



BEETROOT
ACADEMY

Блочная модель CSS

Стили браузера по

умолчанию

У браузера есть стили по умолчанию.

<https://www.w3.org/TR/css-ui-3/#default-style-sheet>

Видно, что все элементы имеют свойство **display** благодаря которому браузер знает какой элемент является строчным, а какой блочным

Естественно встроенные стили можно переопределить своими стилями и сделать например строчный элемент блочный или наоборот.

Свойство display

display : значение

Значение	Что означает
block	Элемент показывается как блочный.
inline	Элемент отображается как строчный.
none	Удаляет элемент из документа. При этом занимаемое им место займет нижележащий элемент.
inline-block	Блочный элемент, перед и после которого нет переноса на новую строку. То есть внешне элемент ведет себя как строчный, но внутри элемента может быть блочные элементы.

Свойство visibility

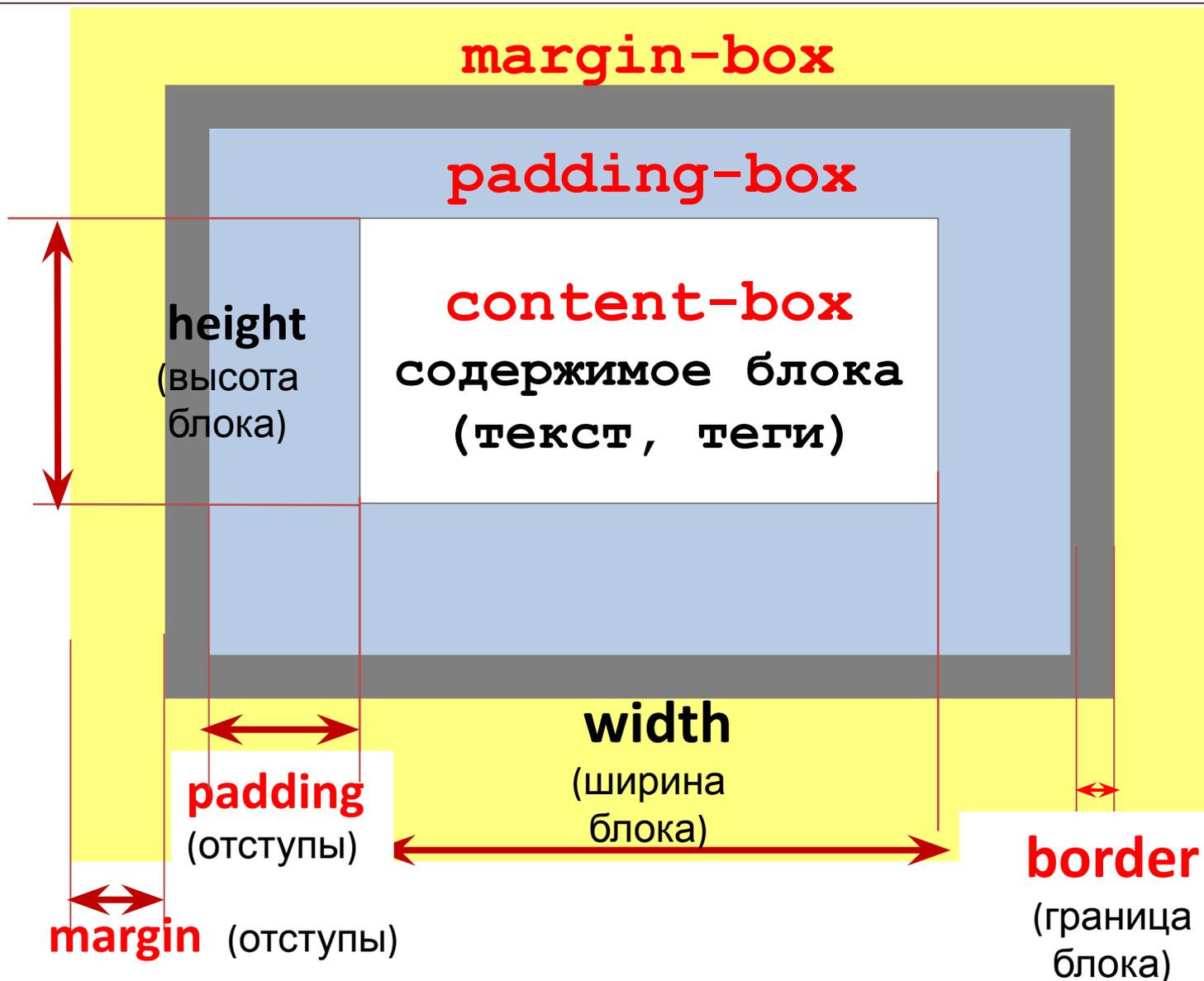
visibility : значение

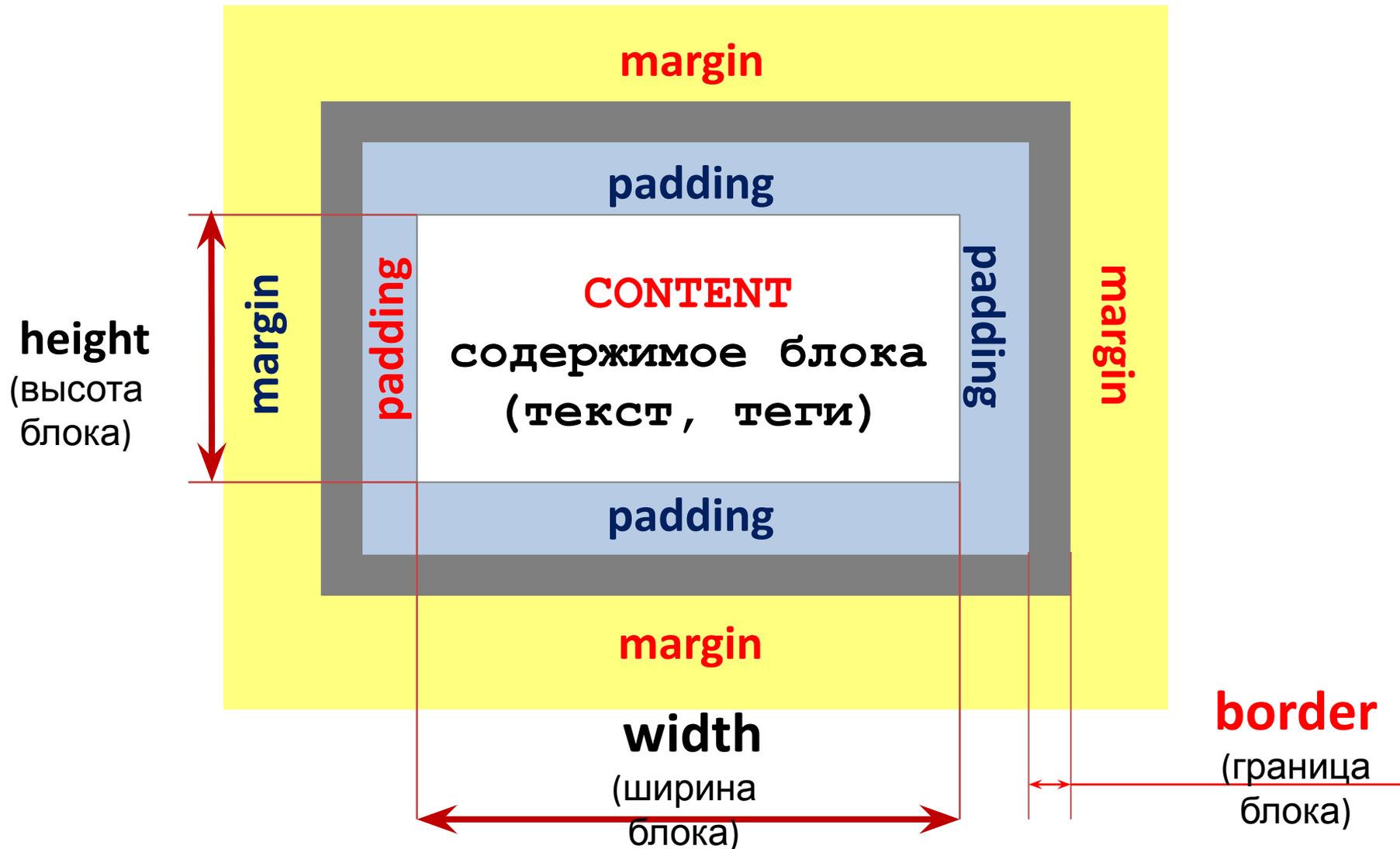
visible	Отображает элемент как видимый.
hidden	Элемент становится невидимым (полностью прозрачным), но он остается в потоке.
collapse	Для не табличных элементов эквивалентно hidden Для таблицы - заданные строки и колонки убираются

