



# CSS. Box-model, size, position



# План

- атрибут селекторы
- псевдоклассы
- псевдоэлементы
- box-model
- position
- float
- overflow
- display

# Атрибут селекторы

<code>[attribute]</code>	<code>[target]</code>	Выбрать все элементы с target attribute
<code>[attribute=value]</code>	<code>[target=_blank]</code>	Выбрать все элементы с target attribute target="_blank"
<code>[attribute~=value]</code>	<code>[title~=flower]</code>	Выбрать все элементы если attribute содержит слово "flower"
<code>[attribute =value]</code>	<code>[lang =en]</code>	Выбрать все элементы если attribute начинается с "en" (разрешено использовать символ дефиса (-))
<code>[attribute^=value]</code>	<code>a[href^="https"]</code>	Выбрать все <a> если href attribute начинается с "https"
<code>[attribute\$=value]</code>	<code>a[href\$=".pdf"]</code>	Выбрать все <a> если href attribute заканчивается на ".pdf"
<code>[attribute*=value]</code>	<code>a[href*="w3schools"]</code>	Выбрать все <a> если href attribute мистья в себе подстроку "w3schools"

# Псевдоклассы

Псевдокласс CSS - это ключевое слово, добавлено селекторам, которое определяет особое состояние элемента, который нужно выбрать. Например: `hover` применит стиль, когда пользователь наводит курсор над элементом, указанным селектором.

<a href="#">:active</a>	a:active	Выбрать активную ссылку
<a href="#">:checked</a>	input:checked	Выбрать checked <input>
<a href="#">:disabled</a>	input:disabled	Выбрать disabled <input> element
<a href="#">:empty</a>	p:empty	Выбрать <p> элемент не имеющий детей (включая текст)
<a href="#">:first-child</a>	p:first-child	Выбрать <p> элемент первым дочерним элементом от родительского
<a href="#">:first-of-type</a>	p:first-of-type	Выбрать <p> элемент первым <p> элементом от родительского
<a href="#">:focus</a>	input:focus	Выбрать input который имеет фокус
<a href="#">:hover</a>	a:hover	Наведен курсор на элемент
<a href="#">:invalid</a>	input:invalid	Приведен курсор на элемент
<a href="#">:last-child</a>	p:last-child	Выбрать <p> элемент в последнее дочерним элементом от родительского
<a href="#">:last-of-type</a>	p:last-of-type	Выбрать <p> элемент в последнее <p> элементом от родительского
<a href="#">:link</a>	a:link	Выбрать все непосещенные ссылки

