

# CLIENT-SIDE WEB

## CSS POSITIONING

20

EXTERNAL TRAINING .NET/WEB

EPAM SARATOV. SUMMER 2020



# ПЛАН ЗАНЯТИЯ

- Блочная модель (Box Model)
- Ширина и высота
- Типы элементов
- Overflow
- Позиционирование
- Float

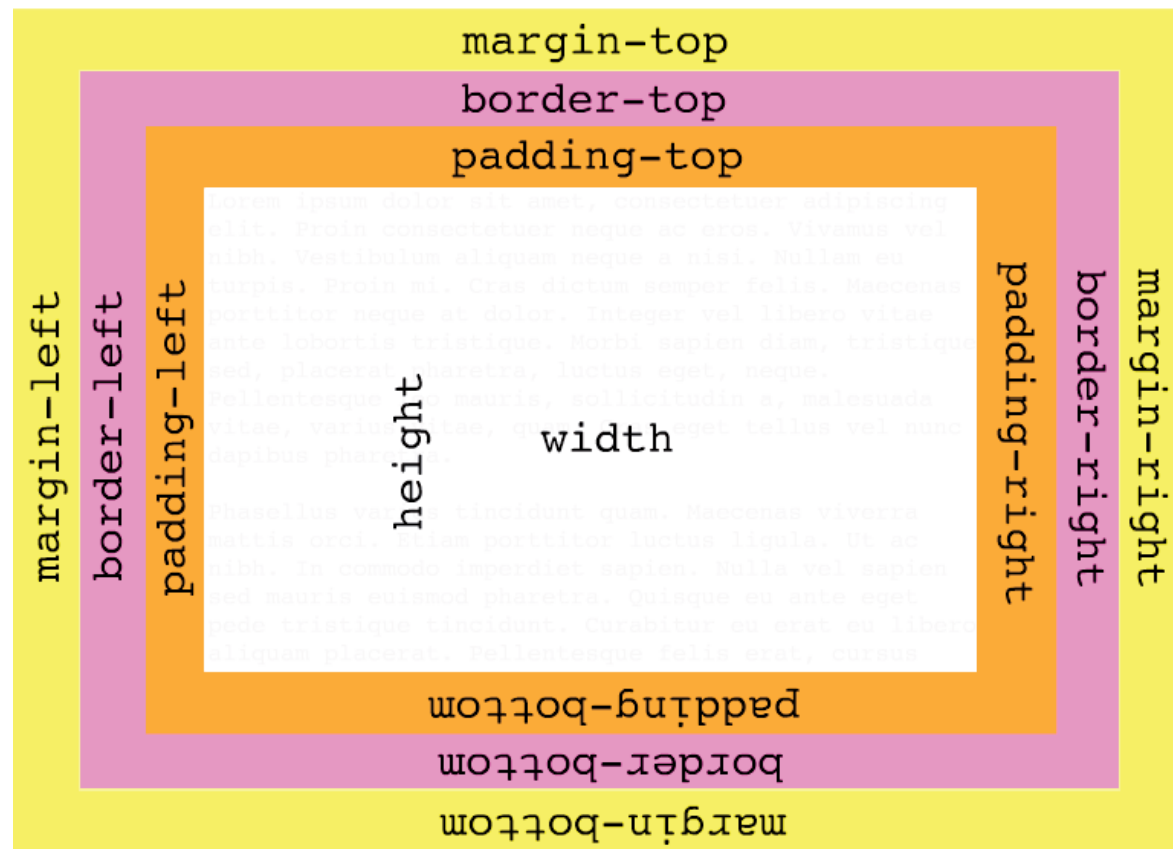


# BOX MODEL

Любой блочный элемент состоит из набора свойств, накладываемых друг на друга. Основа блока — это контент, или содержимое (текст, изображение и др.)

- width — ширина;
- height — высота;
- padding — поля вокруг контента (создают пустое пространство от контента до внутреннего края границ);
- border — границы;
- margin — невидимое пустое пространство от внешнего края границ;

Порядок влияния этих свойств на блок жёстко определён и не может быть нарушен.



# PADDING

- Padding — расстояние от внутреннего края границы или края блока до воображаемого прямоугольника, ограничивающего содержимое блока;
- Основное предназначение — создать пустое пространство вокруг содержимого элемента (например, текста), чтобы он не прилегал плотно к краю элемента.

```
p {  
  padding-top: 25px;  
  padding-bottom: 25px;  
  padding-right: 50px;  
  padding-left: 50px;  
}
```

```
p {  
  padding: 25px 50px;  
}
```

The padding property can have from one to four values.

