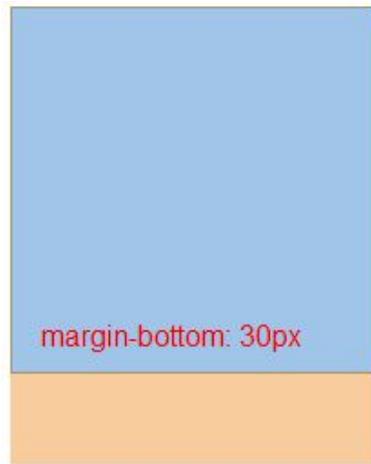
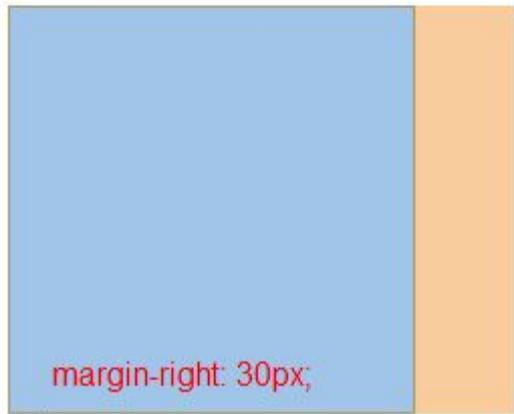
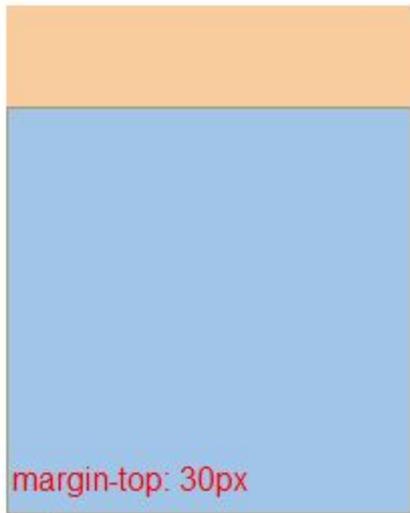
1. стиль определен как `style="..."`, то **a**=1, иначе **a**=0;
  2. значение **b** равно количеству идентификаторов (тех, которые начинаются с #) в селекторе;
  3. значение **c** равно количеству классов, псевдоклассов и селекторов атрибутов;
  4. значение **d** равно количеству селекторов типов элементов и псевдо-элементов.
- Универсальный селектор (\*), комбинаторы (+, >, ~, ') и отрицающий псевдокласс (:not()) не влияют на специфичность. (Однако селекторы, объявленные *внутри*.not(), влияют)
  - *important* важнее

пример	a	b	c	d	итого
	<code>style="..."</code>	<code>id</code>	<code>.class</code> <code>:hover</code> <code>[href=""]</code>	<code>p</code> <code>p::after</code>	
<code>p</code>	0	0	0	1	1
<code>#fox .red</code>	0	1	1	0	110
<code>div .red</code>	0	0	1	1	11
<code>a:hover</code>	0	0	1	1	11
<code>div::after</code>	0	0	0	2	2

# CSS свойства

```
//Записывается внутри фигурных скобок {...}  
//Состоит из конструкции ключ/значение разделенные двоеточием, и с точкой с  
запятой в конце  
//Один CSS селектор может содержать любое количество свойств  
//key: value;  
//width: 100px;
```

```
width: value | % | auto | calc();  
height: value | % | auto | calc();  
max-width: value | % | auto | calc();  
min-width: value | % | auto | calc();  
max-height: value | % | auto | calc();  
min-height: value | % | auto | calc();
```



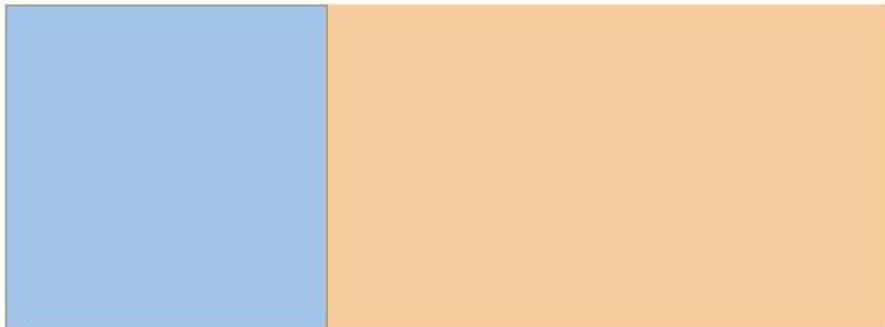
```
margin-top: 10px;  
margin-right: 20px;  
margin-bottom: 15px;  
margin-left: 20px;
```

```
margin: 10px 20px 15px 20px;  
margin: 10px 20px 15px;
```

