Веб-разработка
Лекция №3. Верстка.

Худяев Алексей Сергеевич

Тензор, 2016

Предпосылки

Предпосылки

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

Предпосылки

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»

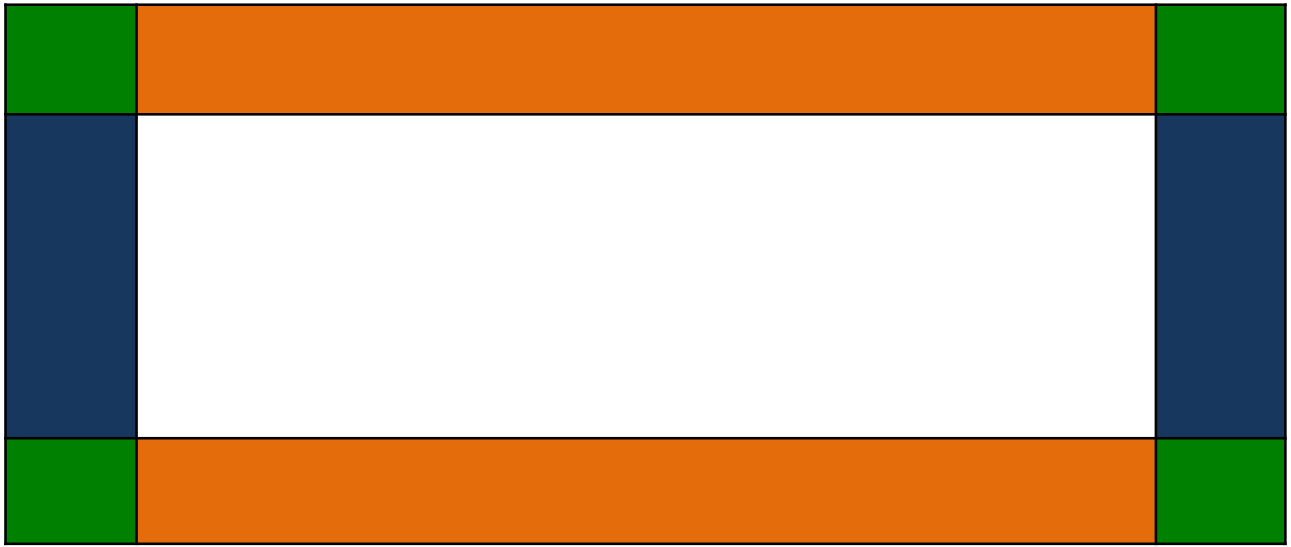
(http://en.wikipedia.org/wiki/Spacer_GIF)

Предпосылки


```
<table width="100%" background="/i/bg_top.jpg" style="background-repeat: no-repeat;" border="0" cellspacing="0" cellpadding="0">
<tr>
  <td width="75%" valign="bottom">
    <img src="/i/_.gif" width="1" height="40" border="0" alt=""><br>
    <table border="0" cellspacing="0" cellpadding="0"><tr valign="top">
      <td><img src="/i/_.gif" width="25" height="1" border="0" alt=""><br></td>
      <td><a href="/"><img src="/i/logo.gif" width="213" height="90" border="0" alt=""></a><br></td>
      <td><img src="/i/slogan.gif" width="124" height="90" border="0" alt=""><br></td>
    </tr></table>
  </td>

  <td bgcolor="#ffffff" background=" "><img src="/i/_.gif" width="1" height="1" border="0" alt=""><br></td>
  <td width="25%" bgcolor="#0C2E82" background=" ">
    <table width="100%" border="0" cellspacing="2" cellpadding="2"><tr valign="top">
      <td align="center">
        <table width="90%" cellpadding="0" cellspacing="1" border="0">
        <tr valign="top">
          <td colspan="3"><span class="white small"><b>Персональный кабинет</b></span></td>
        </tr>

        <tr valign="top">
          <td><span class="white small">логин</span></td>
          <td><span class="white small">пароль</span></td>
          <td> </td>
        </tr>
```

Предпосылки

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



Предпосылки

Недостатки:

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

Предпосылки

Что же делать? CSS!

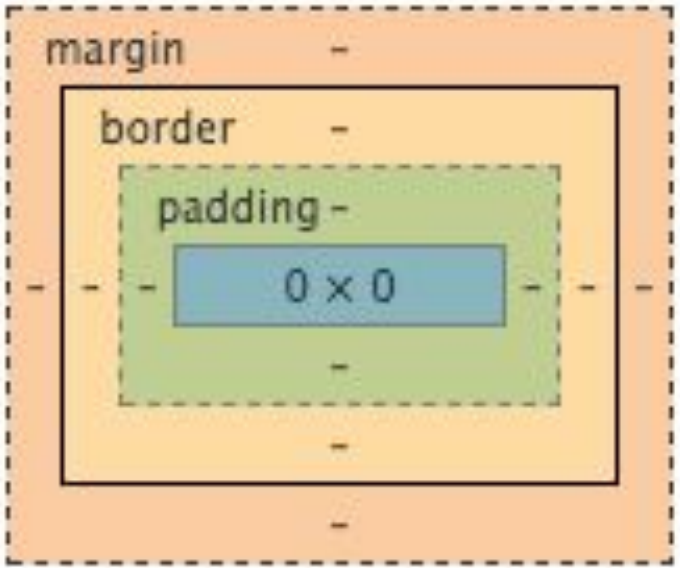
1. Позволяет отделить стиль от оформления
2. Позволяет уменьшить размер сайта (в байтах)
3. Улучшается кэширование
4. Лучше для поисковиков
5. Разные представления для разных устройств. Выше совместимость.

CSS Box Model

CSS Box Model

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

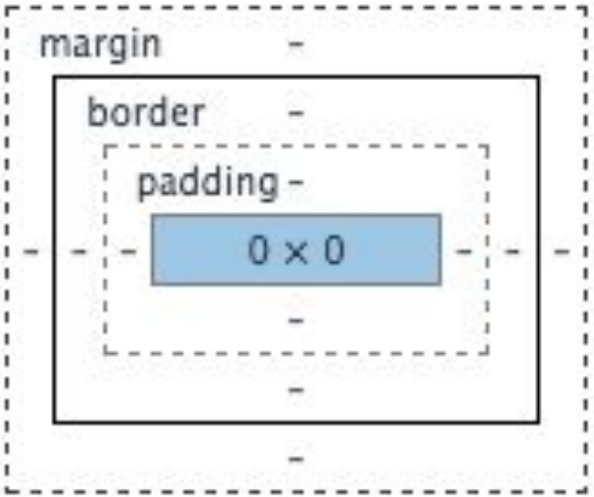
CSS Box Model



CSS Box Model

Область содержимого (область контента)

