

# Веб-разработка Лекция №3. Верстка.

Худяев Алексей Сергеевич

Тензор, 2016





Веб-страницы развиваются. Растут требования к оформлению. Это уже не просто текстовый документ, это менюшки, логотипы, баннеры, и прочее «оформление»



CSS еще не является основным средством оформления.

Для оформления используются

- 1. Многочисленные картинки
- 2. Атрибуты на тэгах
- 3. Для создания «сетки» используются таблицы
- 4. Для задания фиксированных размеров в сетке использовали «спейсеры»

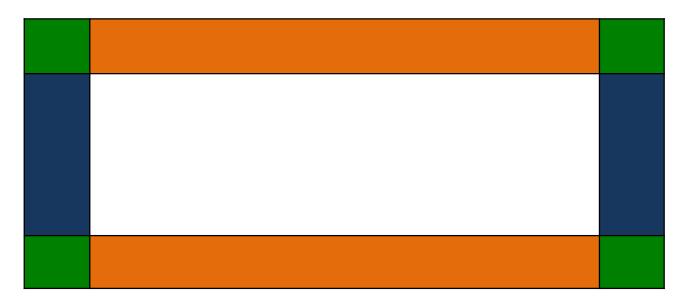
(http://en.wikipedia.org/wiki/Spacer GIF)



```
<img src="/i/ .gif" width="1" height="40" border="0" alt=""><br>
 <imq src="/i/ .qif" width="25" height="1" border="0" alt=""><br>
  <a href="/"><img src="/i/logo.gif" width="213" height="90" border="0" alt=""></a>
  <imq src="/i/slogan.gif" width="124" height="90" border="0" alt=""><br>
 <img src="/i/ .qif" width="1" height="1" border="0" alt=""><br/>br>
<span class="white small"><b>Персональный кабинет</b></span>
   <span class="white small">логин</span>
    <span class="white small">пароль</span>
    &nbsp:
```



Таблицы использовались даже для того, что бы сделать блок со скругленными углами!





#### Недостатки:

- 1. Все в одной куче
- 2. Очень много кода
- 3. Трудно вносить изменения
- 4. Страдает индексация поисковиками
- 5. Страдает совместимость с устройствами



#### Что же делать? CSS!

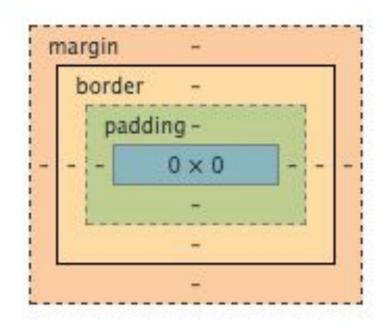
- 1. Позволяет отделить стиль от оформления
- 2. Позволяет уменьшить размер сайта (в байтах)
- 3. Улучшается кэширование
- 4. Лучше для поисковиков
- Разные представления для разных устройств. Выше совместимость.





- Любой элемент на странице представляется прямоугольным блоком или боксом.
- У любого бокса есть размеры
- Боксы можно вкладывать друг в друга

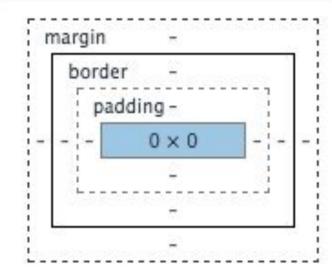






# Область содержимого (область контента)

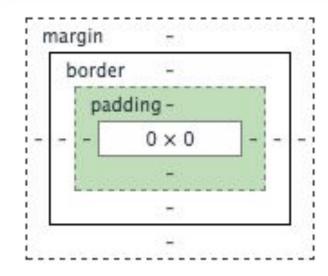
- Здесь располагается содержимое блока (в т.ч. и вложенные блоки)
- На область контента распространяется фон (если есть)





### Отступы (паддинги, padding)

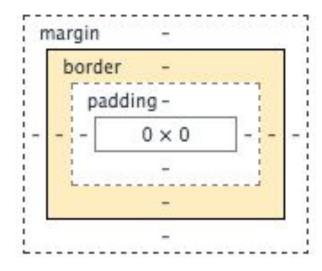
- Промежуток между рамкой и областью содержимого
- На отступ распространяется фон (если он есть)