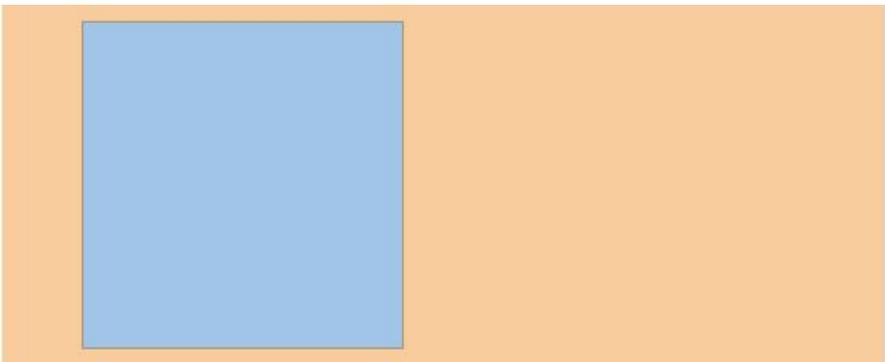
```
margin-top: 10px;  
margin-right: 20px;  
margin-bottom: 10px;  
margin-left: 20px;
```

```
margin: 10px 20px 10px 20px;  
margin: 10px 20px 10px;  
margin: 10px 20px;
```

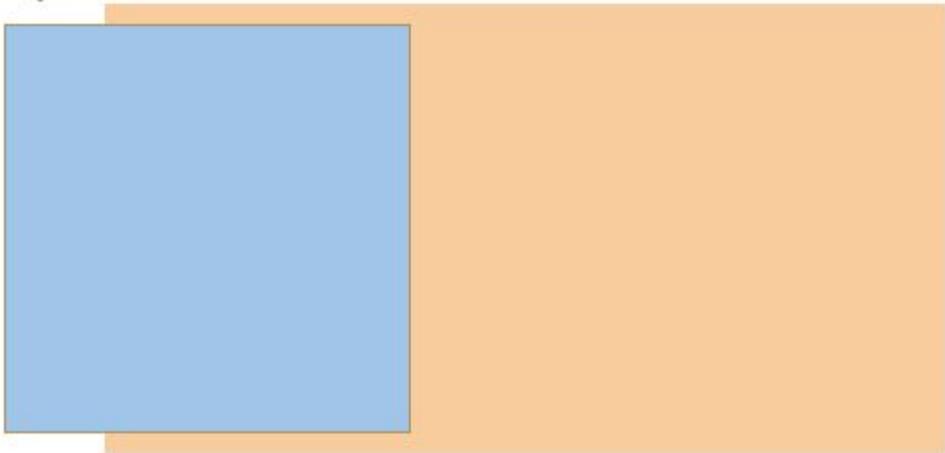
# margin для блочных элементов



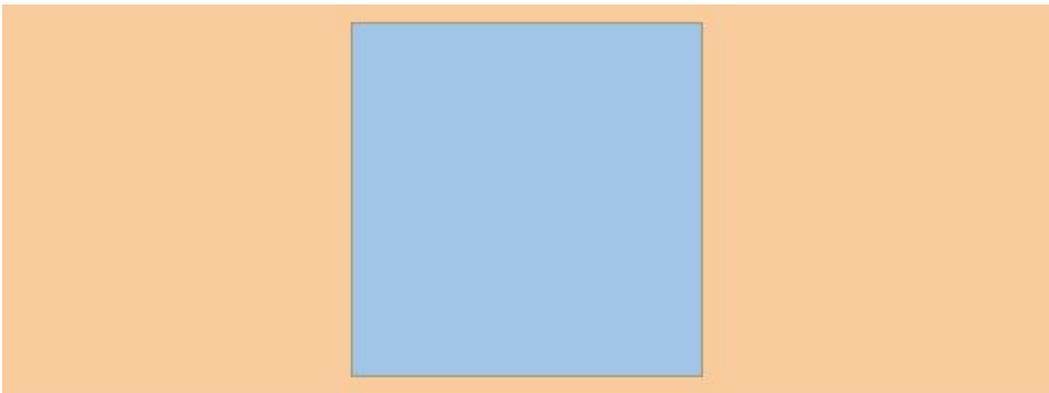
```
width: 200px;  
height: 200px;  
border: 1px solid;
```



```
width: 200px;  
height: 200px;  
border: 1px solid;  
margin: 10px 50px;
```



```
width: 200px;  
height: 200px;  
border: 1px solid;  
margin: 10px -50px;
```