display:	Примеры элементов
<code>inline</code>	<code>span, a, em, i, u, b</code> и т.д
<code>inline-block</code>	- <code>img, object, embed, iframe</code> ; - <code>input, textarea, select, button</code> (элементы форм) - <code>audio, video, canvas</code>
<code>block</code>	<code>div, p, h1...h6, ...</code>
<code>list-item</code>	<code>li</code>
<code>table</code>	<code>table</code>

Блочная модель

Любой HTML-элемент можно представить в виде прямоугольника с 4-мя областями `content`, `margin`, `border`, `padding`





По умолчанию ширина блока на страниц рассчитывается как

`width + border-left + padding-left + padding-right + border-right`

Например

```
.block {  
  width: 300px;  
  padding: 20px;  
  border: 10px solid red;  
}
```

Ширина блока

`300px + 20px + 20px + 10px + 10px = 360px;`

В CSS 3 добавлено свойство

box-sizing: `content-box` | `border-box`

которое меняет алгоритм расчета ширины блока

Например

```
* {  
  box-sizing: border-box;  
}  
.block {  
  width: 300px;  
  padding: 20px;  
  border: 10px solid red;  
}
```

Ширина блока будет 300px

то есть свойства `width` и `height` блока включают в себя `padding` и `border-width`

Наименование сторон блока

top

left

right

bottom



Свойства min-width, min-height блока

```
.myblock {  
  max-width : 600px;  
  min-width : 400px;  
}
```

```
.myblock {  
  max-height : 800px;  
  min-height : 400px;  
}
```

max-width

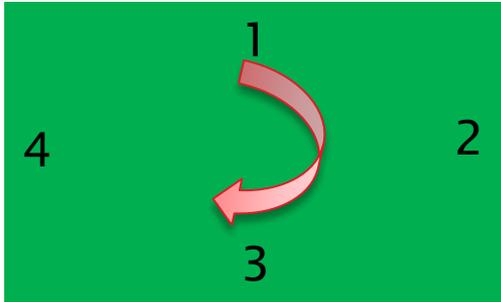
ширина блока не может быть **более** заданного значения

min-width

ширина блока не может быть **менее** заданного значения

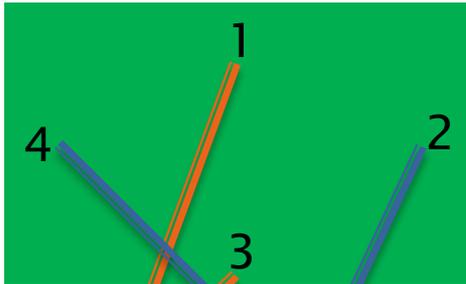
А между min-width и max-width ширина блока подстраивается под ширину своего родителя

Сокращенная запись свойств margin, padding блока

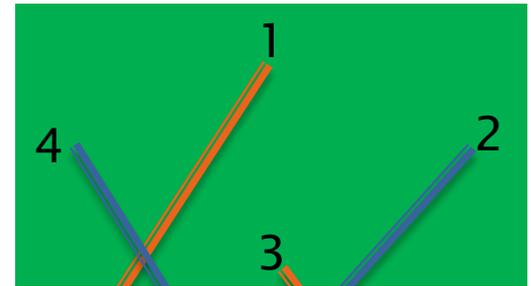


```
#parent{  
    margin : 20px;  
}
```

```
#parent{  
    padding : 10px 20px 10px 0;  
}
```



```
#parent {  
    margin : 10px 20px ;  
}
```



```
#parent {  
    margin : 10px 30px 20px ;  
}
```

Свойства блока padding, margin

```
.myblock {  
  padding-top : 10px;  
  padding-right : 20px;  
  padding-bottom : 10px;  
  padding-left : 20px;  
}
```