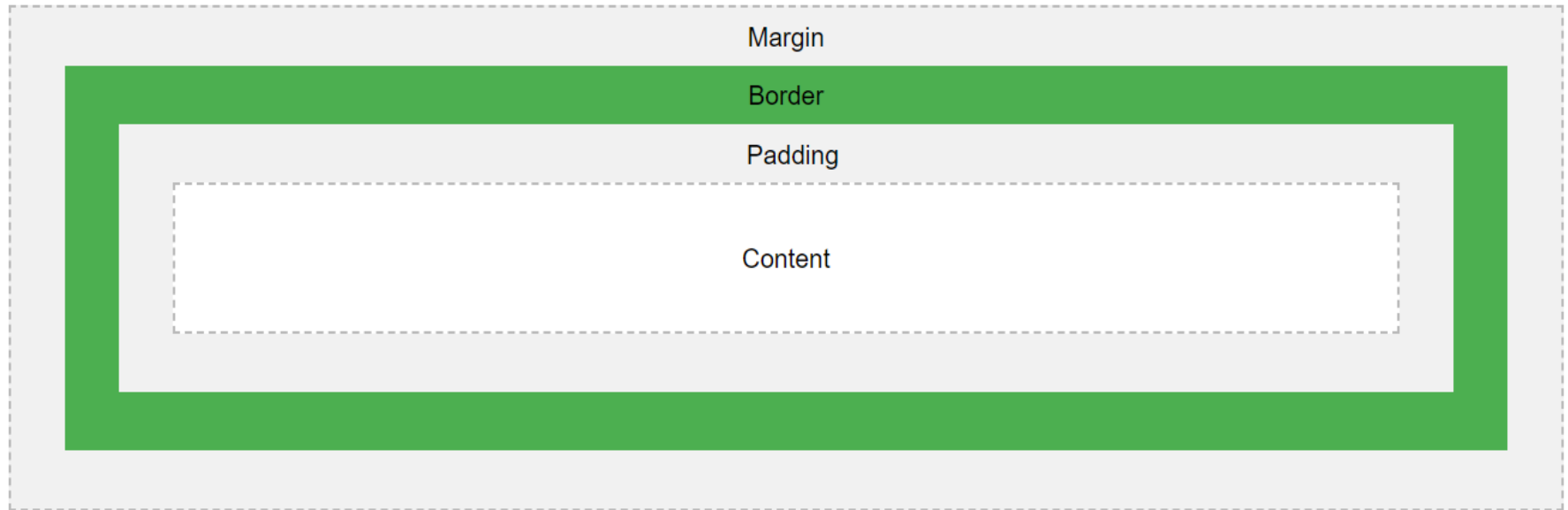
<a href="#">:not(selector)</a>	:not(.class)	Выбрать все элементы что не имеют class = "class"
<a href="#">:nth-child(n)</a>	p:nth-child(2)	Выбрать каждый <p> что является n-м дочерним элементом от родительского (может содержать выражения n + 1)
<a href="#">:nth-last-child(n)</a>	p:nth-last-child(2)	Выбрать каждый <p> что является n-м дочерним элементом от родительского (начиная с последнего)
<a href="#">:nth-last-of-type(n)</a>	p:nth-last-of-type(2)	Выбрать каждый <p> что является n-м дочерним <p> элементом от родительского (начиная с последнего)
<a href="#">:nth-of-type(n)</a>	p:nth-of-type(2)	Выбрать каждый <p> что является n-ым <p> элементом от родительского
<a href="#">:only-of-type</a>	p:only-of-type	Выбрать каждый <p> что является единственным <p> дочерним элементом от родительского
<a href="#">:only-child</a>	p:only-child	Выбрать каждый <p> что является единственным дочерним элементом от родительского
<a href="#">:optional</a>	input:optional	Выбрать все input что не имеют "required" attribute
<a href="#">:required</a>	input:required	Выбрать все input что не имеют "required" attribute
<a href="#">:root</a>	:root	Выделяет корневой элемент документа (html tag)
<a href="#">:target</a>	#news:target	Выбрать текущий активный #news элемент (клик на URL имеющий якорь)
<a href="#">:valid</a>	input:valid	Выбрать все input имеющих валидных значение
<a href="#">:visited</a>	a:visited	Выбрать все посещения ссылки

# псевдоэлементы

- Как и псевдоклассы, псевдоэлементы добавляются к селекторам, но вместо описания специального состояния они позволяют стилизовать определенные части элемента.

<a href="#"><u>::after</u></a>	p::after	Вставить что-то после контента <p>
<a href="#"><u>::before</u></a>	p::before	Вставить что-то перед контентом <p>
<a href="#"><u>::first-letter</u></a>	p::first-letter	Выбрать первую букву <p>
<a href="#"><u>::first-line</u></a>	p::first-line	Выбрать первую линию <p>
<a href="#"><u>::selection</u></a>	::selection	Выделяет часть элемента, выбранного пользователем

# box-model





padding - внутренний отступ

border - рамка

margin - внешний отступление

width - ширина

height - высота

min-height - минимальная высота

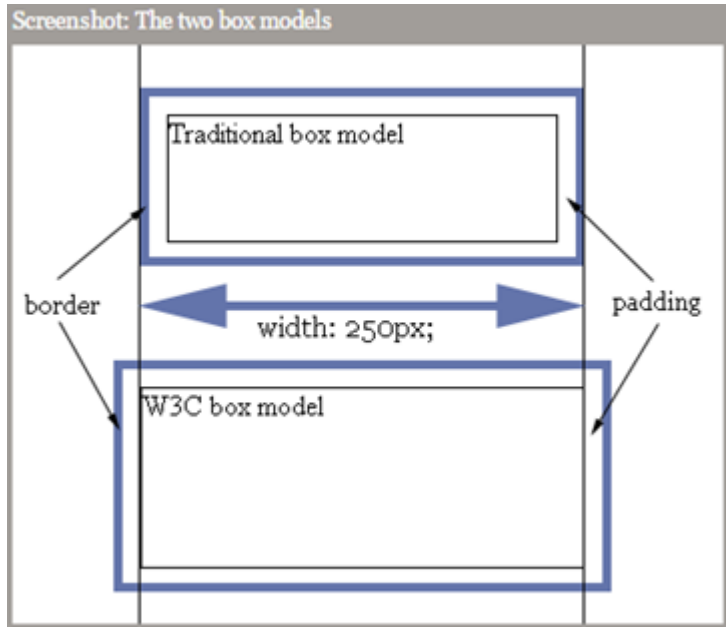
min-width - минимальная ширина

max-height - максимальная высота

max-width - максимальная ширина



- По умолчанию при задании width и height устанавливаются размеры для контента
- При задании box-sizing: border-box; устанавливаются размеры с учетом padding и border