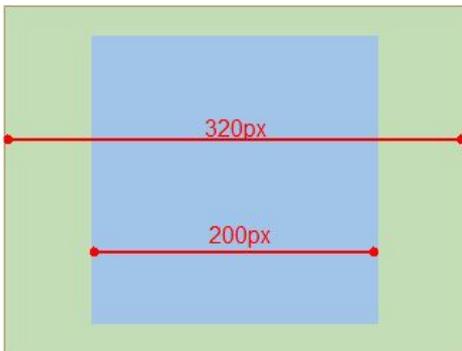


```
width: 200px;  
height: 200px;  
border: 1px solid;  
margin: 10px auto;
```

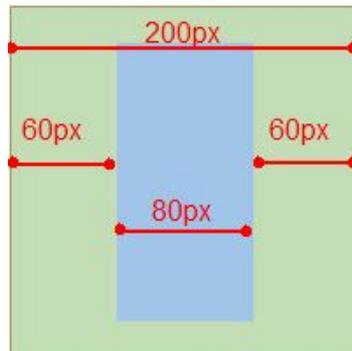
# Поля в CSS



```
height: 200px;  
border: 1px solid;  
padding: 20px 80px;
```



```
width: 200px;  
height: 200px;  
border: 1px solid;  
padding: 20px 60px;  
display: inline-block;
```



```
box-sizing: border-box;
```

## НАШИ ПРЕДЛОЖЕНИЯ

Участники клуба my life:) получают возможность пользоваться самыми разными эксклюзивными предложениями.



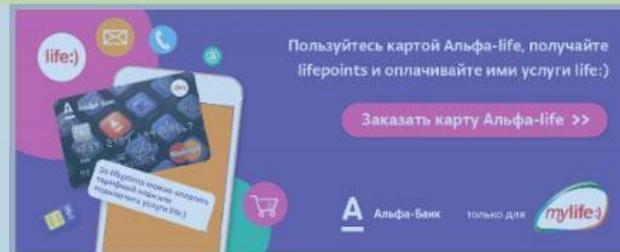
**МАТРАНГ**  
04.10  
Re:Public

**GUF & SLIM**  
12.10  
Re:Public

**Feduk**  
08.11  
Прайм Холл

**СКИДКА  
5 РУБЛЕЙ  
НА ЛЮБОЙ  
КОНЦЕРТ**

mylife:)



life:)

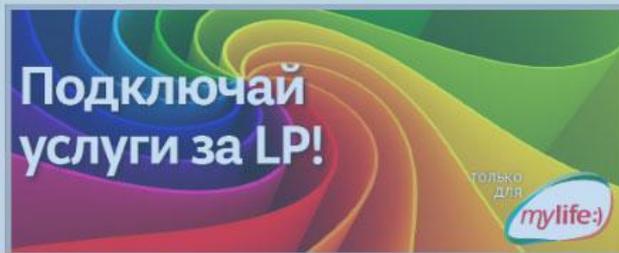
Пользуйтесь картой Альфа-life, получайте lifepoints и оплачивайте ими услуги life:)

[Заказать карту Альфа-life >>](#)

mylife:)

А Альфа-Банк только для mylife:)

mylife:)



**Подключай  
услуги за LP!**

только для mylife:)