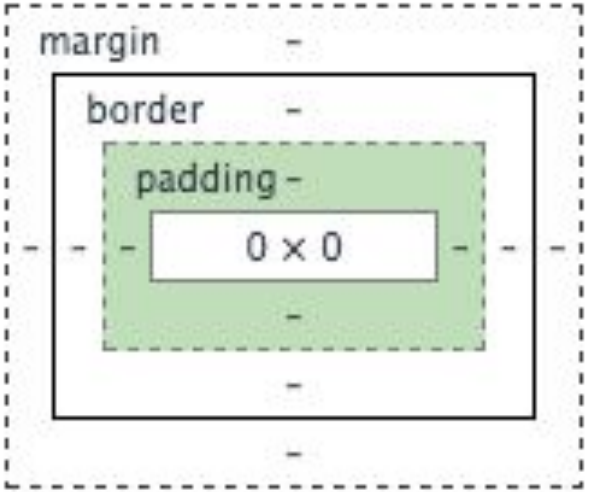
- Здесь располагается содержимое блока (в т.ч. и вложенные блоки)
- На область контента распространяется фон (если есть)



CSS Box Model

Отступы (паддинги, padding)

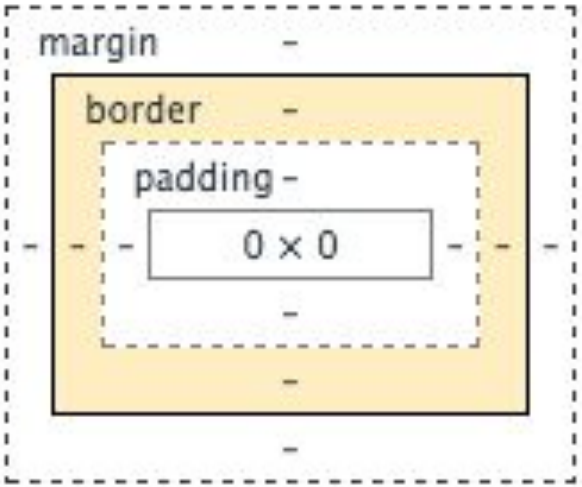
- Промежуток между рамкой и областью содержимого
- На отступ распространяется фон (если он есть)



CSS Box Model

Граница, рамка, бордер (border)

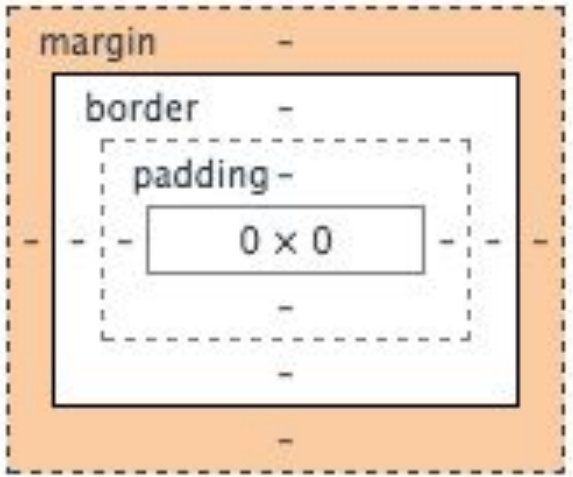
- Рамка блока
- Может иметь толщину но быть прозрачной



CSS Box Model

Границы, маржины (margin)

- Отступ от данного блока до окружающих



CSS Box Model

Размеры блока

- width
- height

Размерность НЕ включает в себя padding и border

CSS Box Model

```
.box {
  width: 200px;
  padding: 20px;
  border: 1px solid red;
  margin: 15px;
}
```

Итоговая ширина: 200 + 20 * 2 + 1 *2 = 242px

CSS Box Model

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

???
```

Позиционирование

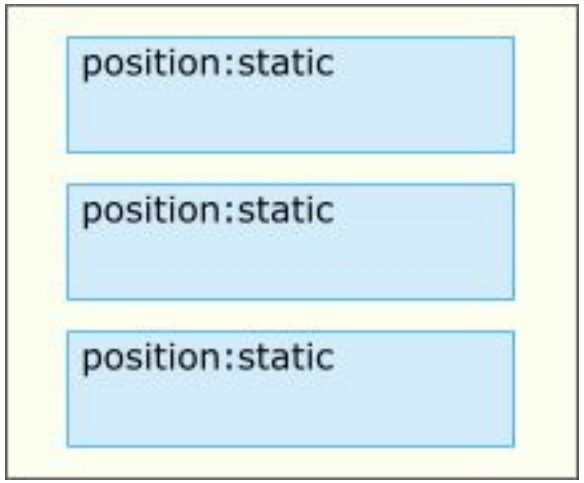
Позиционирование

4 типа позиционирования:

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

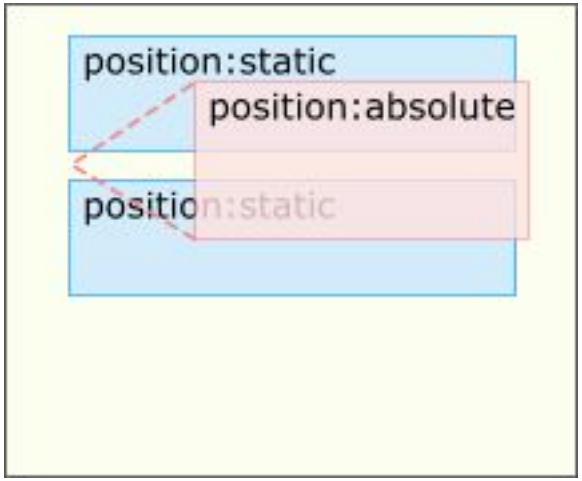
Позиционирование

Static



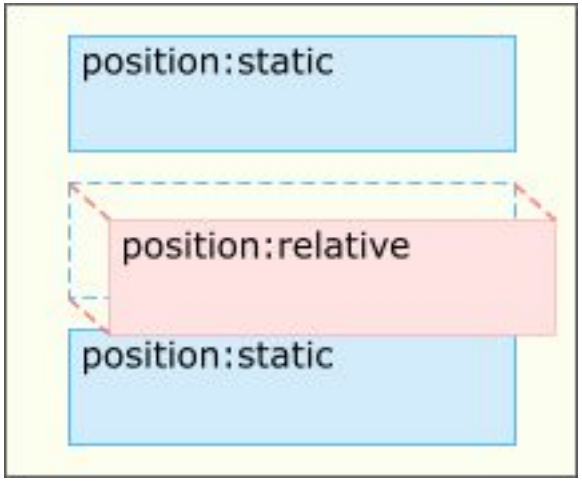
Позиционирование

Absolute



Позиционирование

Relative



Позиционирование

Fixed

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

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

Координаты

Координаты

При абсолютном и относительном позиционировании (absolute и relative) можно управлять «координатами» элемента.

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

Координаты

При абсолютном и относительном позиционировании (absolute и relative) можно управлять «координатами» элемента.

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

Координаты

При абсолютном и относительном позиционировании (absolute и relative) можно управлять «координатами» элемента.

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

Координаты

При абсолютном и относительном позиционировании (absolute и relative) можно управлять «координатами» элемента.