### Граница, рамка, бордер (border)

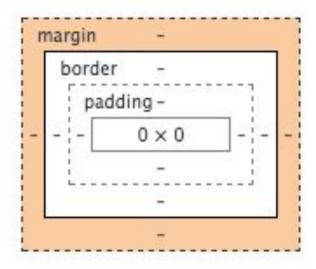
- Рамка блока
- Может иметь толщину но быть прозрачной





### Границы, маржины (margin)

• Отступ от данного блока до окружающих





### Размеры блока

- width
- height

Размерность HE включает в себя padding и border



```
.box {
 width: 200px;
 padding: 20px;
 border: 1px solid red;
 margin: 15px;
Итоговая ширина: 200 + 20 * 2 + 1 * 2 = 242px
```



```
.box {
 width: 100%;
 padding: 20px;
 border: 1px solid red;
 margin: 15px;
???
```



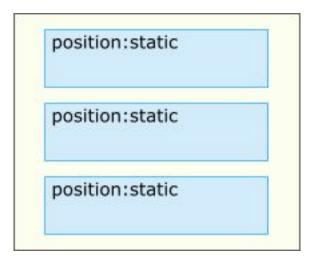


#### 4 типа позиционирования:

- 1. Static
- 2. Absolute
- 3. Relative
- 4. Fixed

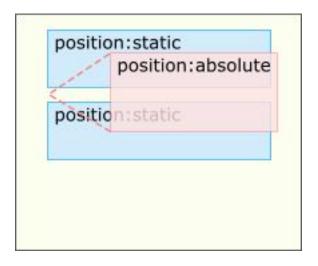


#### Static



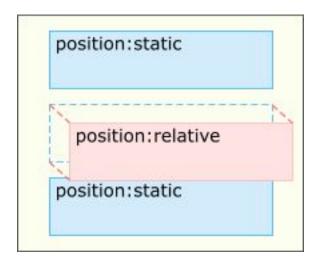


#### **Absolute**





#### Relative





**Fixed** 

Так же, как и absolute, но HE скроллируется вместе со страницей.

Если не влез, то «скрытую» часть никогда не увидеть. Браузер не покажет для нее скролл.





```
.absBox {
  left: 10px;
  top: 20px;
}
```



```
.absBox {
  right: 30px;
  bottom: 50px;
}
```



```
.absBox {
  left: 10px;
  right: 30px;
  top: 20px;
  bottom: 50px;
}
```



```
.absBox {
  left: 0;
  right: 0;
  top: 0;
  bottom: 0;
}
```



# «Точка отсчета»

### «Точка отсчета»



Координаты для left, top, right, bottom рассчитываются от **ближайшего предка** с позиционированием, отличным от static.

Если такового нет – рассчитываются от окна.

Для этого часто используют position: relative без задания каких-либо смещений.





#### Два основных типа боксов

- Блочные
- Строчные

### Управление типом: свойство display

- Блочные == block
- Строчные == inline



В чем разница?

#### Блочные:

- 1. Занимают всю ширину если не указано обратное
- Высота рассчитывается по вложенным боксам если не задано обратное



В чем разница?

### Строчные

- 1. Ширина и высота строго по контенту
- 2. Не применяются вертикальные маржины и паддинги



В потоке элементы стоят друг за другом.

Строчные без переносов.

Блочные – с переносом.



CтрокаCтрока c <strong>выделенным</strong> текстом

Строка

Строка с выделенным текстом



```
p { display: inline; }
```

Строка Строка с выделенным текстом



```
p { display: inline; }
strong { display: block; }
```

Строка Строка с

выделенным

текстом



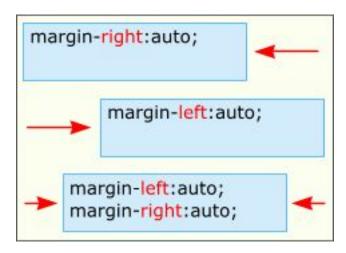
Ширина блочного бокса по умолчанию рассчитывается автоматически.