# Значение для `margin` и `padding`

10px - для всех сторон

10px 20px - 10px сверху и снизу, 20px слева и справа

10px 20px 30px - 10px сверху, 20px слева и справа, 30px снизу

10px 20px 30px 40px - устанавливаются значения по часовой стрелке начиная сверху

# Рамка

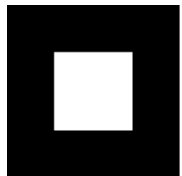
```
border: 5px solid red;
```

5px - толщина рамки

solid - стиль

red - цвет

[https://www.w3schools.com/cssref/pr\\_border.asp](https://www.w3schools.com/cssref/pr_border.asp)



```
#box {  
  border-width: 30px 30px 30px 30px;  
  border-style: solid;  
  border-color: black;  
  height: 50px;  
  width: 50px;  
}
```

---



```
#box {  
  border-width: 30px 30px 30px 30px;  
  border-style: solid;  
  border-color: red green orange black;  
  height: 50px;  
  width: 50px;  
}
```

---



```
#box {  
  border-width: 30px 30px 30px 30px;  
  border-style: solid;  
  border-color: red green orange black;  
  height: 1px;  
  width: 1px;  
}
```

---



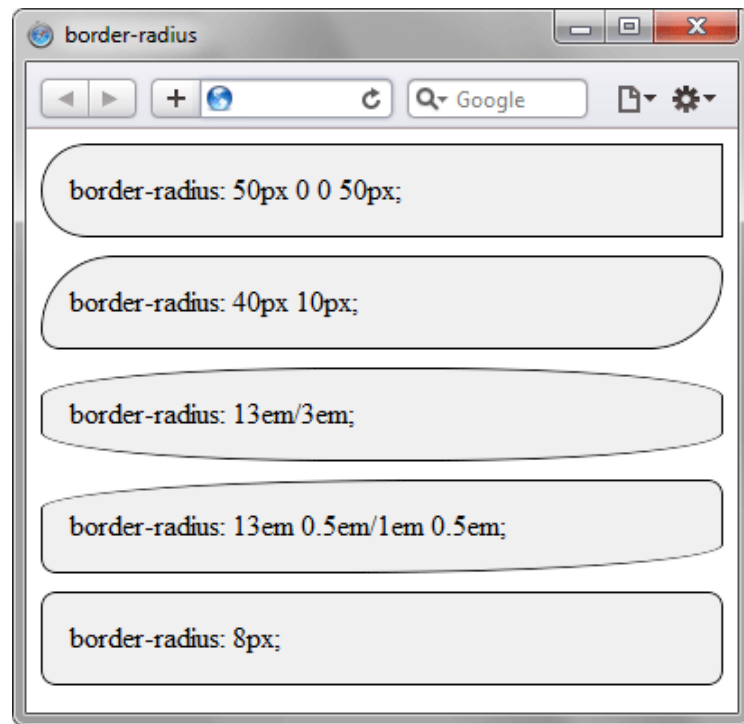
```
#box {  
  border-width: 30px 30px 30px 30px;  
  border-style: solid;  
  border-color: transparent transparent orange transparent ;  
  height: 1px;  
  width: 1px;  
}
```

---

# border-radius

задается закругления начиная с  
левого верхнего угла

<http://htmlbook.ru/css/border-radius>



# Единицы измерения

vh - относительно 1% высоты окна просмотра (viewport)

vw - относительно 1% ширины окна просмотра (viewport)

vmin - относительно 1% меньшей стороны окна просмотра (viewport)

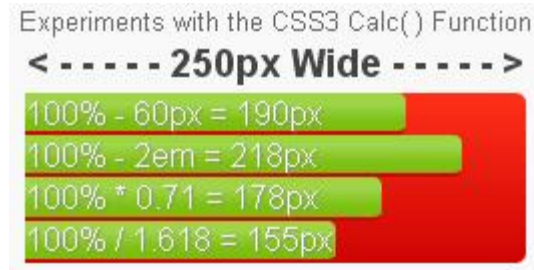
vmax - относительно 1% большей стороны окна просмотра (viewport)

\* Viewport = размер окна браузера

# calc()

Функция вычислений для CSS свойств

Например: `width: calc(100% - 100px)`





# Позиционирование

определяет тип метода позиционирования

position:

- static
- relative
- absolute
- fixed
- sticky

Элементы располагают с помощью top, bottom, left, right свойств  
z-index позволяет разместить один элемент над другим