```
.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.

Поток

Поток

Два основных типа боксов

- Блочные
- Строчные

Управление типом: свойство display

- Блочные == block
- Строчные == inline

Поток

В чем разница?

Блочные:

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

Поток

В чем разница?

Строчные

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

Поток

В потоке элементы стоят друг за другом.

Строчные без переносов.

Блочные – с переносом.

Поток

Строка
Строка с выделенным текстом

Строка
Строка с **выделенным** текстом

Поток

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

Строка Строка с **выделенным** текстом

Поток

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

Строка Строка с
выделенным
текстом

Поток

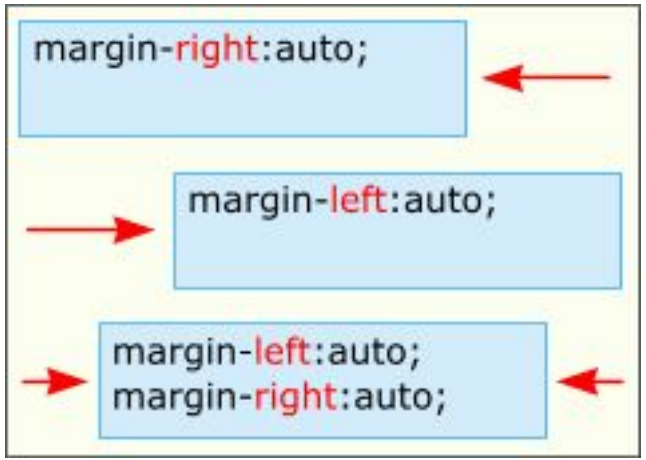
Ширина блочного бокса по умолчанию рассчитывается автоматически.

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

Выравнивание блоков

Выравнивание блоков

Если ширина указана явно можно управлять выравниванием блока



Выравнивание блоков

Вертикальное выравнивание...

Выравнивание блоков

Выравнивание блоков

Flexbox!

Вертикальное выравнивание!

см. http://frontender.info/a-guide-to-flexbox/

Схлопывание границ

Схлопывание границ

```
<h1>Заголовок</h1>
<p>Абзац текста...</p>

h1 { margin: 20px 0; }
p { margin: 10px 0; }
```

Схлопывание границ


```
<h1>Заголовок</h1>
<p>Абзац текста...</p>

h1 { margin: 20px 0; }
p { margin: 10px 0; }
```

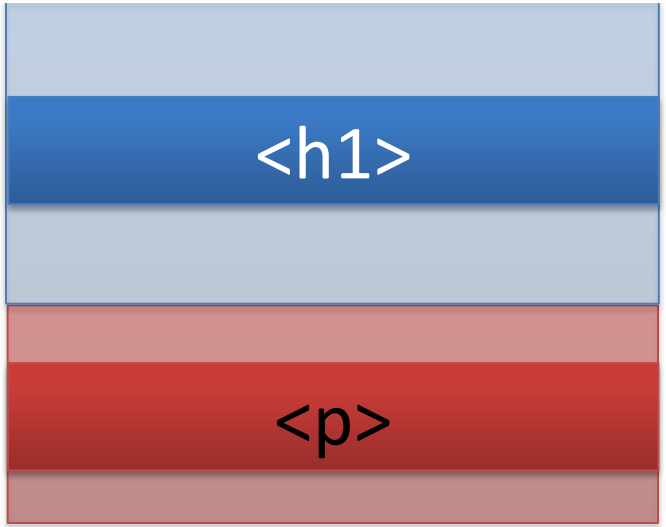
Сколько между блоками?

Схлопывание границ

<h1>Заголовок</h1>
Абзац текста...

h1 { margin: 20px 0; }
p { margin: 10px 0; }

Сколько между блоками?
Ответ: 30px

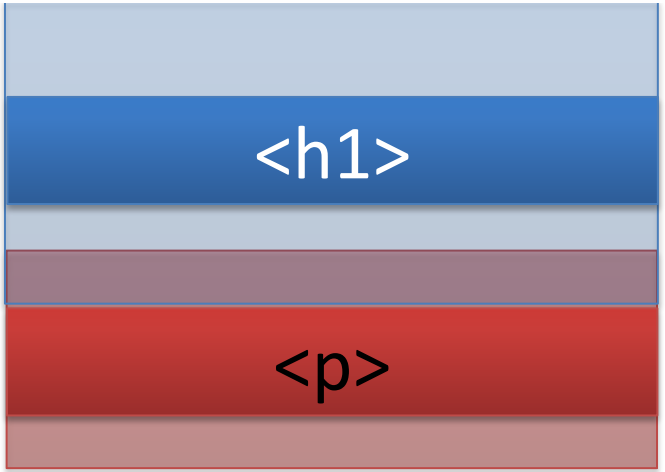


Схлопывание границ

```
<h1>Заголовок</h1>
<p>Абзац текста...</p>

h1 { margin: 20px 0; }
p { margin: 10px 0; }
```

Сколько между блоками?

Ответ: ~~30px~~ 20px



Схлопывание границ

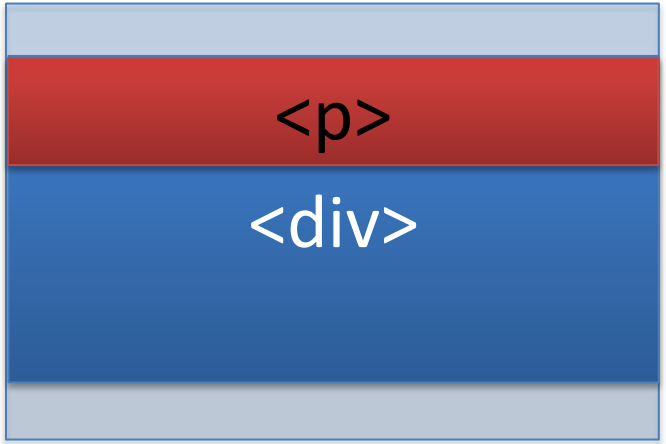
```
<div>
  <p>Абзац...</p>
  <p>Абзац...</p>
</div>

div { margin: 5px 0; }
p { margin: 10px 0; }
```