=

```
.myblock {  
  padding : 10px 20px 10px 10px;  
}
```

```
.myblock {  
  padding : 10px 20px;  
}
```

```
.myblock {  
  margin-top : 10px;  
  margin-right : 20px;  
  margin-bottom : 10px;  
  margin-left : 20px;  
}
```

=

```
.myblock {  
  margin : 10px 20px 10px 10px;  
}
```

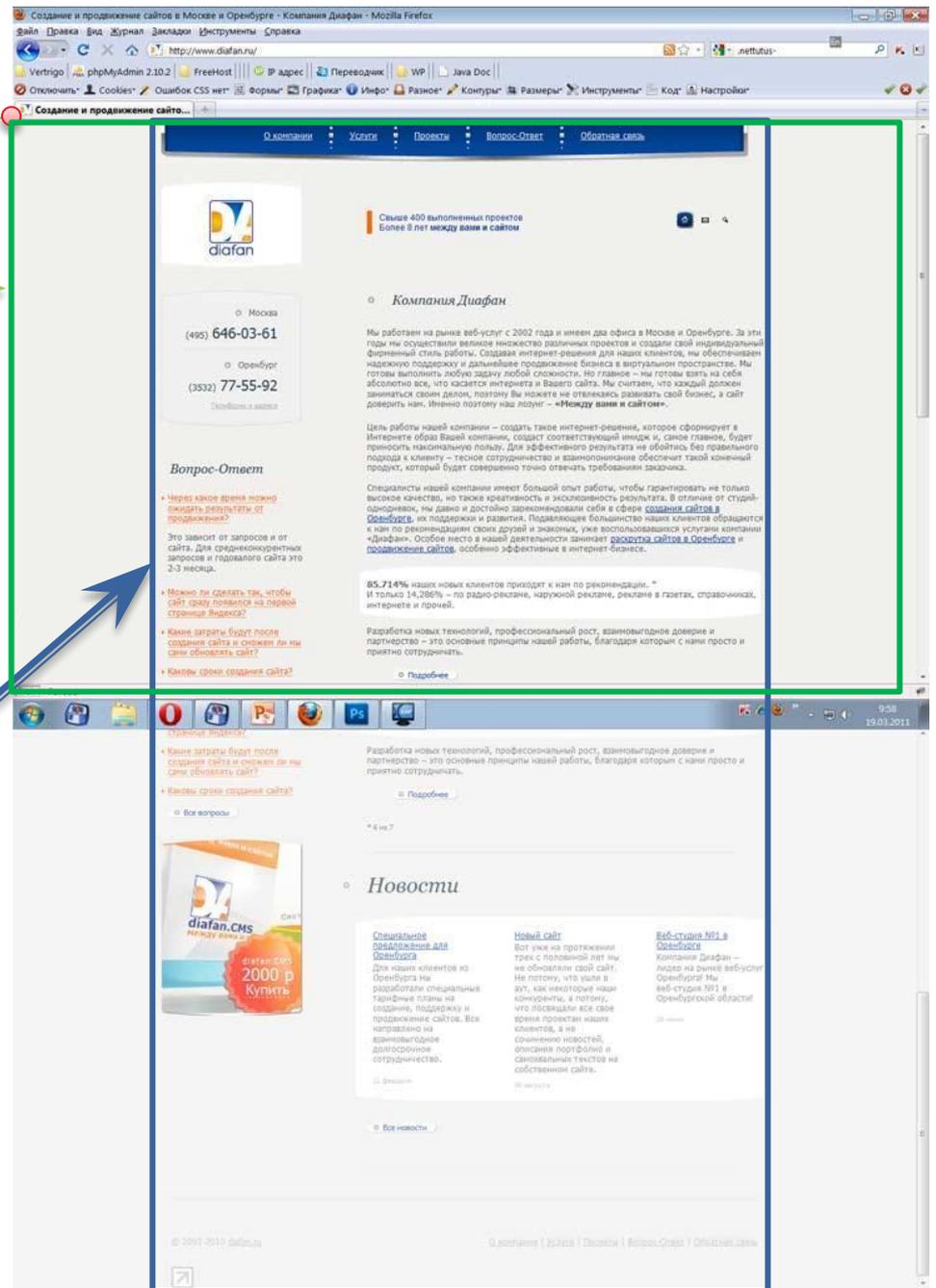
```
.myblock {  
  margin : 10px 20px;  
}
```