- **padding: 25px 50px 75px 100px;**
  - top padding is 25px
  - right padding is 50px
  - bottom padding is 75px
  - left padding is 100px
- **padding: 25px 50px 75px;**
  - top padding is 25px
  - right and left paddings are 50px
  - bottom padding is 75px
- **padding: 25px 50px;**
  - top and bottom paddings are 25px
  - right and left paddings are 50px
- **padding: 25px;**
  - all four paddings are 25px

## Possible Values

Value	Description
<i>length</i>	Defines a fixed padding (in pixels, pt, em, etc.)
<i>%</i>	Defines a padding in % of the containing element

<http://jsfiddle.net/RCFKr/1/>



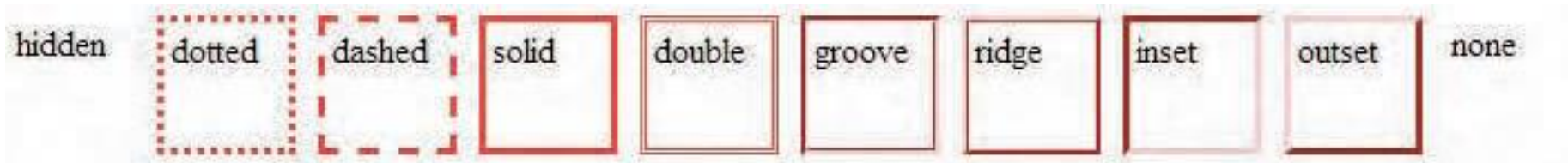
# BORDER

- Border — это граница вокруг padding`а элемента на одной или нескольких сторонах;
- Каждая граница обладает толщиной, стилем и цветом.

border-color
border-top-color
border-right-color
border-bottom-color
border-left-color

```
div#hidden {border-style: hidden;}
div#dotted {border-style: dotted;}
div#dashed {border-style: dashed;}
div#solid {border-style: solid;}
div#double {border-style: double;}
div#groove {border-style: groove;}
div#ridge {border-style: ridge;}
div#inset {border-style: inset;}
div#outset {border-style: outset;}
div#none {border-style: none; }
```

```
div#properties {
  border-top: 1px solid darkkhaki;
  border-right: 1px dashed darkkhaki;
  border-bottom: 1px solid darkkhaki;
  border-left: 1px dashed darkkhaki;
  background: lightyellow;
}
div#border {
  border: 1px dashed darkkhaki;
}
```



# MARGIN

- Margin — пустое пространство от внешнего края границы, полей или содержимого блока.

```
p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 150px;  
  margin-left: 50px;  
}
```

```
p {  
  margin: 100px 50px;  
}
```

The margin property can have from one to four values.

- **margin: 25px 50px 75px 100px;**
  - top margin is 25px
  - right margin is 50px
  - bottom margin is 75px
  - left margin is 100px
- **margin: 25px 50px 75px;**
  - top margin is 25px
  - right and left margins are 50px
  - bottom margin is 75px
- **margin: 25px 50px;**
  - top and bottom margins are 25px
  - right and left margins are 50px
- **margin: 25px;**
  - all four margins are 25px

## Possible Values

Value	Description
auto	The browser calculates a margin
<i>length</i>	Specifies a margin in px, pt, cm, etc. Default value is 0px
%	Specifies a margin in percent of the width of the containing element
inherit	Specifies that the margin should be inherited from the parent element

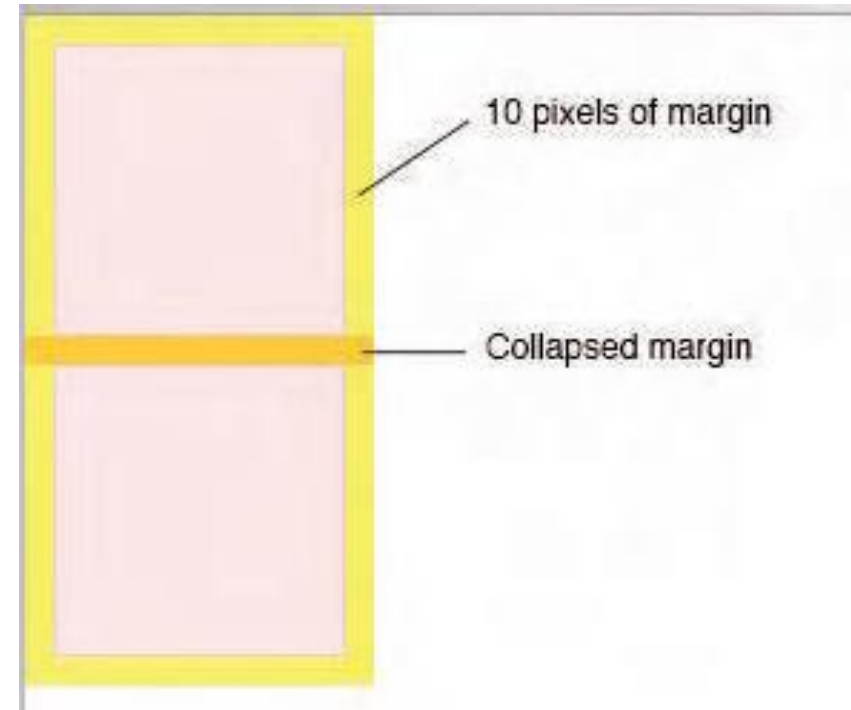
# ОСОБЕННОСТИ MARGIN

- Margin прозрачен, на него не распространяется цвет фона или фоновая картинка, заданная для элемента. Но если фон установлен у родительского элемента, он будет заметен и на margin т.к. он прозрачен;
- Margin может принимать отрицательное значение. Это приводит к сдвигу всего элемента в указанную сторону. Если задать `margin-left: -10px`, то это сдвинет элемент на десять пикселей влево;
- Margin, заданный в процентах, вычисляется относительно размеров контента блока (как по вертикали, так и по горизонтали).