Схлопывание границ

```
<div>
  <p>Абзац...</p>
  <p>Абзац...</p>
</div>

div { margin: 5px 0; }
p { margin: 10px 0; }
```

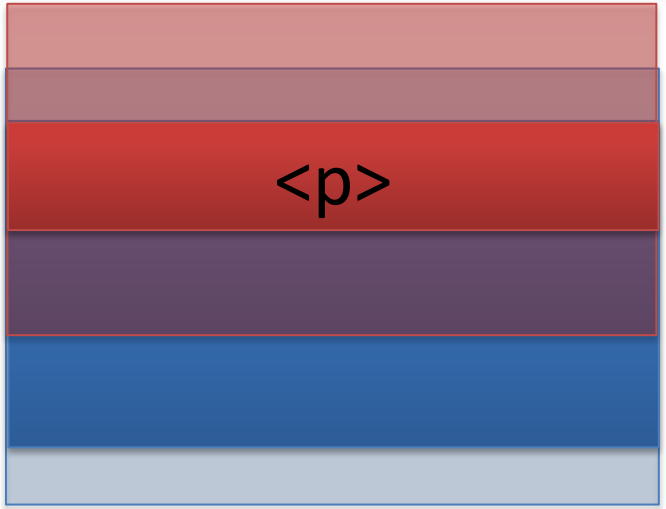
Схлопывание границ

```
<div>
  <p>Абзац...</p>
  <p>Абзац...</p>
</div>

div { margin: 5px 0; }
p { margin: 10px 0; }
```



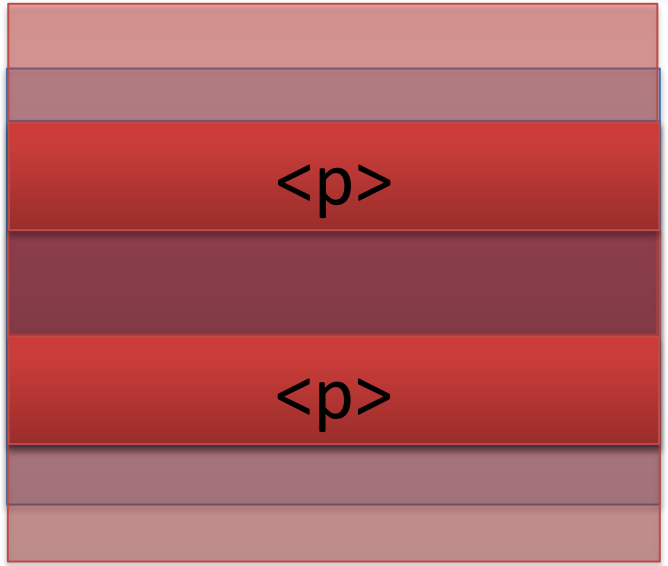
Схлопывание границ

```
<div>
  <p>Абзац...</p>
  <p>Абзац...</p>
</div>

div { margin: 5px 0; }
p { margin: 10px 0; }
```



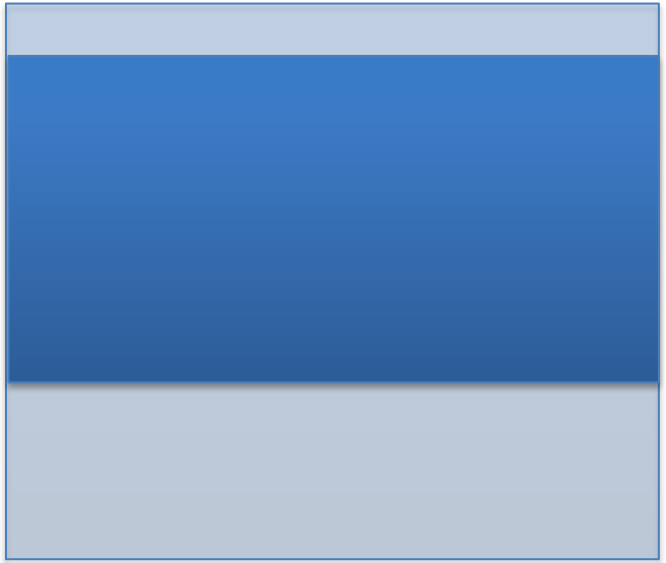
Схлопывание границ

```
<div>
  <p>Абзац...</p>
  <p>Абзац...</p>
</div>

div { margin: 5px 0; }
p { margin: 10px 0; }
```



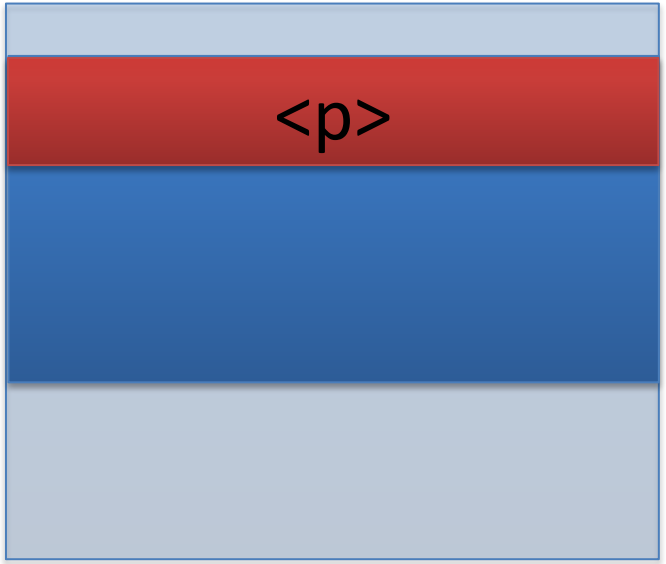
Схлопывание границ

```
<div>
  <p>Абзац...</p>
  <p>Абзац...</p>
</div>

div { margin: 5px 0 15px 0; }
p { margin: 10px 0; }
```