Точка привязки
(origin point)

Область просмотра
браузера (viewport)

Размер этой области
зависит от разрешения
экрана – и она привязана
к элементу **html**

Область страницы
(canvas)



Этапы расчета величин в браузере

Загружается HTML-код документа, CSS файлы, подключается модуль CSS

2. Относительные величины шрифтов заданные в `%`, `em`, `ex`, ... преобразуются в абсолютные (`px`) без учета области просмотра.

3. Преобразование относительных величин `width`, `height`, `margin`, `padding`, `border-width` блоков заданных в `%`, `em`, `ex`, ... в абсолютные с учетом области просмотра.

Элемент HTML является связующим между документом и областью просмотра браузера.

То есть на этом этапе рассчитывается взаимное расположение блоков на странице

4. Округление значений и отображение документа

Ширина и высота блока

Если в свойствах блока не указаны

`width`, `height`, `padding`, `margin`, `border`

то значения `width` и `height` устанавливаются в значение `auto`

`height: auto` - это значит что если в элементе появляется какой-то контент, то высота блока будет определяться размером этого контента.

`width: auto` - это значит что ширина блока определяется шириной его контейнера (то есть шириной его родителя).

Например для элемента `<body>` родителем является элемент `<html>` и поэтому элемент `<body>` будет по ширине растянут на всю область просмотра, а его высота будет определяться его содержимым

Высота блока

Если в свойствах **указано значение `height`** то высота блока рассчитывается

- **`height: auto`** - его контентом;
- если **`height`** указана в **%** - то относительно высоты его контейнера (то есть родителя);
- если **`height`** указана в **em** - определяется кеглем шрифта текущей области;
- для корневого элемента `<html>` высота блока определяется относительно области просмотра

Ширина и высота блока

Если в свойствах **указано значение width** то ширина блока определяется

- **width:auto** - зависит от других свойств (таких как `margin`, `padding`, `border`);
- если указана в **%** - то относительно ширины его контейнера;
- если указана в **em** - определяется кеглем шрифта текущей области;



ЗАПОМНИТЬ !!!

При верстке страниц высота (`height`) блокам задается только в крайних случаях, и делать это нужно осознанно, учитывая то, что ограничение блока по высоте будет срезать его контент, если высота этого контента превышает высоту блока

Превышение ширины контента заданной ширины блока

Если количество контента внутри блока будет превышать заданные размеры блока то текстовый контент выйдет за границы блока

При этом он не оказывает никакого влияния на соседние элементы

Управление переполнением контентом блока осуществляется с помощью свойства **overflow**

001/overflow.html

overflow:	
hidden	контент который выходит за пределы блока обрезается
auto	полосы прокрутки у блока будут показываться когда количество контента превысит размеры блока
scroll	полосы прокрутки у блока будут показываться всегда (независимо от того сколько контента в блоке)
visible	Будет показан весь контент без полос прокрутки

Выполняет задачу взаимодействия с:

- соседними элементами по расположению на странице;
- между элементом и его контейнером;

Значение указывается

- в **px**;
- если значение указано в **em** - кеглем шрифта текущей области;
- если значение указано в **%** - то относительно **ширины** его контейнера (то есть его родителя);
- если значение указано как **auto** - то расчет производит сам браузер и зависит величина от других свойств (**width**);
Если задать для **margin** по вертикали значение **auto**, то оно всегда сбрасывается в нуль.
- могут применяться отрицательные значения;

Margin-collapse (взаимодействие margins смежных блоков)

```
margin-bottom: 150px;
```

150px



```
margin-top: 100px;
```

001/margin_negative.html

Запомнить

1. **Вертикальные margins** между стоящими друг под другом блоками будет равен наибольшему из них.

```
margin-bottom: -50px;
```

```
margin-top: 0;
```