Блок будет занимать полную доступную ширину с учетом маржинов, бордеров и паддингов





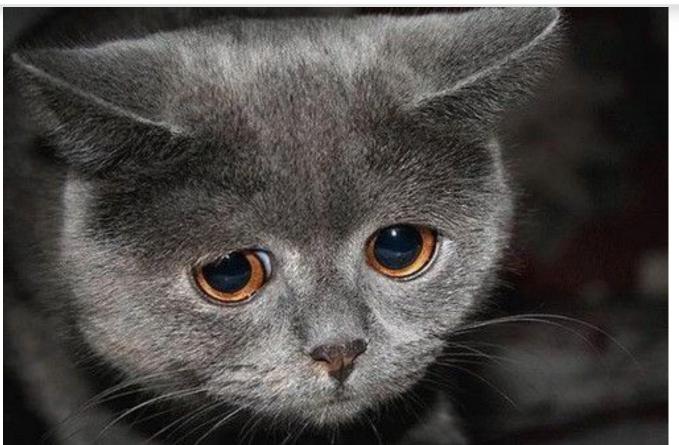
Если ширина указана явно можно управлять выравниванием блока





Вертикальное выравнивание...







#### Flexbox!

Вертикальное выравнивание!

CM. <a href="http://frontender.info/a-guide-to-flexbox/">http://frontender.info/a-guide-to-flexbox/</a>





```
<h1>Заголовок</h1>Абзац текста...</h1>h1 { margin: 20px 0; }p { margin: 10px 0; }
```



```
<h1>Заголовок</h1>Абзац текста...</h1>h1 { margin: 20px 0; }p { margin: 10px 0; }
```

Сколько между блоками?



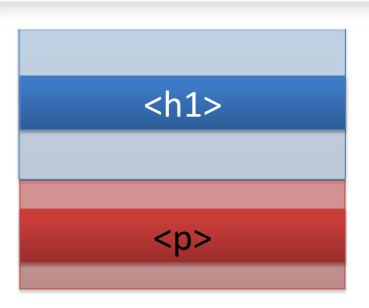
```
<h1>3аголовок</h1>Абзац текста...
```

h1 { margin: 20px 0; }

p { margin: 10px 0; }

Сколько между блоками?

Ответ: 30рх





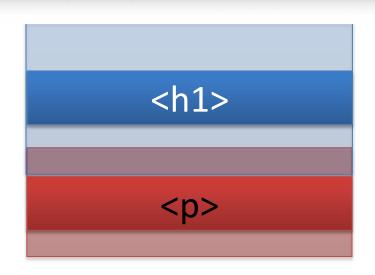
```
<h1>3аголовок</h1>Абзац текста...
```

h1 { margin: 20px 0; }

p { margin: 10px 0; }

Сколько между блоками?

Ответ: <del>30рх</del> 20рх





```
<div>
 <р>Абзац...</р>
 <р>Абзац...</р>
</div>
div { margin: 5px 0; }
p { margin: 10px 0; }
```

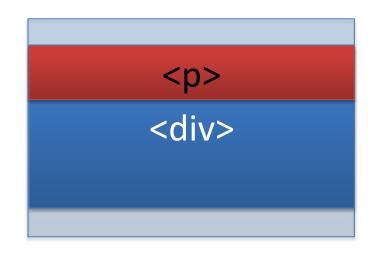


```
<div>
 <р>Абзац...</р>
 <р>Абзац...</р>
</div>
div { margin: 5px 0; }
p { margin: 10px 0; }
```



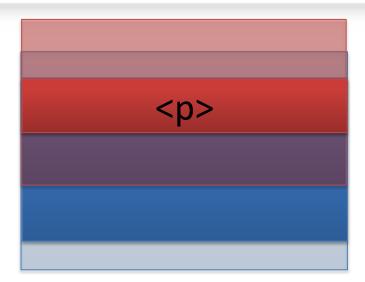


```
<div>
 <р>Абзац...</р>
 <р>Абзац...</р>
</div>
div { margin: 5px 0; }
p { margin: 10px 0; }
```



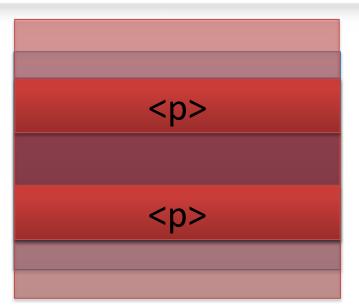


```
<div>
 <р>Абзац...</р>
 <р>Абзац...</р>
</div>
div { margin: 5px 0; }
p { margin: 10px 0; }
```





```
<div>
 <р>Абзац...</р>
 <р>Абзац...</р>
</div>
div { margin: 5px 0; }
p { margin: 10px 0; }
```