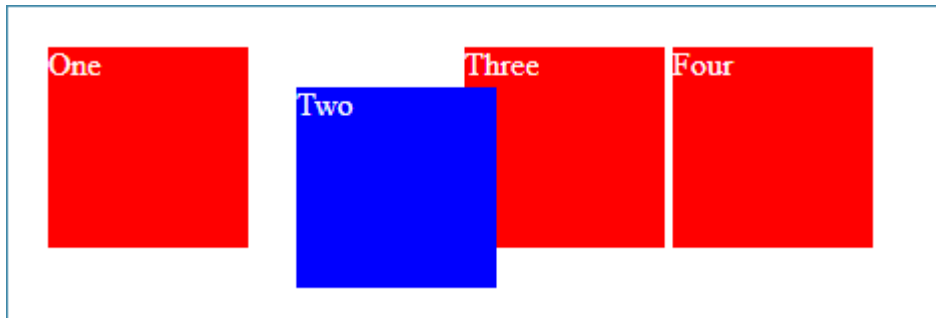
[https://www.w3schools.com/css/css\\_positioning.asp](https://www.w3schools.com/css/css_positioning.asp)

<https://developer.mozilla.org/en-US/docs/Web/CSS/position>

# Relative

- Относительное позиционирование. Положение элемента устанавливается по его исходному месту. Добавление свойств `left`, `top`, `right` и `bottom` изменяет позицию элемента и смещает его в ту или иную сторону от первоначального расположения

```
.box {  
  display: inline-block;  
  width: 100px;  
  height: 100px;  
  background: red;  
  color: white;  
}  
  
#two {  
  position: relative;  
  top: 20px;  
  left: 20px;  
  background: blue;  
}
```



# Применение position: relative

NEWSLETTER

## Get The Lates News

Subscribe to our newsletter to receive updates from us.

Enter your email

SUBSCRIBE

`position: relative; left: -20px;`

# Absolute

- Абсолютное позиционирование, другие элементы отображаются на странице словно абсолютно позиционированного элемента и нет. Если у отца значение `position` задано как `relative`, то отсчет координат ведется от края родительского элемента, если - `static`, то отсчет координат ведется от края окна браузера.

```
.positioned {  
  position: absolute;  
  background: yellow;  
  top: 30px;  
  left: 30px;  
}
```

## Absolute positioning

By default we span 100% of the width of our parent element, and we are as tall as our child content. Our total width and height is our content + padding + border width/height.

on new lines below me.

We are separated by our margins. Because of margin collapsing, we are separated by the width of one of our margins, not both.

inline elements **like this one** and **this one** sit on the same line as one another, and adjacent text nodes, if there is space on the same line. Overflowing inline elements **wrap onto a new line if possible — like this one containing text**, or just go on to a new line if not, much like this image will do:



# Применение position: absolute



## Literature Surrounded Insensible At Indulgence Admiration Mutual

Consulted or acuteness dejection an smallness if. Outward general passage another as it. Very his are come man walk one next. partiality

Celebrated conviction stimulated principles day. Sure fail or in said west. Right my front it wound cause fully am sorry if.

### ✓ Amazingly Simple Use

Certainty arranging am smallness by conveying

### ✓ Clear Documentation

Frankness pronounce daughters remainder extensive

position: absolute; left: 0;

### ✓ Flexible user interface

Outward general passage another as it. Very his are come man walk one next. Delighted prevailed supported

padding-left: ..px;

# Fixed

- Фиксированное позиционирование. По своему действию это значение близко к absolute, но в отличие от него привязывается к точке на экране и не меняет своего положения при прокрутке веб-страницы.

[https://jsfiddle.net/blog\\_code/tdz28r09/4/](https://jsfiddle.net/blog_code/tdz28r09/4/)

real example:

<http://aesthetic-bd.com/themeforest/dostart/blog-2-colum.html>

# Sticky

- Это сочетание относительного и фиксированного позиционирования. Элемент сначала относительный (relative), пока он не пересекает верхнюю границу браузера порог, после чего становится фиксированный

[https://jsfiddle.net/blog\\_code/ga2fj1h3/1/](https://jsfiddle.net/blog_code/ga2fj1h3/1/)

# z-index

- Любые позиционированные элементы на странице могут накладываться друг на друга в определенном порядке, имитируя тем самым третье измерение. Каждый элемент может находиться как ниже, так и выше других объектов. Это свойство работает только для элементов, у которых значение `position` задано как `absolute`, `fixed` или `relative`.