# MARGIN COLLAPSING

- Для margin характерен collapsing — объединение вертикальных margin`ов близлежащих элементов;
- Два блока расположены вплотную, у каждого из них задан margin: 10px;
- Расстояние между блоками будет равно 10px, а не 20px. Это происходит за счёт collapsing, при котором близлежащие отступы объединяются.



```
<body>
  <div id='top-right-bottom-left'></div>
  <div id='top-right-bottom-left-1'></div>
</body>
```

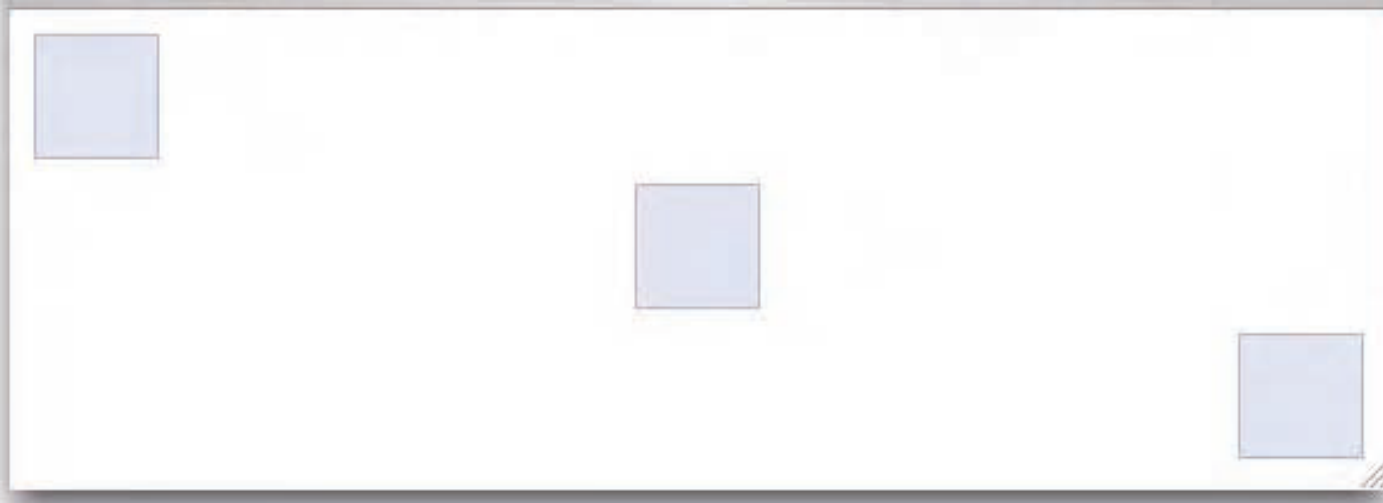
<http://jsfiddle.net/0dpo545o/>



# ВЫРАВНИВАНИЕ ЧЕРЕЗ MARGIN: AUTO

Используя значение `margin: auto`, можно выполнять горизонтальное выравнивание или центрирование элемента.

```
<body>
  <div id='left'>
  </div>
  <div id='center'>
  </div>
  <div id='right'>
  </div>
</body>
```



```
body {
  margin: 10px;
  padding: 0;
}
div {
  width: 50px;
  height: 50px;
  background: rgb(218, 220, 243);
  border: 1px solid rgb(154, 157, 203);
}
div#left {
  margin: 10px auto 10px 0;
}
div#center {
  margin: 10px auto;
}
div#right {
  margin: 10px 0 10px auto;
}
```

# РАСЧЁТ РАЗМЕРОВ БЛОКА

Реальная ширина блока складывается из нескольких компонентов:

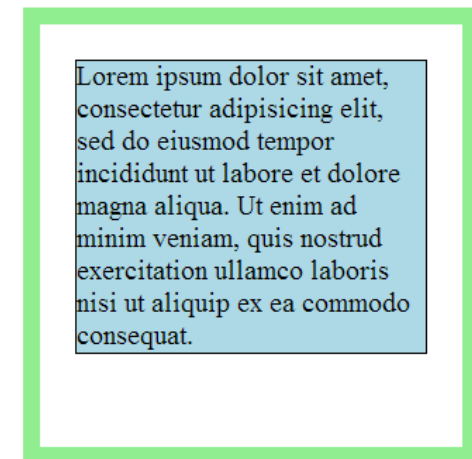
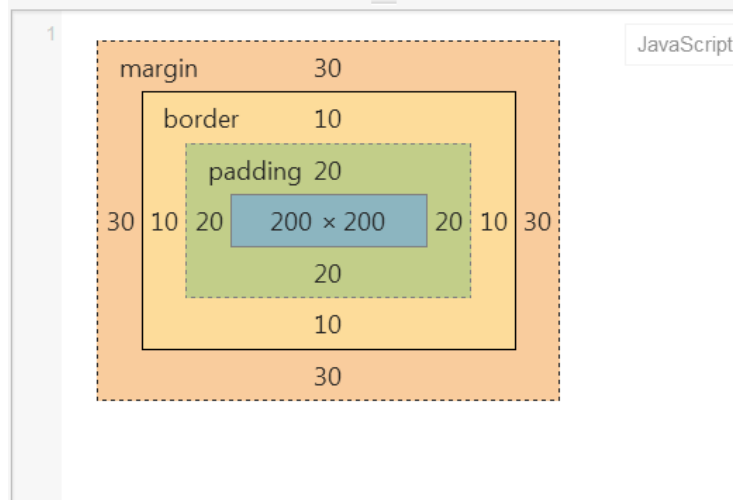
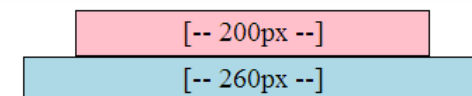
- width + padding-left + padding-right + border-left + border-right + margin-left + margin-right

Реальная высота вычисляется аналогично:

- height + padding-top + padding-bottom + border-top + border-bottom + margin-top + margin-bottom

```
1 <div class="line line-s">[-- 200px --]</div>
2 <div class="line line-b">[-- 260px --]</div>
3
4 <div class="container">
5   <p>Lorem ipsum dolor sit amet, consectetur
   adipisicing elit, sed do eiusmod
   tempor incididunt ut labore et dolore magna aliqua.
   Ut enim ad minim veniam, quis nostrud exercitation
   ullamco laboris nisi ut aliquip ex ea commodo
   consequat.
6   </p>
7 </div>
```

```
1 .container {
2   width: 200px;
3   height: 200px;
4   border: 10px solid lightgreen;
5   padding: 20px;
6   margin: 30px;
7 }
8 p {
```



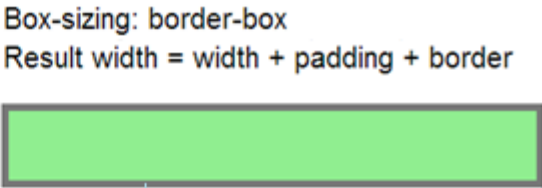
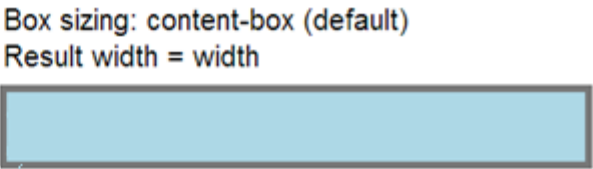
<http://jsfiddle.net/Qf5nm/4/>

# BOX-SIZING

Применяется для изменения алгоритма расчёта ширины и высоты элемента.

content-box	Основывается на стандартах CSS, при этом свойства width и height задают ширину и высоту контента и не включают в себя значения отступов, полей и границ.
border-box	Свойства width и height включают в себя значения полей и границ, но не отступов (margin). Эта модель используется браузером Internet Explorer в режиме несовместимости.
padding-box	Свойства width и height включают в себя значения полей, но не отступов (margin) и границ (border).

```
.border-box {
  box-sizing: border-box;
  background: lightgreen;
}
.content-box {
  box-sizing: content-box;
  background: lightblue;
}
.container div {
  width: 300px;
  padding: 10px;
  margin: 0;
  border: 4px solid #777;
}
```



# BOX-SIZING ПРИМЕР

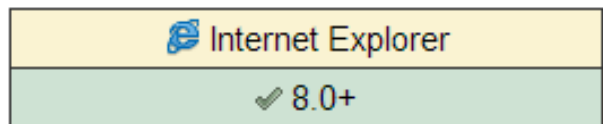
```
1 <p>Default behavior</p>
2 <div>
3   <input type="text" class="content-box"></input>
4 </div>
5
6 <p>Box-sizing: border-box</p>
7 <div>
8   <input type="text" class="border-box"></input>
9 </div>
```

HTML

```
1 .border-box {
2   box-sizing: border-box;
3   background: lightgreen;
4 }
5
6 div {
7   width: 30%;
8   border: 2px solid black;
9   margin: 0 0 30px 0;
10  padding: 10px;
11 }
12
13 input {
14   width: 100%;
15   padding: 10px;
16   margin: 0;
17   background: lightblue;
18 }
```