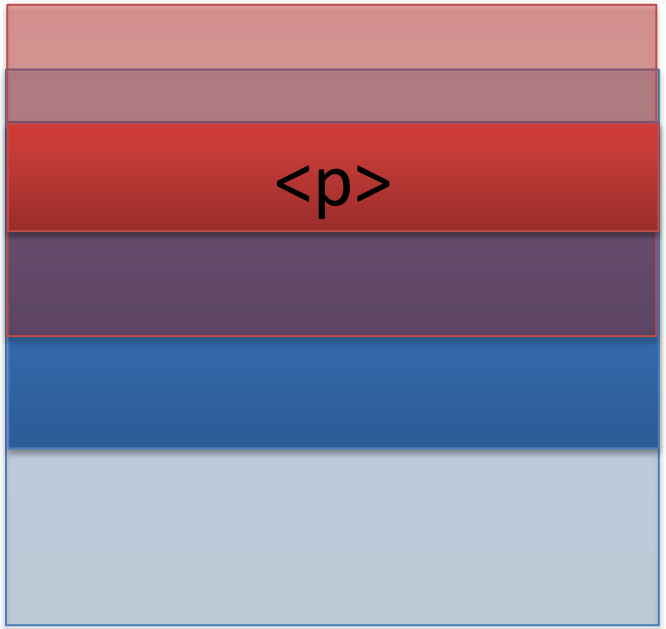
Схлопывание границ

```
<div>
  <p>Абзац...</p>
  <p>Абзац...</p>
</div>

div { margin: 5px 0 15px 0; }
p { margin: 10px 0; }
```



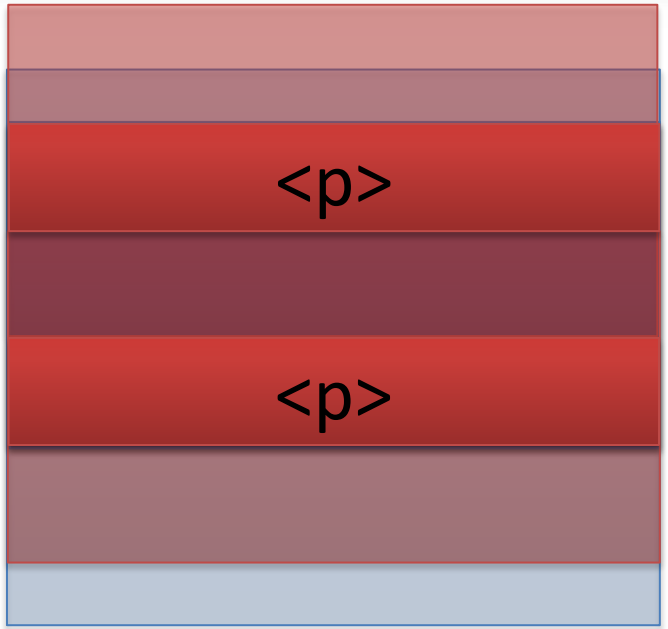
Схлопывание границ

```
<div>
  <p>Абзац...</p>
  <p>Абзац...</p>
</div>

div { margin: 5px 0 15px 0; }
p { margin: 10px 0; }
```



Схлопывание границ

```
<div>
  <p>Абзац...</p>
  <p>Абзац...</p>
</div>

div { margin: 5px 0 15px 0; }
p { margin: 10px 0; }
```



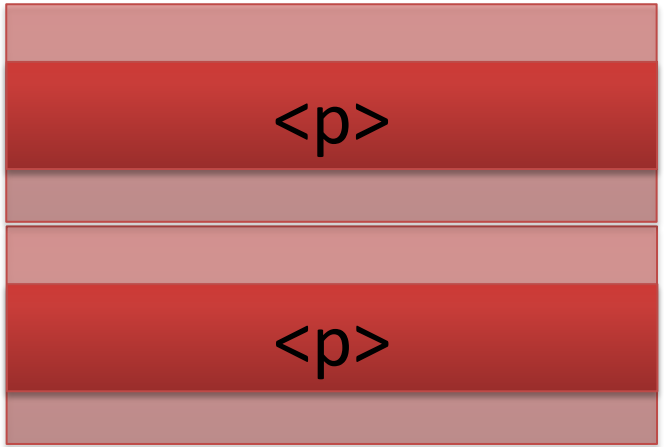
Схлопывание границ

Зачем это нужно?

Схлопывание границ

Зачем это нужно?

p { margin: 5px 0; }



Ожидание: 5px

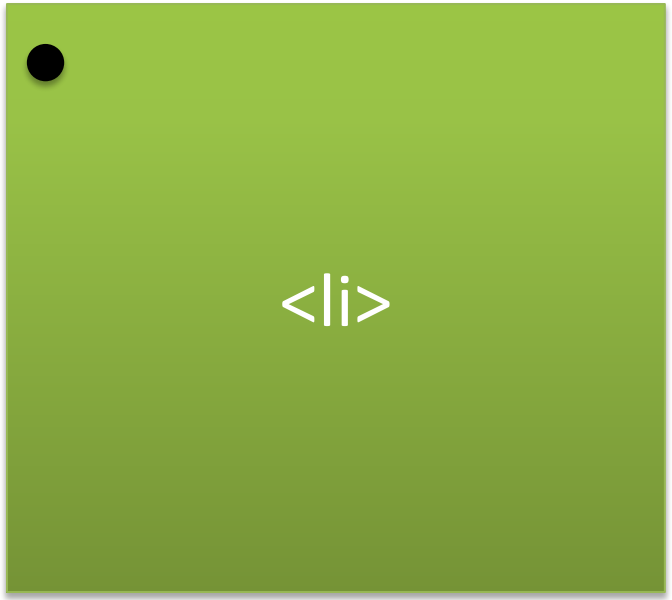
Реальность: 10px

Схлопывание: у элемента всегда те границы, которые ему нужны

Схлопывание границ

Зачем это нужно?

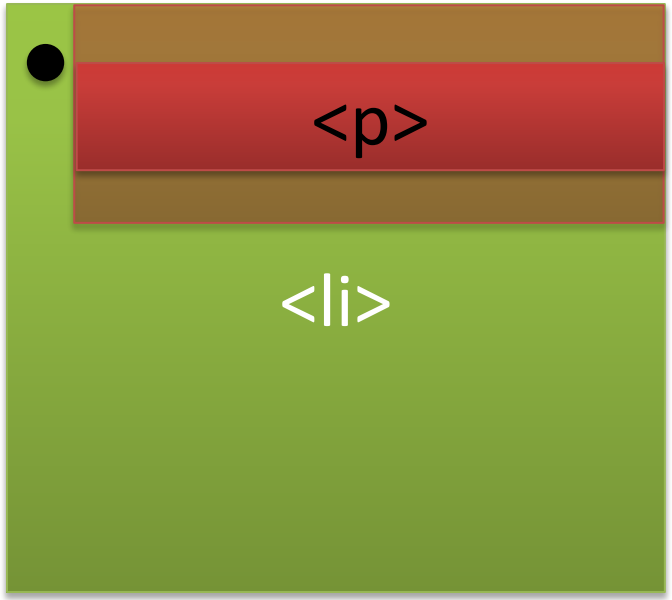
```
<ul>
  <li>
    <p>...</p>
  </li>
</ul>
```



Схлопывание границ

Зачем это нужно?