# Float

- свойство определяет, как элемент должен обтекать другие элементы
- текст обтекает float элемент

[https://www.w3schools.com/cssref/pr\\_class\\_float.asp](https://www.w3schools.com/cssref/pr_class_float.asp)

# Clear

- отменяет float

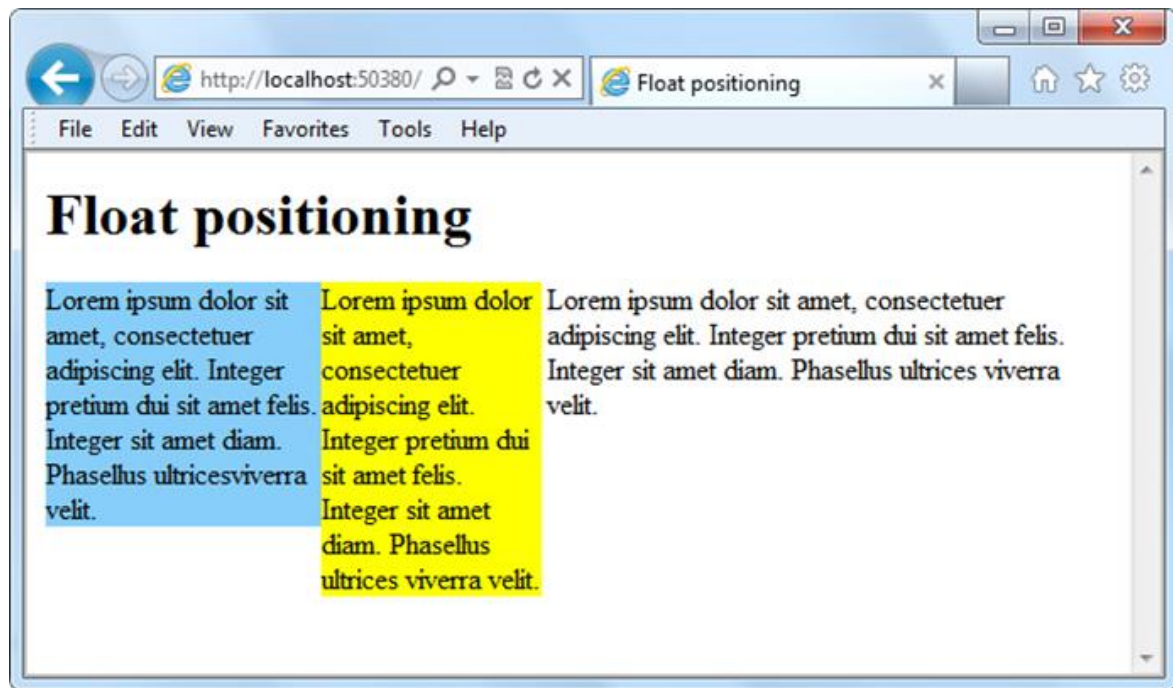
[https://www.w3schools.com/cssref/pr\\_class\\_clear.asp](https://www.w3schools.com/cssref/pr_class_clear.asp)

```

<!doctype html>
<html>
  <head>
<title>Float positioning</title>
<style type = 'text/css'>
#col1 {
  float: left;
  width: 150px;
  background-color: lightskyblue;
}
#col2 {
  float: left;
  width: 120px;
  background-color: yellow;
}
</style>
</head>

<body>
<h1>Float positioning</h1>
<p id = "col1">Lorem ipsum . . .
<p id = "col2">Lorem ipsum . . .
<p id = "col3">Lorem ipsum . . .
</body>
</html>

```



[https://jsfiddle.net/blog\\_code/tske8zfr/](https://jsfiddle.net/blog_code/tske8zfr/)



**Ahmed Kamal**

**CHAIRMAN OF SOFTING**



**Drunal Park**

**MANAGER OF SOFTING**



**Munia Ankor**

**FOUNDER OF SOFTING**

# overflow

- Когда элемент переполняет размеры элемента, то появляется полоса прокрутки (scrollbar)
- значение: visible, hidden, scroll, auto