1

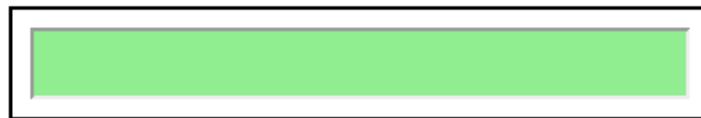
JavaScript



Default behavior



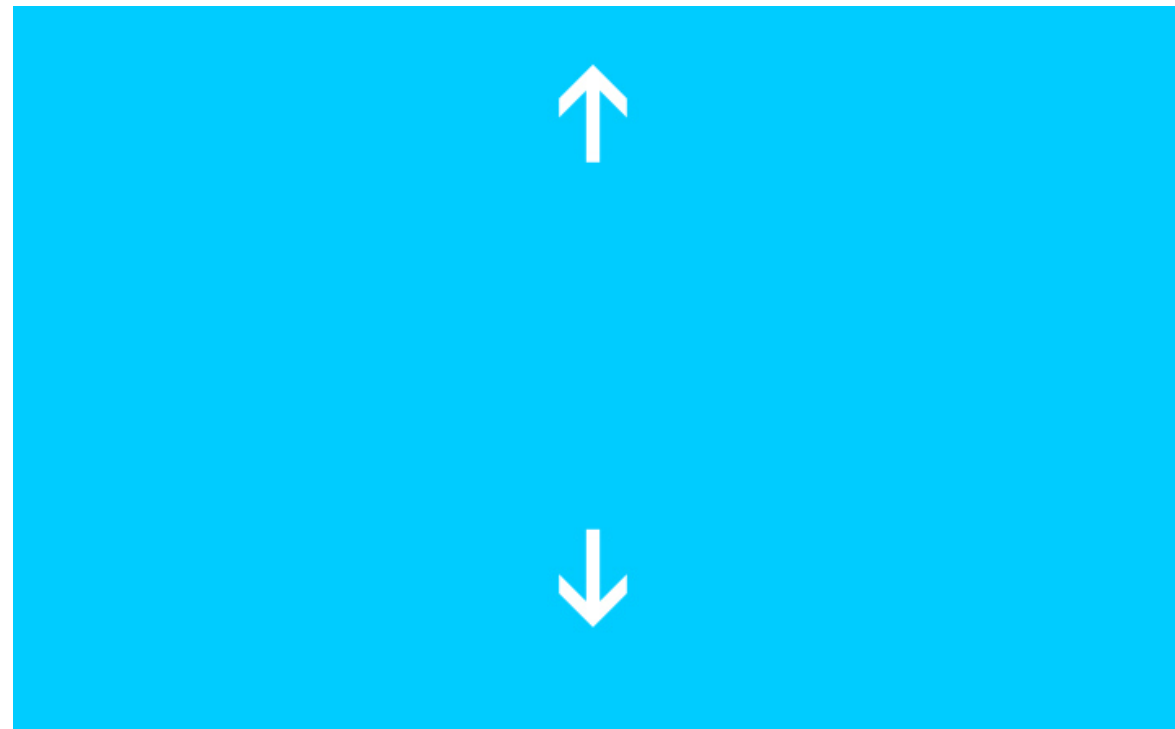
Box-sizing: border-box



<http://jsfiddle.net/Zuxf7/3/>

# ВЫСОТА В ПРОЦЕНТАХ

- Принципы построения ширины и высоты существенно различаются. Это касается того случая, когда значения `width` и `height` не указаны и принимаются по умолчанию как `auto`;
- Ширина может быть задана в процентах от ширины родителя;
- Для задания высоты в процентах необходимо явно задавать высоту родителя.