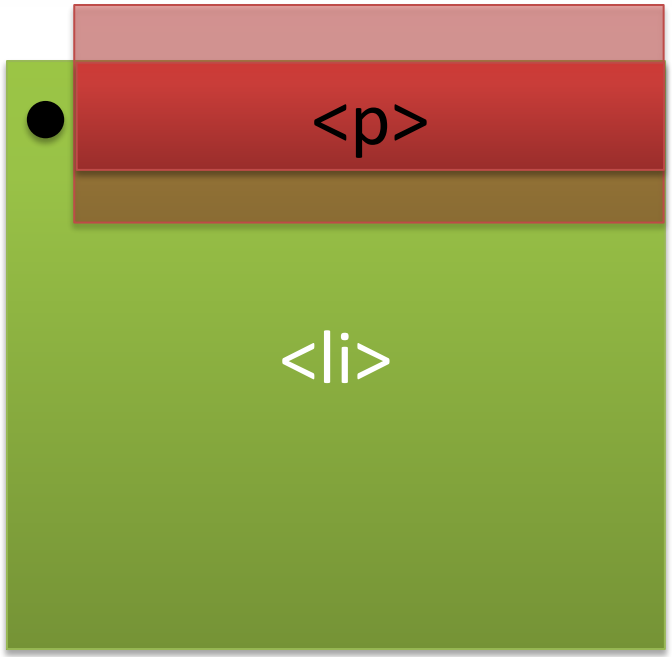
```
<ul>
  <li>
    <p>...</p>
  </li>
</ul>
```



Схлопывание границ

Зачем это нужно?

```
<ul>
  <li>
    <p>...</p>
  </li>
</ul>
```



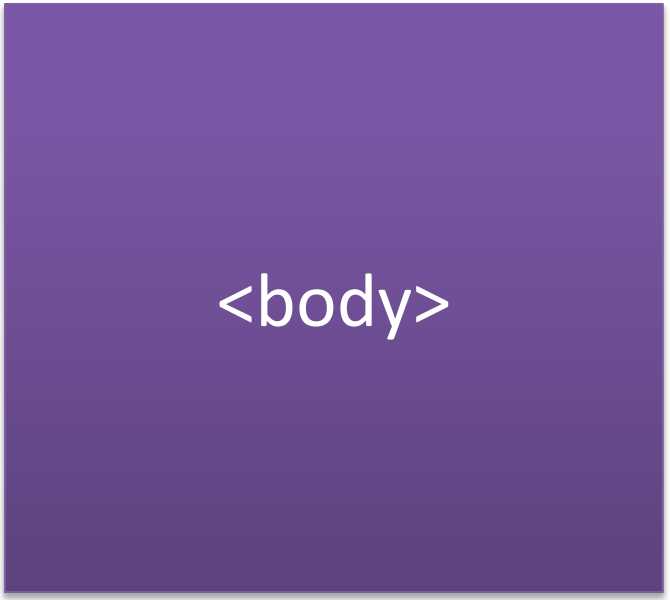
Схлопывание границ. Проблемы...

Проблемы

```
<body>
  <div id=“heading”>
    <h1>Тензор</h1>
    <p>С 1996 года...</p>

body, #heading { margin: 0 }
#heading h1 { margin: 10px; }
#heading p { margin: 5px; }
```

Проблемы

```
<body>
  <div id=“heading”>
    <h1>Тензор</h1>
    <p>С 1996 года...</p>

body, #heading { margin: 0 }
#heading h1 { margin: 10px; }
#heading p { margin: 5px; }
```

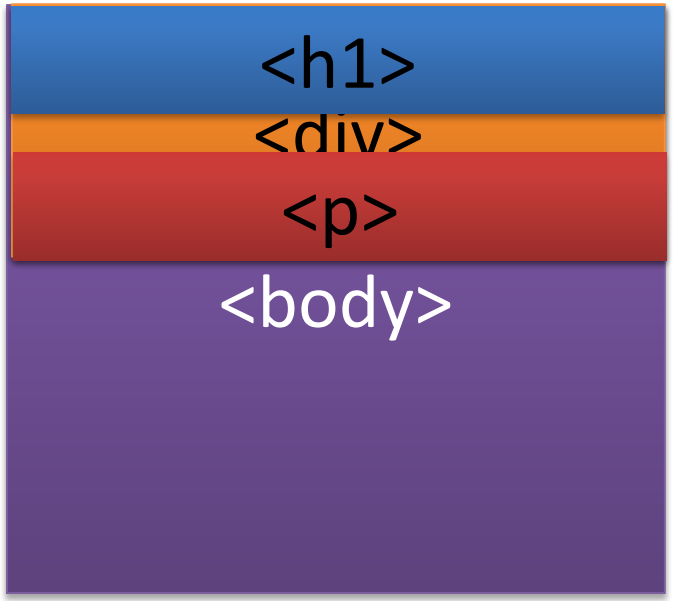


Проблемы

```
<body>
  <div id=“heading”>
    <h1>Тензор</h1>
    <p>С 1996 года...</p>

body, #heading { margin: 0 }
#heading h1 { margin: 10px; }
#heading p { margin: 5px; }
```

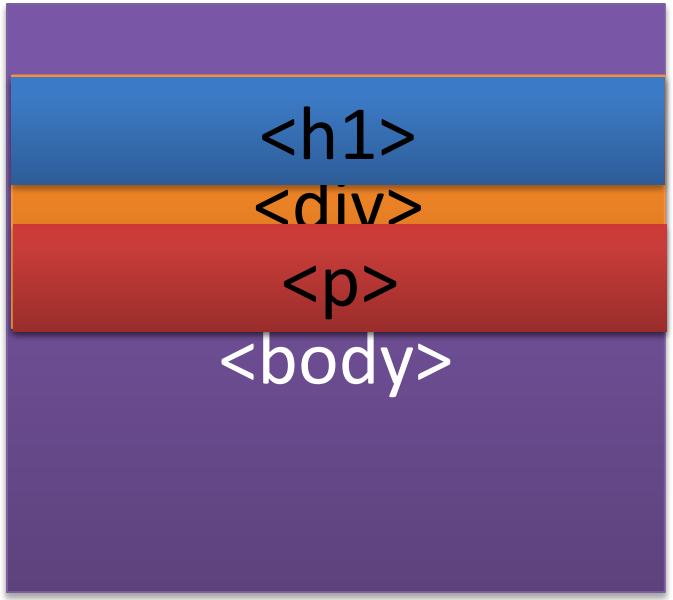
Проблемы

```
<body>
  <div id="heading">
    <h1>Тензор</h1>
    <p>С 1996 года...</p>

body, #heading { margin: 0 }
#heading h1 { margin: 10px; }
#heading p { margin: 5px; }
```



Проблемы

```
<body>
  <div id="heading">
    <h1>Тензор</h1>
    <p>С 1996 года...</p>

body, #heading { margin: 0 }
#heading h1 { margin: 10px; }
#heading p { margin: 5px; }
```



Границы вывалились из #heading!!!