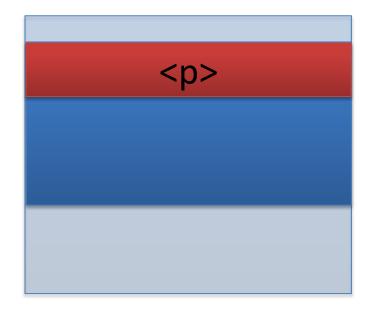


```
<div>
 <р>Абзац...</р>
 <р>Абзац...</р>
</div>
div { margin: 5px 0 15px 0; }
p { margin: 10px 0; }
```



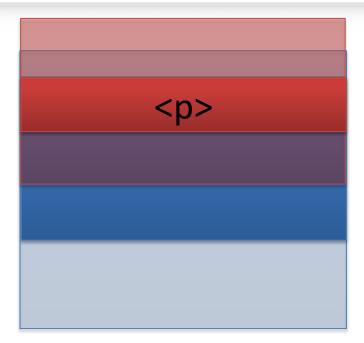


```
<div>
 <р>Абзац...</р>
 <р>Абзац...</р>
</div>
div { margin: 5px 0 15px 0; }
p { margin: 10px 0; }
```



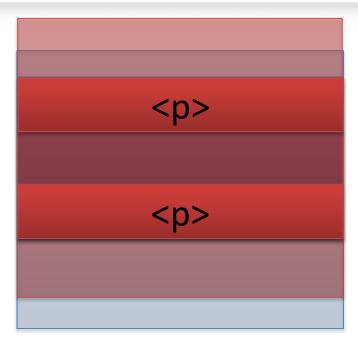


```
<div>
 <р>Абзац...</р>
 <р>Абзац...</р>
</div>
div { margin: 5px 0 15px 0; }
p { margin: 10px 0; }
```





```
<div>
 Абзац...
 <р>Абзац...</р>
</div>
div { margin: 5px 0 15px 0; }
p { margin: 10px 0; }
```







Зачем это нужно?

p { margin: 5px 0; }

Ожидание: 5рх

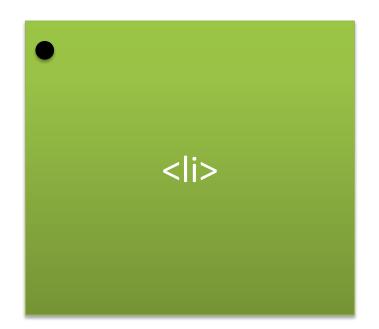
Реальность: 10рх

Схлопывание: у элемента всегда те границы, которые

ему нужны

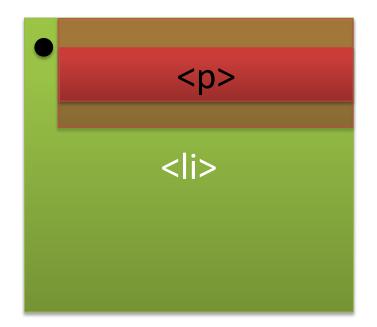


```
</ri>
```



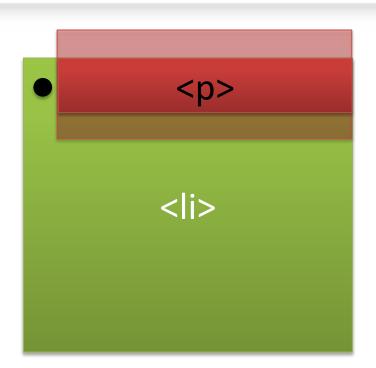


```
</ri>
```





```
</ri>
```





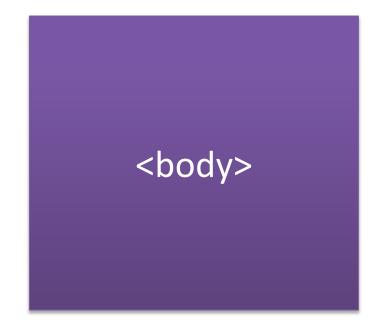
# Схлопывание границ. Проблемы...



```
<body>
 <div id="heading">
   <h1>Teнзop</h1>
   С 1996 года...
body, #heading { margin: 0 }
#heading h1 { margin: 10px; }
#heading p { margin: 5px; }
```