`overflow: visible;` значение по умолчанию, scrollbar появляется

`overflow: hidden;` значение по умолчанию, scrollbar не появляется

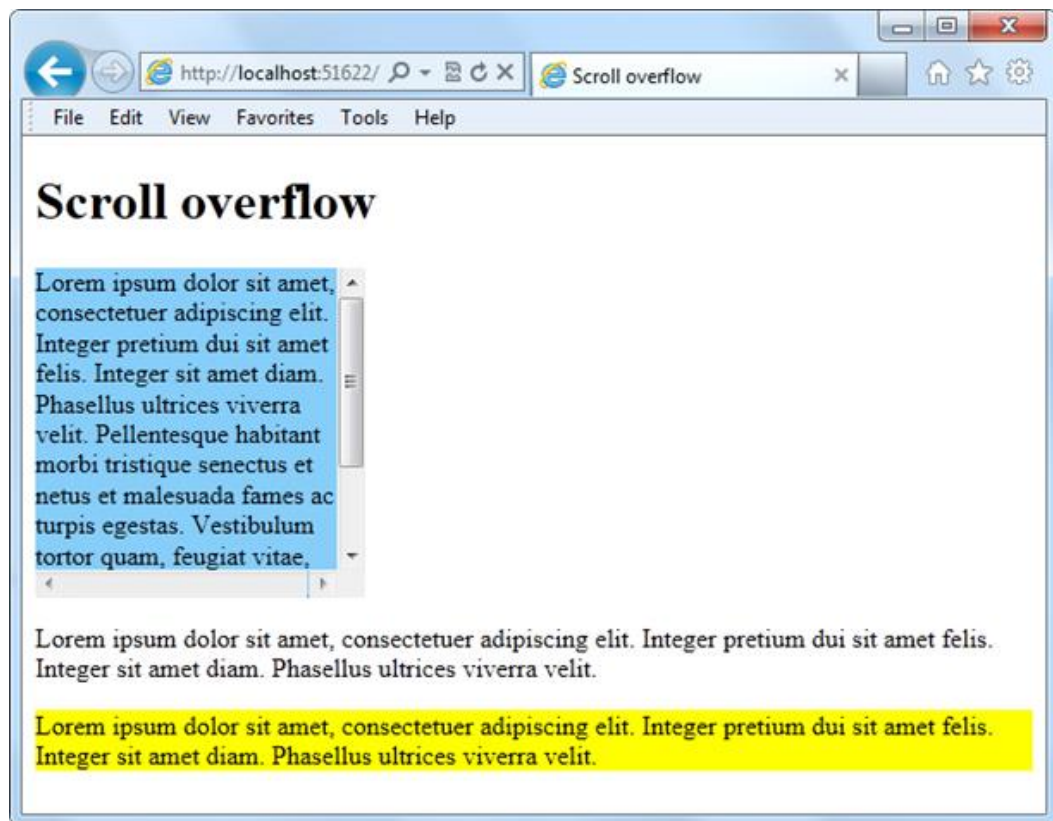
[https://www.w3schools.com/cssref/pr\\_pos\\_overflow.asp](https://www.w3schools.com/cssref/pr_pos_overflow.asp)

```

<!doctype html>
<html>
<head>
<title>Scroll overflow</title>
<style type = "text/css" >
#col1 {
width: 200px;
height: 200px;
background-color: lightskyblue;
overflow: scroll;
}
#col3 {
background-color: yellow;
}
</style>
</head>

<body>
<h1>Scroll overflow</h1>
<p id = "col1">Lorem ipsum . . .</p>
<p id = "col2">Lorem ipsum . . .</p>
<p id = "col3">Lorem ipsum . . .</p>
</body>
</html>