Проблемы

Как починить?

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

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

Проблемы

Как починить?

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

Плюсы: самый простой способ

Минусы: размер блока увеличится

Проблемы

Как починить?

2. Заменить маржин на паддинг с нужной стороны

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

Проблемы

Как починить?

2. Заменить маржин на паддинг с нужной стороны

Плюсы: визуально полностью идентично тому, что хотели

Минусы: неочевидный набор правил

Проблемы

Как починить?

3. «Вырвать» из потока

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

Проблемы

Как починить?

3. «Вырвать» из потока

Плюсы: визуальная идентичность ожидаемому

Минусы: меняется раскладка, усложняются стили

Float

Float

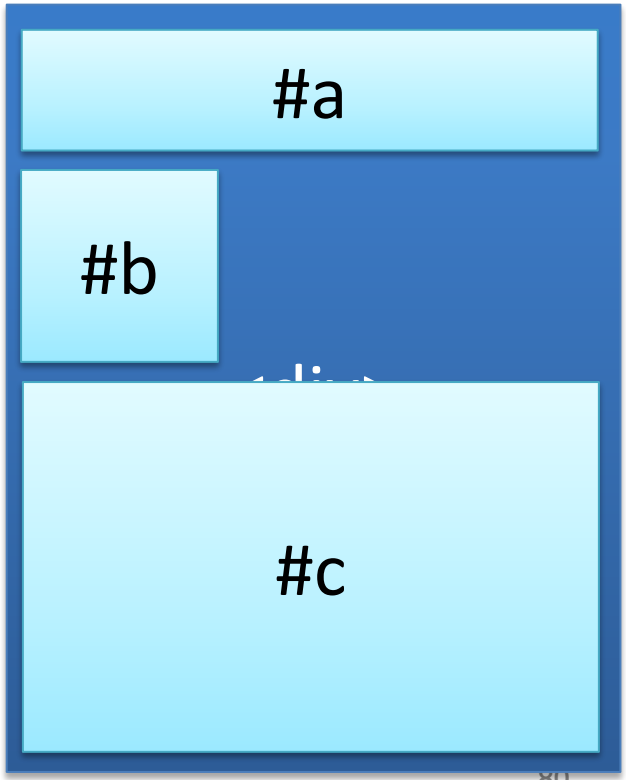
```
<div>
   <div id="a">...</div>
   <div id="b">...</div>
   <div id="c">...</div>
</div>
```