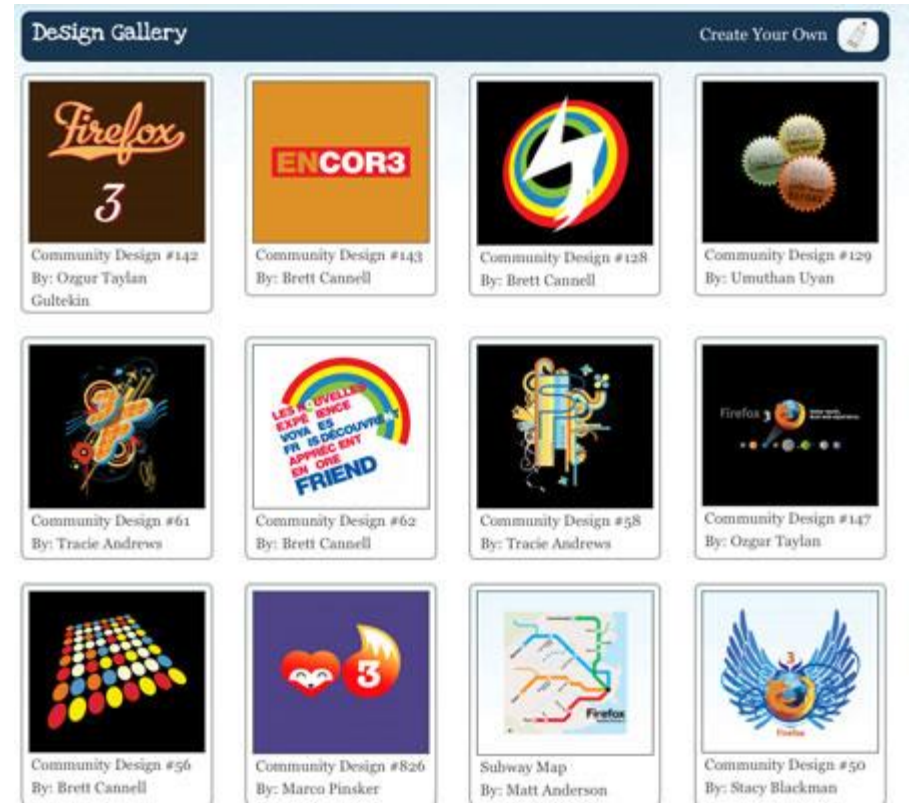


<http://jsfiddle.net/Z9CbY/>



# ОСНОВНЫЕ ТИПЫ ЭЛЕМЕНТОВ

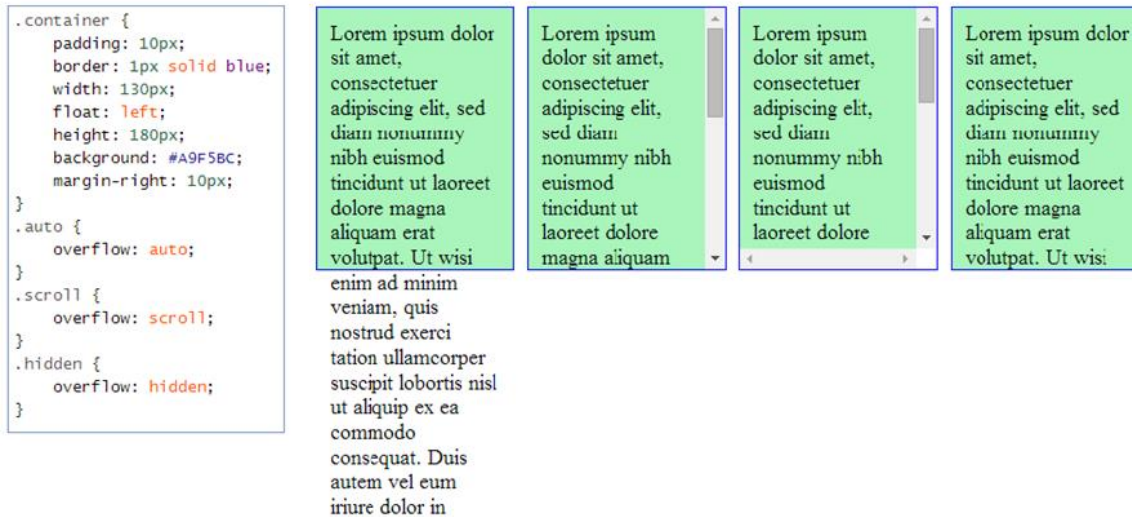
- block
  - Занимает всю доступную ширину, высота определяется содержимым, всегда начинается с новой строки;
- inline
  - Являются непосредственной частью строки, нельзя задавать ширину и высоту, нельзя указать вертикальные отступы;
- inline-block
  - Продолжает находиться в строке (inline), но при этом может иметь свойства блочного элемента.



<http://jsfiddle.net/k6k96/1/>

# OVERFLOW

- Свойство `overflow` управляет отображением содержания блочного элемента, если оно целиком не помещается и выходит за область заданных размеров.