```

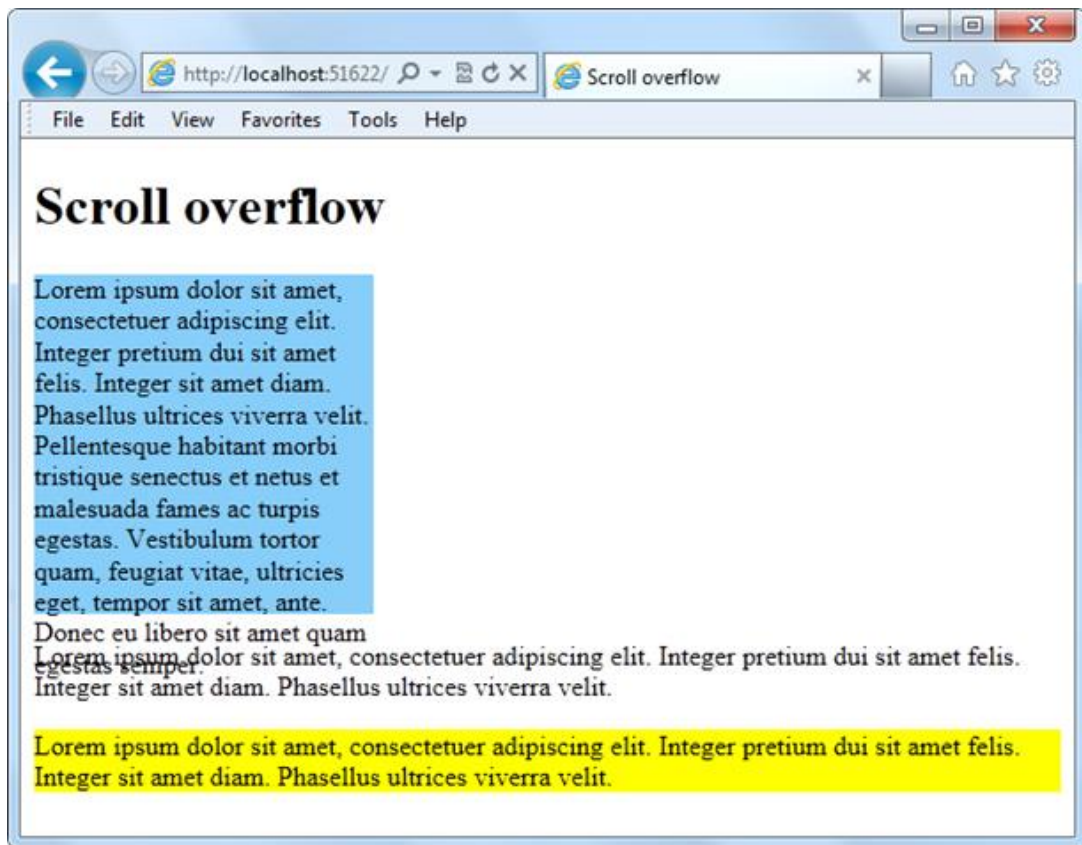


```

<!doctype html>
<html>
<head>
<title>Scroll overflow</title>
<style type = "text/css" >
#col1 {
width: 200px;
height: 200px;
background-color: lightskyblue;
overflow: visible;
}
#col3 {
background-color: yellow;
}
</style>
</head>

<body>
<h1>Scroll overflow</h1>
<p id = "col1">Lorem ipsum . . .</p>
<p id = "col2">Lorem ipsum . . .</p>
<p id = "col3">Lorem ipsum . . .</p>
</body>
</html>

```



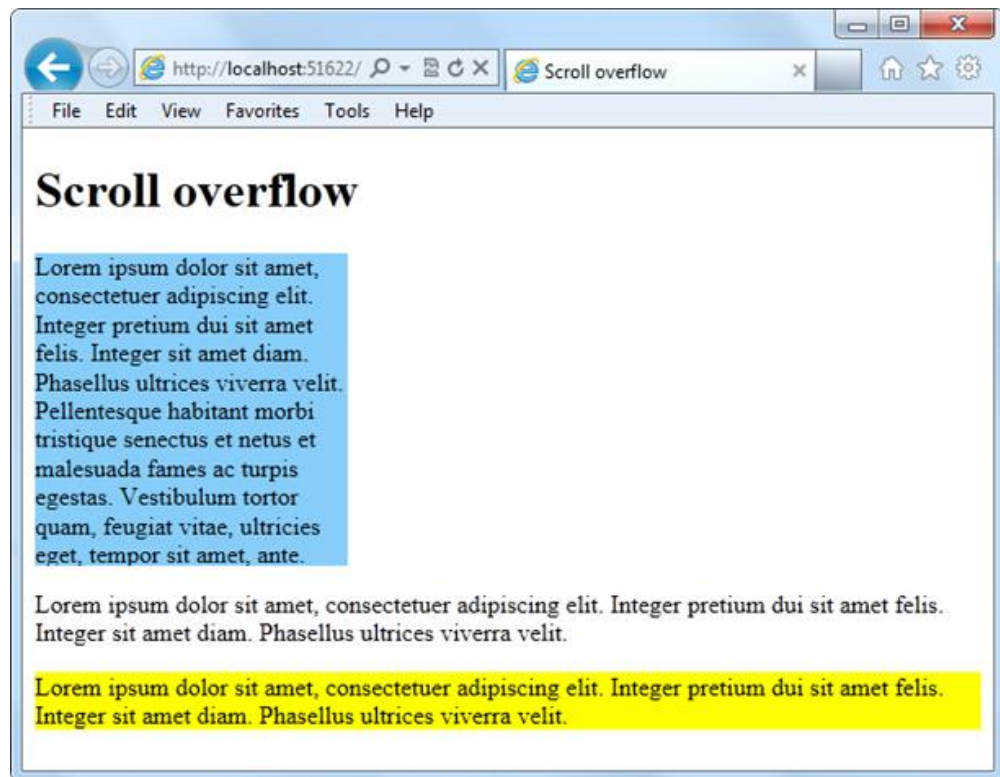


```

<!doctype html>
<html>
<head>
<title>Scroll overflow</title>
<style type = "text/css" >
]#col1 {
width: 200px;
height: 200px;
background-color: lightskyblue;
overflow: hidden;
}
]#col3 {
background-color: yellow;
}
</style>
</head>

<body>
<h1>Scroll overflow</h1>
<p id = "col1">Lorem ipsum . . .</p>
<p id = "col2">Lorem ipsum . . .</p>
<p id = "col3">Lorem ipsum . . .</p>
</body>
</html>