Float

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

#b {
  float: left;
}
```

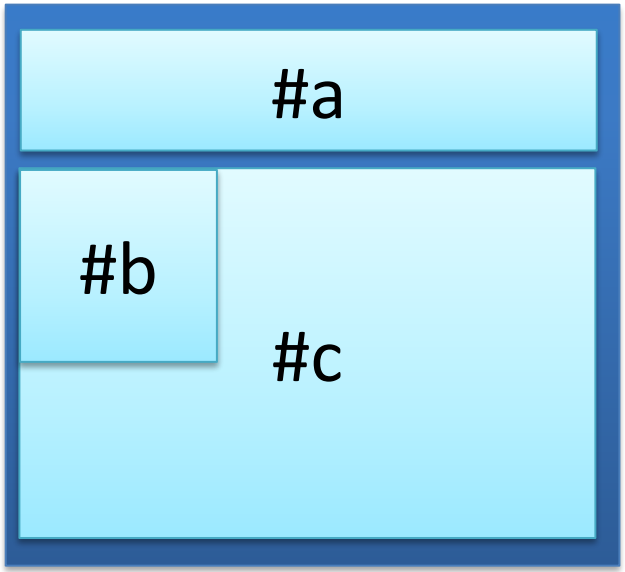
Float

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



Float

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

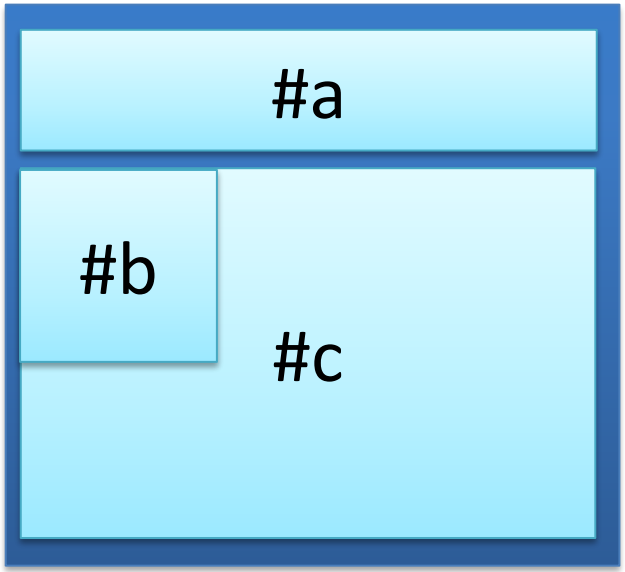


Float

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

Float

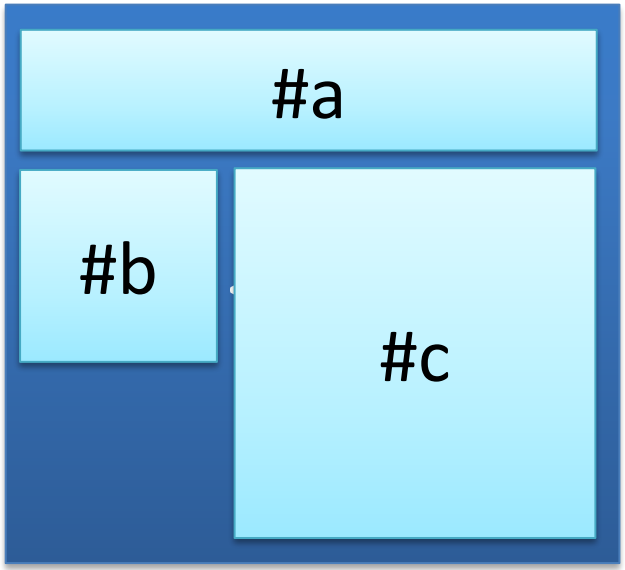
```
#b, #c { float: left }
```



Float

#b, #c { float: left }

Если влезет, подтянется к предыдущему float-блоку

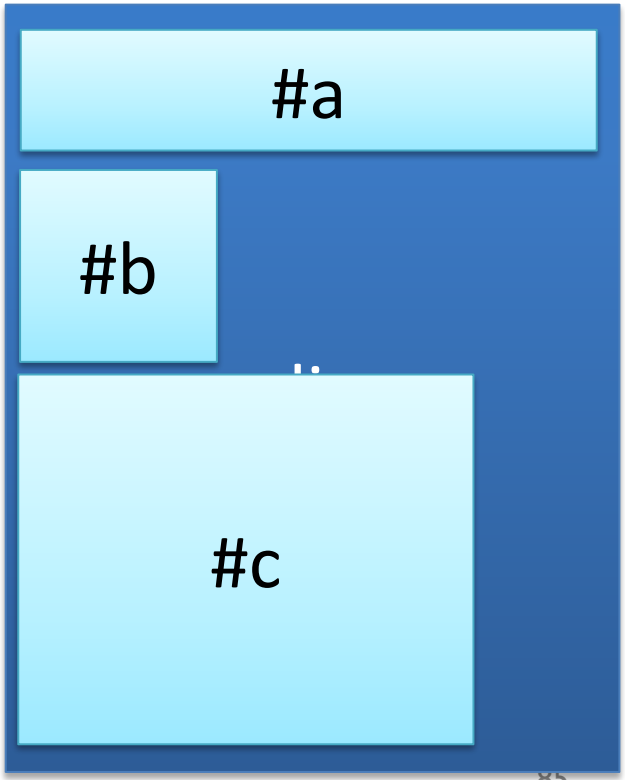


Float

#b, #c { float: left }

Если влезет, подтянется к предыдущему float-блоку

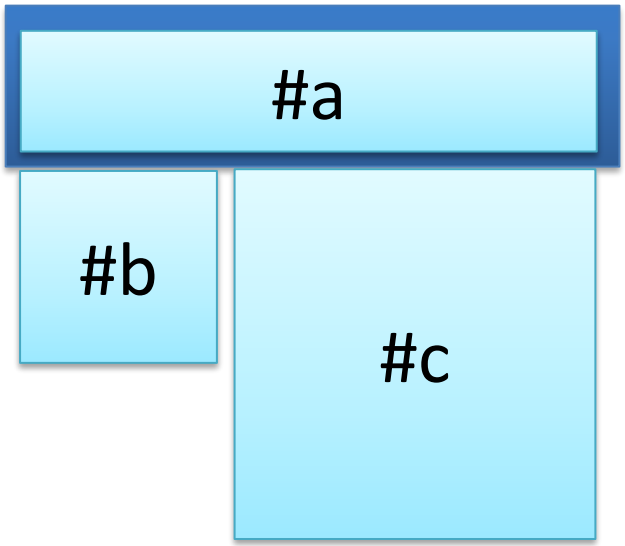
Если не влезет, встанет за ним



Float

Что случается с родительским блоком?

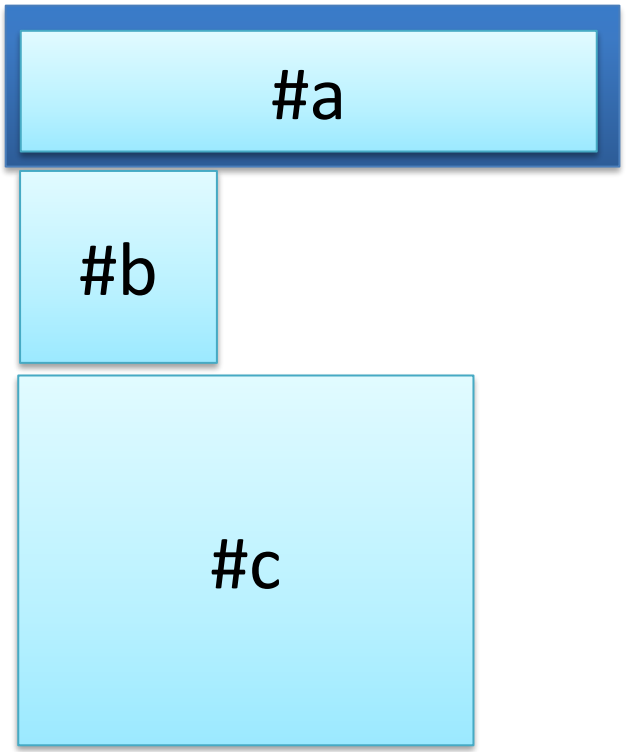
Он не учитывает float-боксы при расчете высоты



Float

Что случается с родительским блоком?

Он не учитывает float-боксы при расчете высоты



Float

Что случается с родительским блоком?

Он не учитывает float-боксы при расчете высоты

При этом идущие за родителем блоки тоже подтянутся!

Float

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

Float

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

Свойство clear

Значения:

- left
- right
- both

Float

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

Свойство clear

Значения:

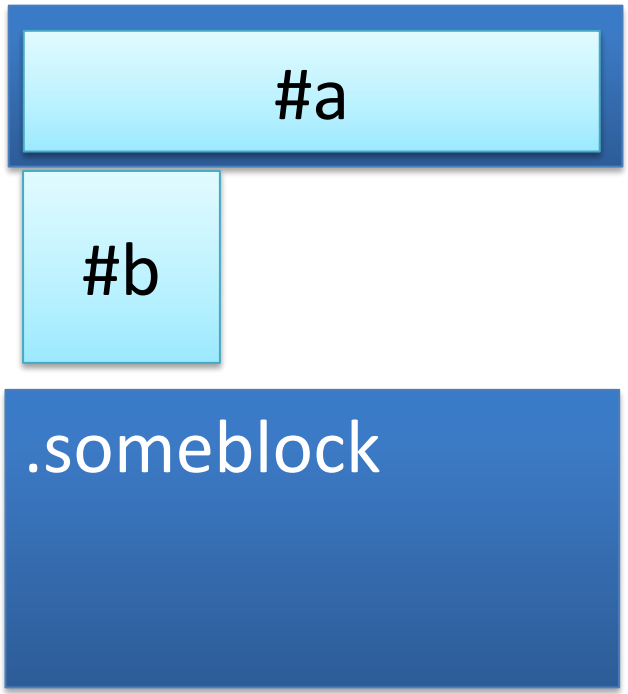
- left – нет флоатов слева
- right – нет флоатов справа
- both – нет по любой стороне

Float

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

Свойство clear

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



Float

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

Свойство clear

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

Полезные ссылки

Учебник

- http://softwaremaniacs.org/blog/category/primer/ - **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/

Вопросы есть?

Спасибо за внимание!