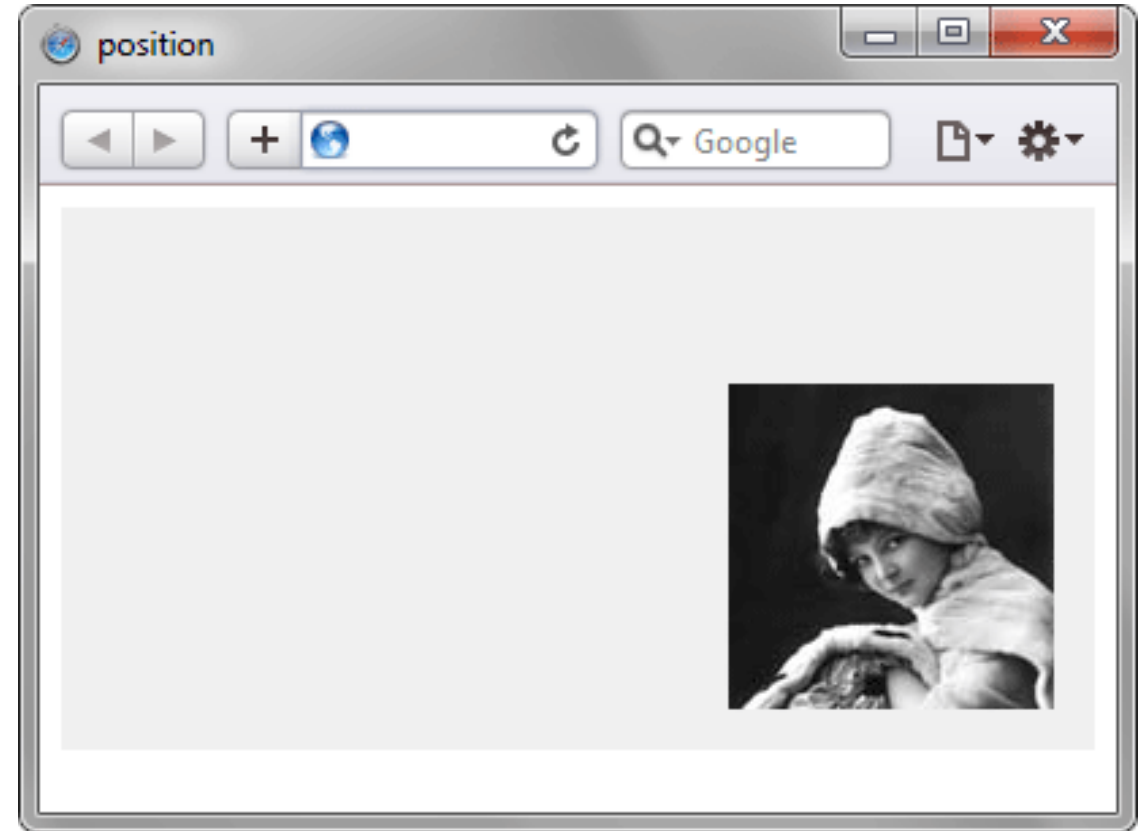


`overflow: auto | hidden | scroll | visible | inherit`

<http://jsfiddle.net/Qr7Up/>

# POSITION

- Устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице.



position: absolute | fixed | relative | static | inherit

# ОТНОСИТЕЛЬНОЕ ПОЗИЦИОНИРОВАНИЕ (RELATIVE)

- Элемент позиционируется относительно своего предполагаемого положения;
- Смещение задаётся указанием координат left/right/top/bottom.

Здесь написан какой то текст

Здесь написан какой то текст

Здесь **position: relative** написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

<https://jsfiddle.net/turdmittii/1dqawvug/2/>

# АБСОЛЮТНОЕ ПОЗИЦИОНИРОВАНИЕ (ABSOLUTE)

- Элемент исчезает с того места, где он должен был быть, и позиционируется заново. Остальные элементы располагаются так, как будто этого элемента никогда не было;
- Координаты top/bottom/left/right для нового местоположения отсчитываются от ближайшего родителя с позиционированием, отличным от static;
- Ширина элемента устанавливается по содержимому;
- Элемент получает display: block, которые перекрывается почти все возможные display.

Здесь написан какой то текст

Здесь **position: absolute** написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

<http://jsfiddle.net/turdmittii/x3nsxkm5/1/>



# ФИКСИРОВАННОЕ ПОЗИЦИОНИРОВАНИЕ (FIXED)

- Считается подвидом абсолютного позиционирования;
- Отличается тем, что позиционируется относительно window (окна браузера).

Здесь написан какой то текст **position: fixed**

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

Здесь написан какой то текст

<http://jsfiddle.net/turdmittii/4wgk7vt6/1/>

# FLOAT

- Определяет, по какой стороне будет выравниваться элемент, при этом остальные элементы будут обтекать его с другой стороны.

`float: left | right | none | inherit;`

Пример: <http://jsfiddle.net/turdmittii/orhk89jb/>



Jupiter is the fifth planet from the Sun and the largest planet in the Solar System.



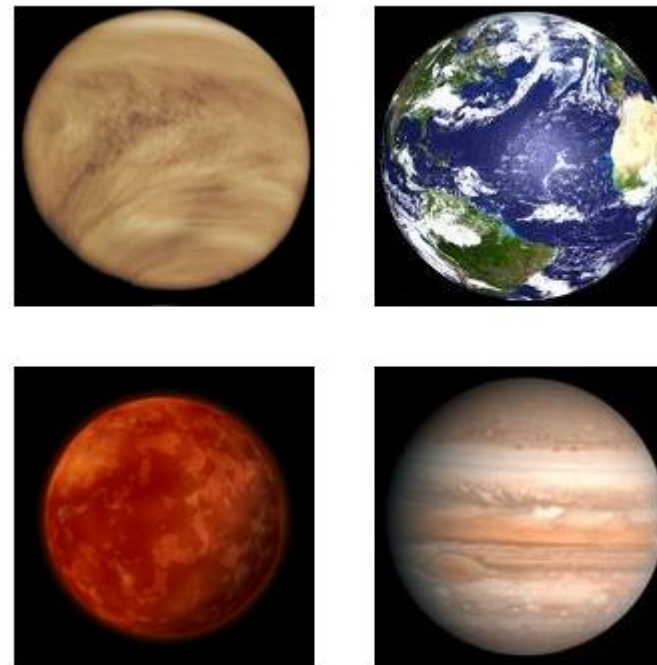
Jupiter is the fifth planet from the Sun and the largest planet in the Solar System.