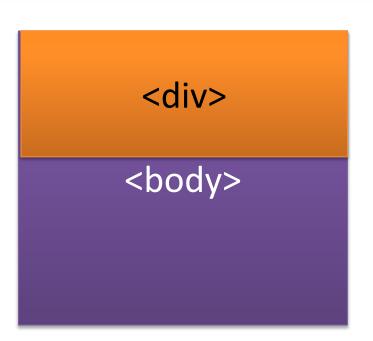


```
<body>
 <div id="heading">
   <h1>Teнзop</h1>
   < 1996 года...</p>
body, #heading { margin: 0 }
#heading h1 { margin: 10px; }
#heading p { margin: 5px; }
```



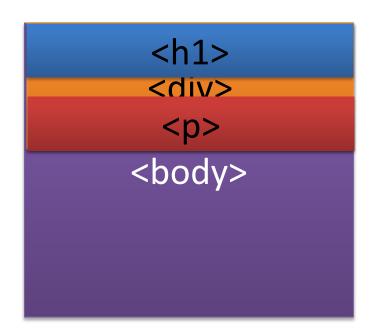


```
<body>
 <div id="heading">
   <h1>Teнзop</h1>
   < 1996 года...</p>
body, #heading { margin: 0 }
#heading h1 { margin: 10px; }
#heading p { margin: 5px; }
```



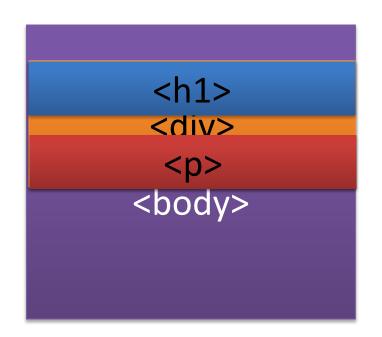


```
<body>
 <div id="heading">
   <h1>Teнзop</h1>
   < 1996 года...</p>
body, #heading { margin: 0 }
#heading h1 { margin: 10px; }
#heading p { margin: 5px; }
```





```
<body>
 <div id="heading">
   <h1>Teнзop</h1>
   < 1996 года...</p>
body, #heading { margin: 0 }
#heading h1 { margin: 10px; }
#heading p { margin: 5px; }
```



Границы вывалились из #heading!!!



#### Как починить?

1. Установить блоку, из которого вывалились границы бордер или паддинг

```
#heading {
  padding-top: 1px;
  /* или */
  border-top: 1px solid transparent;
}
```



#### Как починить?

1. Установить блоку, из которого вывалились границы бордер или паддинг

Плюсы: самый простой способ

Минусы: размер блока увеличится



Как починить?

2. Заменить маржин на паддинг с нужной стороны

```
h1 {
    margin: 10px;
    margin: 0 10px 10px 10px;
    padding-top: 10px;
}
```



Как починить?

2. Заменить маржин на паддинг с нужной стороны

Плюсы: визуально полностью идентично тому, что

хотели

Минусы: неочевидный набор правил



Как починить?

3. «Вырвать» из потока

```
#heading {
    position: absolute;
    /* или */
    float: left;
}
```



Как починить?

3. «Вырвать» из потока

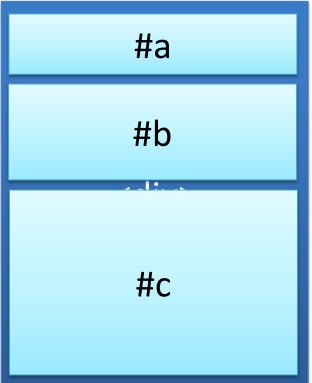
Плюсы: визуальная идентичность ожидаемому

Минусы: меняется раскладка, усложняются стили



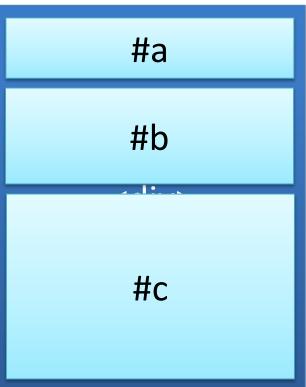


```
<div>
        <div id="a">...</div>
        <div id="b">...</div>
        <div id="c">...</div>
        <div id="c">...</div>
```