50px

Запомнить

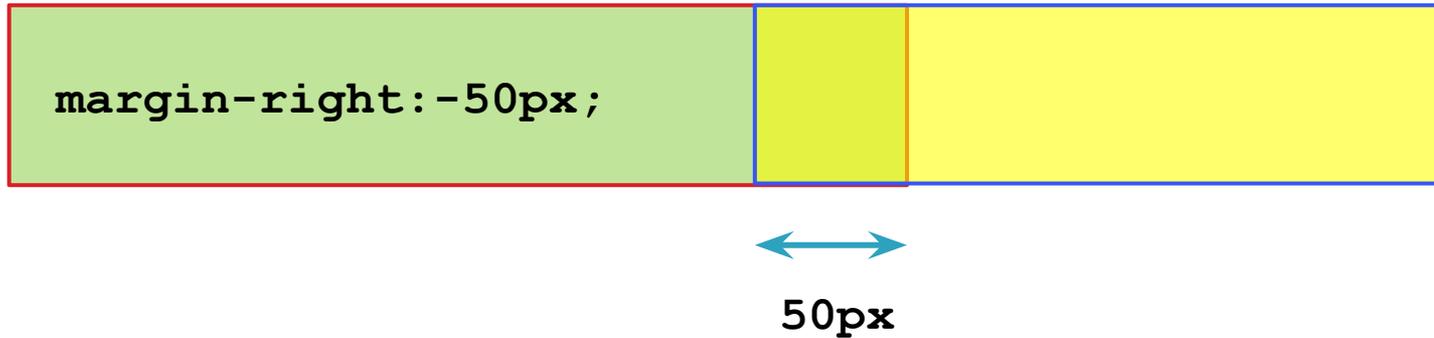
2. Если **верхнему** блоку задать **отрицательный `margin-bottom`** то это позволит нижнему блоку "заехать" на его территорию

```
margin-top: -50px;
```

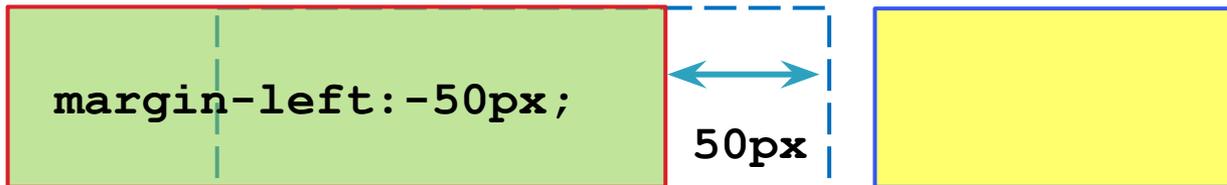
50px

```
margin-top: 0;
```

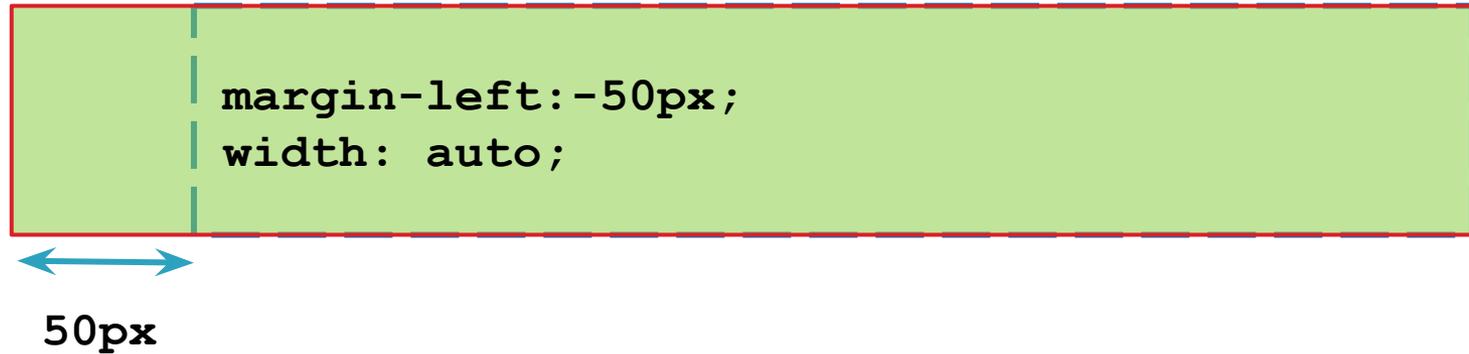
3. Если **верхнему** блоку задать **отрицательный `margin-top`** – блок смещается **вверх**