# Рамки в CSS

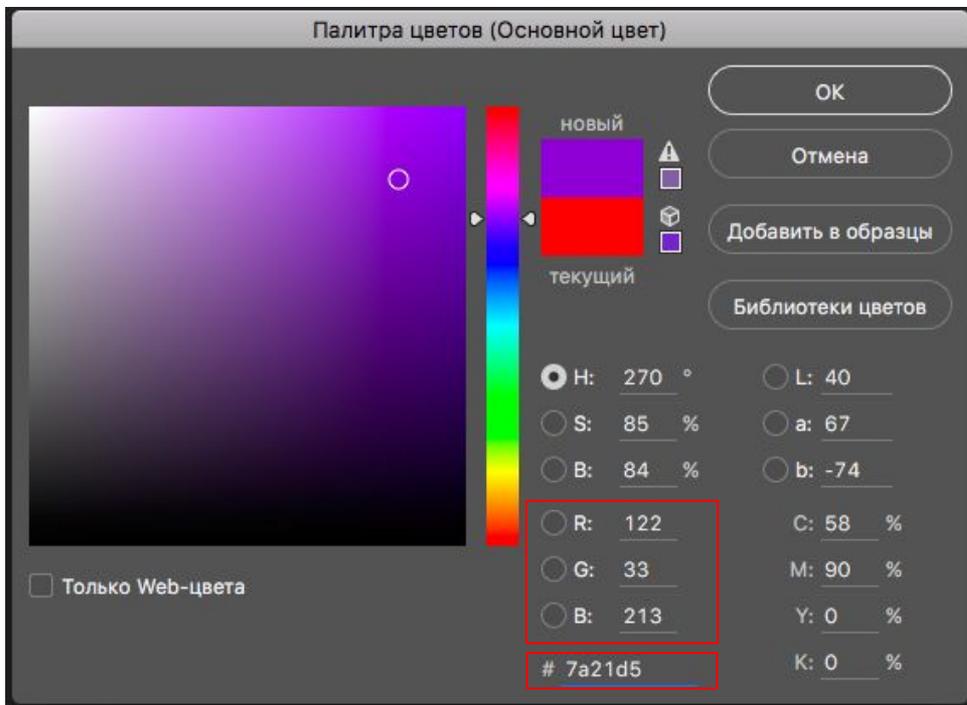
```
border-width: value;
border-style: solid | dashed | dotted;
border-color: color | HEX color | rgb color | rgba color;

border-top: 30px solid red;
border-right: 30px dotted #ccc || #4d5aa7;
border-bottom: 30px dashed rgb(0, 255, 216) || rgba(0, 255, 216, .8);
border-left: 30px solid transparent;

// Скругленные углы CSS3 свойство
border-radius: value | % | auto;

// Группировка свойств
border: [border-width || border-style || border-color];
```

# Цвета в CSS



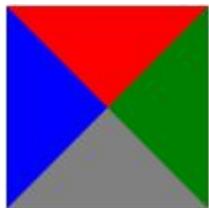
```
// HEX цвет обозначается хэштегом  
"#" и записывается в  
шестнадцатеричной системе счисления,  
белый цвет обозначается #ffffff или  
#fff, а черный #000000 или #000  
//#33aa55 === #3a5
```

```
//RGB цвет обозначается:  
//color: rgb('red', 'green', blue')  
//color: rgb(255,255,255) - белый  
//color: rgb(0,0,0) - черный
```

```
//RGBA отличается от RGB только  
четвертым параметром, который  
отвечает за прозрачность, где 1 -  
непрозрачная заливка, а 0 -  
полностью прозрачный цвет. Дробную  
часть можно указывать как 0.4, так и  
просто .4
```

```
//color: rgba(122, 33, 213, .6);
```

# Лайфхаки с border



```
border-top: 50px solid red;  
border-right: 50px solid green;  
border-bottom: 50px solid gray;  
border-left: 50px solid blue;  
width: 0;
```



```
border-top: 50px solid red;  
border-right: 50px solid transparent;  
border-left: 50px solid transparent;  
width: 0;
```

Более подробно на <https://habr.com/post/126207/>  
или по запросу “геометрия на CSS”

# Другие свойства для блоков

```
// Управляет контентом блока, обрезает видимую часть, или добавляет скролл, если контента больше, чем величина блока.  
overflow: auto | hidden | scroll | visible | inherit;  
// Задаёт стиль булетов для списков ul => li, для нумерованных списков читай документацию на http://htmlbook.ru  
list-style-type: circle | disc | square;  
// Позволяет скрывать и отображать блок, оставляя его в потоке  
visibility: visible | hidden | inherit;  
// Изменяет внешний вид курсора, необходимо для исполнительных элементов на странице, полный перечень возможных значений смотри http://htmlbook.ru  
cursor: default | pointer  
// Выравнивает элементы между собой по горизонтали (!!! для двух и более элементов !!!)  
vertical-align: baseline | bottom | middle | top | inherit;  
// Позволяет очистить поток от обтекания  
clear: none | left | right | both | inherit;
```