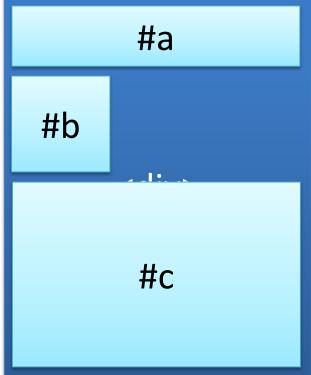


```
<div>
 <div id="a">...</div>
 <div id="b">...</div>
 <div id="c">...</div>
</div>
#b {
 float: left;
```



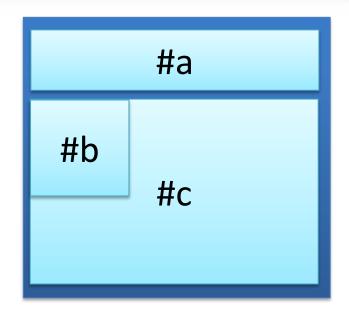


- 1. Смещается к указанной стороне
- Перестает занимать место, схлопывается



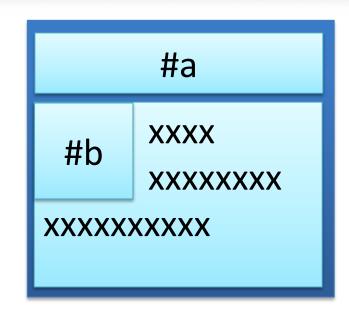


- 1. Смещается к указанной стороне
- Перестает занимать место, схлопывается
- 3. Следующие за ним **блочные** боксы подтягиваются на его место



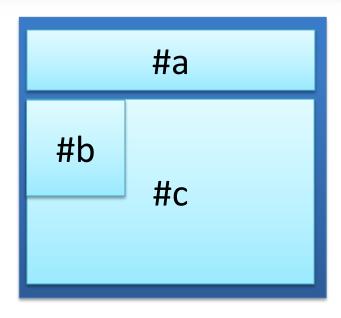


- 1. Смещается к указанной стороне
- Перестает занимать место, схлопывается
- 3. Следующие за ним **блочные** боксы подтягиваются на его место
- Строчные боксы внутри пододвинувшихся начинают обтекать его.





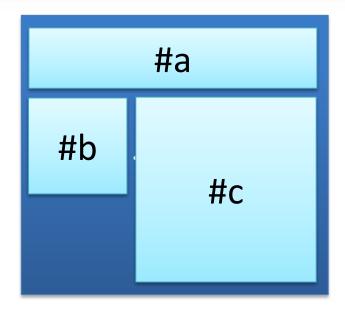
#b, #c { float: left }





#b, #c { float: left }

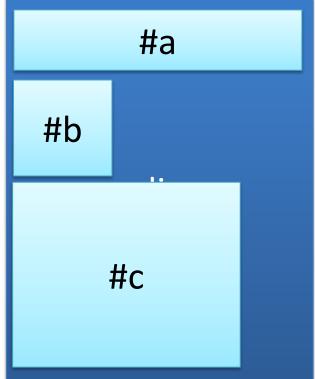
Если влезет, подтянется к предыдущему float-блоку





#b, #c { float: left }

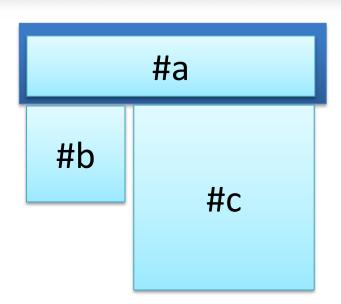
Если влезет, подтянется к предыдущему float-блоку Если не влезет, встанет за ним





Что случается с родительским блоком?

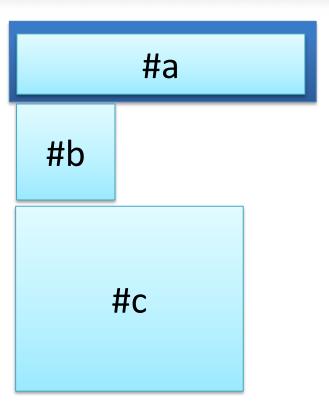
Он не учитывает float-боксы при расчете высоты





Что случается с родительским блоком?