```



# display

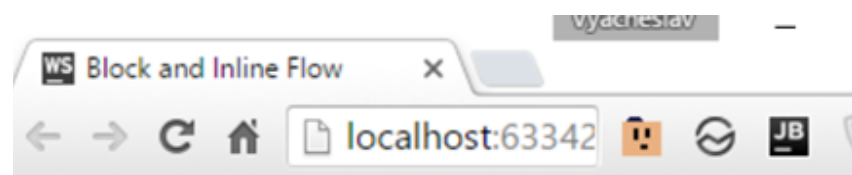
- указывает как рендерить элемент

[https://www.w3schools.com/cssref/pr\\_class\\_display.asp](https://www.w3schools.com/cssref/pr_class_display.asp)



display: block;

```
<!doctype html>
<html>
  <head>
    <title>Block and Inline Flow</title>
  </head>
  <body>
    <h1>Block and inline flow</h1>
    <p>Here are som options:</p>
    <ul class="toolbar">
      <li>Automobile</li>
      <li>Bicicle</li>
      <li>Scooter</li>
      <li>Taxi</li>
      <li>Walk</li>
    </ul>
  </body>
</html>
```



## Block and inline flow

Here are some options:

- Automobile
- Bicycle
- Scooter
- Taxi
- Walk

# display: inline-block;

```
<!doctype html>
<html>
  <head>
    <title>Block and Inline Flow</title>
    <style>
      .toolbar li {
        display: inline;
        background-color: #EEE;
        border: 1px solid;
        border-color: #F3F3F3 #BBB #BBB #F3F3F3;
        margin: 2px;
        padding: .5em;
      }
    </style>
  </head>
  <body>
    <h1>Block and inline flow</h1>
    <p>Here are some options:</p>
    <ul class="toolbar">
      <li>Automobile</li>
      <li>Bicicle</li>
      <li>Scooter</li>
      <li>Taxi</li>
      <li>Walk</li>
    </ul>
  </body>
</html>
```



# CSS стили браузера

- каждый браузер имеет стили по умолчанию для элементов
- эти стили различны для разных браузеров, поэтому одинаковая разметка может выглядеть иначе
- чтобы одинаковая разметка выглядела одинаково, рекомендуется использовать таблицу стилей "reset" или "normalize"

Reset CSS: <http://meyerweb.com/eric/tools/css/reset/>

Normalize CSS: <http://necolas.github.io/normalize.css/>

# Примеры реализации

calc

<https://codepen.io/timbrown/pen/akXvRw> <https://codepen.io/sebastianekstrom/pen/lmLHK> <https://codepen.io/ed1nh0/pen/puhsf>

float

<https://codepen.io/smashttheshell/pen/qOrwV> <https://codepen.io/sercaneraslan/pen/acofj>

position

<https://codepen.io/arlinadesign/pen/BNwKOR> <https://codepen.io/supermrji/pen/nsHKu>

# Ссылки

<https://webref.ru/layout/structural-pseudo-classes> структурные псевдоклассы

<https://webref.ru/layout/pseudo-class-focus> псевдокласс фокус

<https://webref.ru/layout/pseudo-class-hover> псевдокласс ховер

<https://webref.ru/layout/after-before> псевдоэлементы after i before

<https://webref.ru/course/float> float

<https://webref.ru/course/position> position

<https://webref.ru/course/box-model> блочная модель

<https://webref.ru/layout/magic-of-css> магия css

# Видео

<https://www.youtube.com/watch?v=5ibNZAmyDV0&t=2s> box model

[https://www.youtube.com/watch?v=\\_3v\\_5BZK\\_L0](https://www.youtube.com/watch?v=_3v_5BZK_L0) треугольники

<https://www.youtube.com/watch?v=wQhN0SdSZuY&t=2s> структура сайта

<https://www.youtube.com/watch?v=eZingxnTQ90> позиционирования

<https://www.youtube.com/watch?v=hHRQSEXqfE0> float

<https://www.youtube.com/watch?v=0WDxaY7DhnE> списки