Jupiter is the fifth planet from the Sun and the largest planet in the Solar System.



# ОСОБЕННОСТИ FLOAT

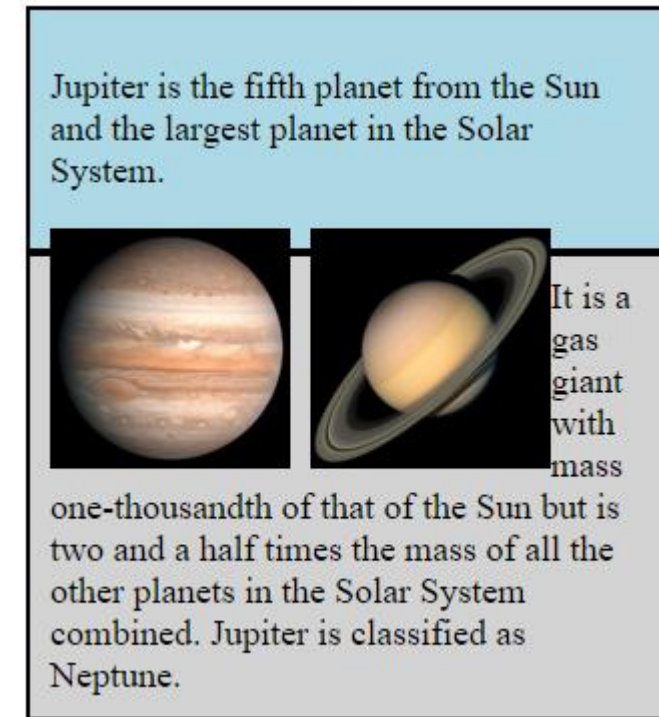
- Ширина обтекаемого блока определяется по его содержимому;
- Элемент при наличии float неявно получает display: block;
- Для вертикальных margin`ов отключается collapsing, в отличие от обычных блочных элементов;
- Если пространства по горизонтали не хватает для того, чтобы вместить элемент, то он сдвигается вниз до тех пор, пока не начнёт помещаться.



Пример: <http://jsfiddle.net/turdmitrii/D7T6v/3/>

# ПРОБЛЕМА С ОБТЕКАЕМЫМИ ЭЛЕМЕНТАМИ

1. Элемент позиционируется как обычно, а затем вынимается из документа потока и сдвигается влево (для left) или вправо (для right) до того момента, пока не коснётся либо границы родителя, либо другого элемента с float.
2. Другие непозиционированные блочные элементы без float ведут себя так, как будто элемента с float нет, так как он убран из потока.
3. Строки (inline-элементы), напротив, «знают» о float и обтекают элемент по сторонам.



Пример: <http://jsfiddle.net/turdmittii/teY2U/8/>

- Применение этого свойства сдвигает элемент вниз до тех пор, пока не закончатся float`ы слева / справа / с обеих сторон.

`clear: left | right | both;`

Jupiter is the fifth planet from the Sun and the largest planet in the Solar System.



It is a gas giant with mass one-thousandth of that of the Sun but is two and a half times the mass of all the other planets in the Solar System combined. Jupiter is classified as Neptune.

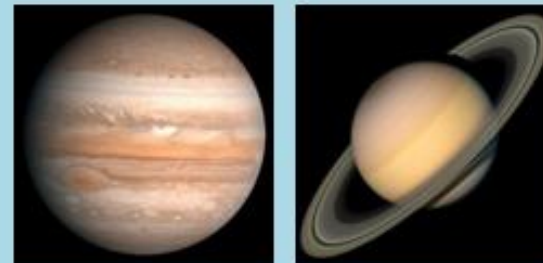
Пример: <http://jsfiddle.net/turdmitrii/teY2U/7/>



# CLEARFIX

- Вместо добавления в разметку лишнего элемента только ради clear, можно использовать псевдо-элемент :after;
- Данный приём получил название Clearfix;
- Данная техника распространена и является довольно популярной.

Jupiter is the fifth planet from the Sun and the largest planet in the Solar System.



It is a gas giant with mass one-thousandth of that of the Sun but is two and a half times the mass of all the other planets in the Solar System combined. Jupiter is classified as Neptune.

Пример: <http://jsfiddle.net/turdmitrii/teY2U/10/>



**THANKS FOR  
ATTENTION!**

VIKENTII EKGART, SARATOV, RUSSIA