Он не учитывает float-боксы при расчете высоты

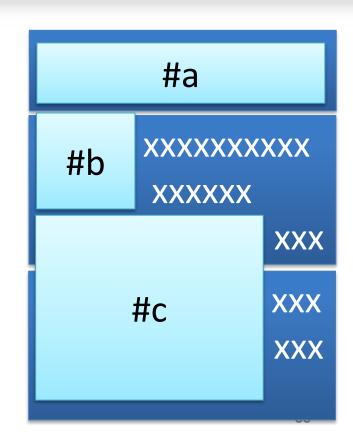




Что случается с родительским блоком?

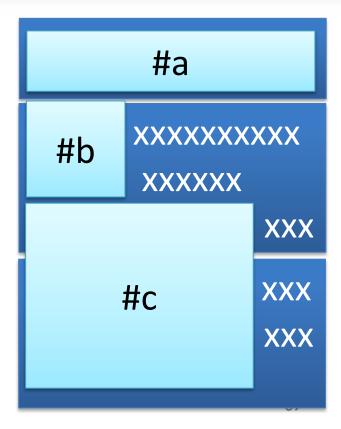
Он не учитывает float-боксы при расчете высоты

При этом идущие за родителем блоки тоже подтянутся!





Что делать, если мы хотим предотвратить «подтягивание» блока?



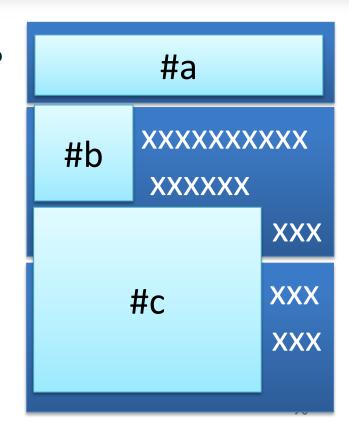


Что делать, если мы хотим предотвратить «подтягивание» блока?

Свойство clear

#### Значения:

- left
- right
- both



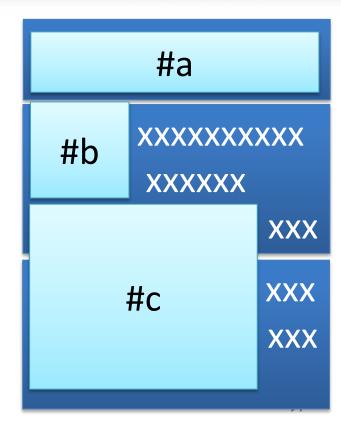


Что делать, если мы хотим предотвратить «подтягивание» блока?

Свойство clear

#### Значения:

- left нет флоатов слева
- right нет флоатов справа
- both нет по любой стороне

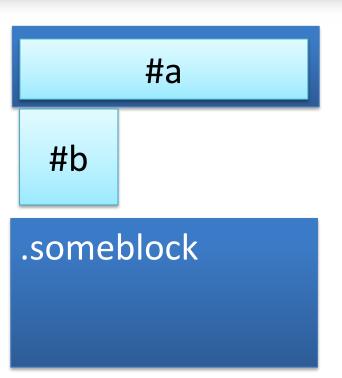




Что делать, если мы хотим предотвратить «подтягивание» блока?

Свойство clear

```
.someblock {
  clear: left;
}
```

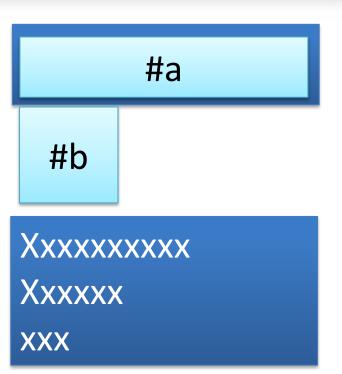




Что делать, если мы хотим предотвратить «подтягивание» блока?

Свойство clear

```
.someblock {
  clear: left;
}
```





## Полезные ссылки

### Учебник



- <a href="http://softwaremaniacs.org/blog/category/primer/">http://softwaremaniacs.org/blog/category/primer/</a> отличный учебник по верстке
- http://softwaremaniacs.org/blog/2005/09/05/css-layout-flow -margins/
- http://softwaremaniacs.org/blog/2005/12/01/css-layout-float/
   t/





# Вопросы есть?



#### Спасибо за внимание!