4. Если **левому** блоку задать **отрицательный `margin-right`** то это позволит **правому** блоку "заехать" на его территорию
5. **Отрицательный `margin-left`** - блок смещается **влево**



5. Если **у блока задана ширина**, и задать ему **отрицательный `margin-left`**, то блок смещается **влево**



6. Если у блока **не задана ширина**, и ему задать **отрицательный *margin-right*** (или ***margin-left***) то ширина блока расширится на этот `margin` в соответствующую сторону

Свойство блока padding

Значения указываются

- в **px**;
- если значение указано в **em** - кеглем шрифта текущей области;
- если значение указано в **%** - то относительно ширины его контейнера ;
- **!!!** отрицательные значения не применяются;

border блока

```
<div class = "myblock"></div>
```

```
.box {  
  border-width: 2px;  
  border-style: solid;  
  border-color: red;  
}
```



```
.box {  
  border: 2px solid red;  
}
```



border -

top
right
bottom
left

-

color
width
style



```
border-top-color: #e5e5e5;  
border-left-style: solid;  
border-bottom-width: 2em;  
border-right-style: dashed;
```

`background-color` : `transparent` | значение цвета;

`background-image` : `none` | `url(путь_к_изображению)`;

`background-repeat` : `repeat` | `no-repeat` | `repeat-x` | `repeat-y`

`background-position` : `0 0` | координаты | ключевые слова | `%`

`background-attachment` `fixed` | `scroll`

`background-color : transparent | значение ;`

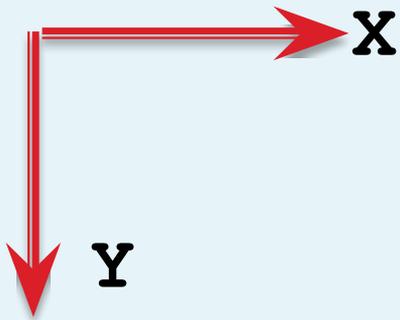
Областью для фона является `border-box` блока

Например `background-color : #d5d5d5 ;`

`background-image : url(путь к файлу) ;`

Например `background-image: none | url(img/picture.jpg) ;`

`background-repeat : repeat | no-repeat | repeat-x | repeat-y`



```
background-position: 0 0 | значение;
```

Областью для фонового изображения **по умолчанию** является *padding-box* блока

padding-box блока

200px

100px



```
background-position: 200px 100px;
```

top left	left top	0% 0%	в левом верхнем углу
top	top center center top	50% 0%	по центру вверху
right top	top right	100% 0%	в правом верхнем углу
left	left center	0% 50%	по левому краю и по центру
center	center center	50% 50%	по центру
right	right center center right	100% 50%	по правому краю и по центру
bottom left	left bottom	0% 100%	в левом нижнем углу
bottom	bottom center center bottom	50% 100%	по центру внизу
bottom right	right bottom	100% 100%	в правом нижнем углу

0 0
left top
top left

50% 0
center top
top center

100% 0
right top
top right



50% 50%
center



0 100%
left bottom
bottom left

50% 100%
center bottom
bottom center

100% 100%
right bottom
bottom right

background-attachment : **scroll** | **fixed**

Устанавливает, будет ли прокручиваться фоновое изображение вместе с содержимым элемента. Изображение может быть зафиксировано и оставаться неподвижным, либо перемещаться совместно с документом.

Фон блока одним правилом

Значения могут располагаться в произвольном порядке

background-attachment

```
background: #fcc url(..../img/im.gif) no-repeat top left fixed;
```

background-color

background-repeat

background-image

background-position

Свойство Inline block

display: inline-block;

Это блок вставленный в строку, при этом по умолчанию:

Например

- `img, object, embed, iframe` (элементы с замещающим контентом);
- `input, textarea, select, button` (элементы форм)
- `audio, video, canvas` (в HTML 5);

Особенности !!!

- его ширина определяется его контентом;
- нет переносов строки до начала и после окончания блока;
- внутри этого элемента можно разместить контент любого типа;
- не участвует в схемах с `margin-collapse`;
- элементу можно задавать `margin, padding`;

