

ОСНОВЫ CSS. Часть II



CSS

Елена Иванова
@liveldi90
Front-end developer
Artec3D

Содержание

- ➔ Блочные и строчные модели.
- ➔ Области просмотра страницы.
- ➔ Позиционирование элементов на странице.
- ➔ Переполнение.
- ➔ Стилизуем ссылки и списки.
- ➔ Генерируемое содержимое.
- ➔ Работа с фоном и рамками.

Блочные и строчные модели

Блочная модель CSS

Внешний отступ (**margin**)

Граница (**border**)

Внутренний отступ (**padding**)

Содержимое элемента
(текст, картинка и т.д.)
(**width** и **height**)

width

— задает ширину для элемента, em, %, px, rem

height

— задает высоту для элемента, em, %, px, rem

max-width, max-height, min-width, min-height

Работает только для блочных элементов!

```
div {  
  width: 10px;  
  height: 30px;  
}
```

`padding` — задает внутренний отступ у элемента

```
padding: 10px 0;
```

— Верх/низ(10px) + право/лево(0)

```
padding: 0 10px;
```

— Верх/низ(0) + право/лево(10px)

```
padding: 10px 9px 6px;
```

— Верх(10px) + право/лево(9px) + низ

```
padding: 10px 4px 9px 3px;
```

— Верх(10px) + право(4px) + низ(9px)
лево(3px)

```
padding-top: 5px;
```

— Верх(5px)

```
padding-right: 5px;
```

— Право(5px)

```
padding-bottom: 5px;
```

— Низ(5px)

```
padding-left: 5px;
```

— Лево(5px)

**Для строчных элементов работает только право и лево!
(фон распространяется во все стороны, налезая на контент)**

margin — задает внешний отступ у элемента

```
margin: 10px 0;
```

— Верх/низ(10px) + право/лево(0)

```
margin: 0 10px;
```

— Верх/низ(0) + право/лево(10px)

```
margin: 10px 9px 6px;
```

— Верх(10px) + право/лево(9px) + низ

```
margin: 10px 4px 9px 3px;
```

— Верх(10px) + право(4px) + низ(9px)
лево(3px)

```
margin-top: 5px;
```

— Верх(5px)

```
margin-right: 5px;
```

— Право(5px)

```
margin-bottom: 5px;
```

— Низ(5px)

```
margin-left: 5px;
```

— Лево(5px)

Для строчных элементов работает только право и лево!

auto

```
margin: 0 auto;
```

```
margin: 0 0 0 auto;
```

```
margin: 0 auto 0 0;
```

```
width: auto;
```

```
height: auto;
```

- Центрирует элемент, относительно его ширины.
- Отбрасывает элемент вправо.
- Отбрасывает элемент влево.
- Ширина по-умолчанию у элемента.
- Высота по-умолчанию у элемента.

Схлопывание вертикальных отступов



Практика: рекомендуется **margin** использовать для указания **боковых и нижних отступов**, для **верхнего padding**, либо **обходиться только нижним**.

Отрицательные отступы

padding — не работает с отрицательными отступами
margin — корректно работает -верх, -право, -лево



Манипулируем блочностью с помощью CSS

```
display: inline;  
display: block;
```

```
display: inline-block;
```

— Делаем из любого элемента **строчный**.

— Делаем из любого элемента **блочный**.

— Остается строчным, но на него действуют padding и margin, как у блочного. Можно задавать ширину и высоту.

Области просмотра страницы

Область просмотра (viewport)



Остальная доступная область



PageSpeed Insights

Поиск

ВОЙТИ

Адаптируйте размер контента для области просмотра

Содержание

- Общая информация
- Рекомендации

Это правило срабатывает, когда PageSpeed Insights обнаруживает, что содержимое страницы шире указанной области просмотра и пользователь вынужден выполнять горизонтальную прокрутку.

Общая информация

Пользователи ПК и мобильных устройств привыкли выполнять вертикальную, а не горизонтальную прокрутку веб-сайтов. Если для просмотра всего содержания необходимо прокрутить сайт по горизонтали или уменьшить масштаб, это вызывает неудобства.

При разработке сайта для мобильных устройств с [мета-тегом viewport](#) вы можете случайно расположить содержание так, что оно не поместится в указанную область просмотра. Например, если изображение шире области просмотра, может возникнуть необходимость в горизонтальной прокрутке. Во избежание этого нужно изменить контент так, чтобы он помещался целиком.

Рекомендации

Так как размер экрана различается на разных устройствах (например, на телефонах/планшетных ПК и даже на разных телефонах), необходимо [настроить область просмотра](#)

При отрицательных сдвигах элементов
вверх или влево, они скрываются за недоступной
частью. Скролла нет.

При сдвигах вниз и вправо. Может появиться
скролл.

Viewport

vw

— 100vw (ширина = 100 % ширине окна)

vh

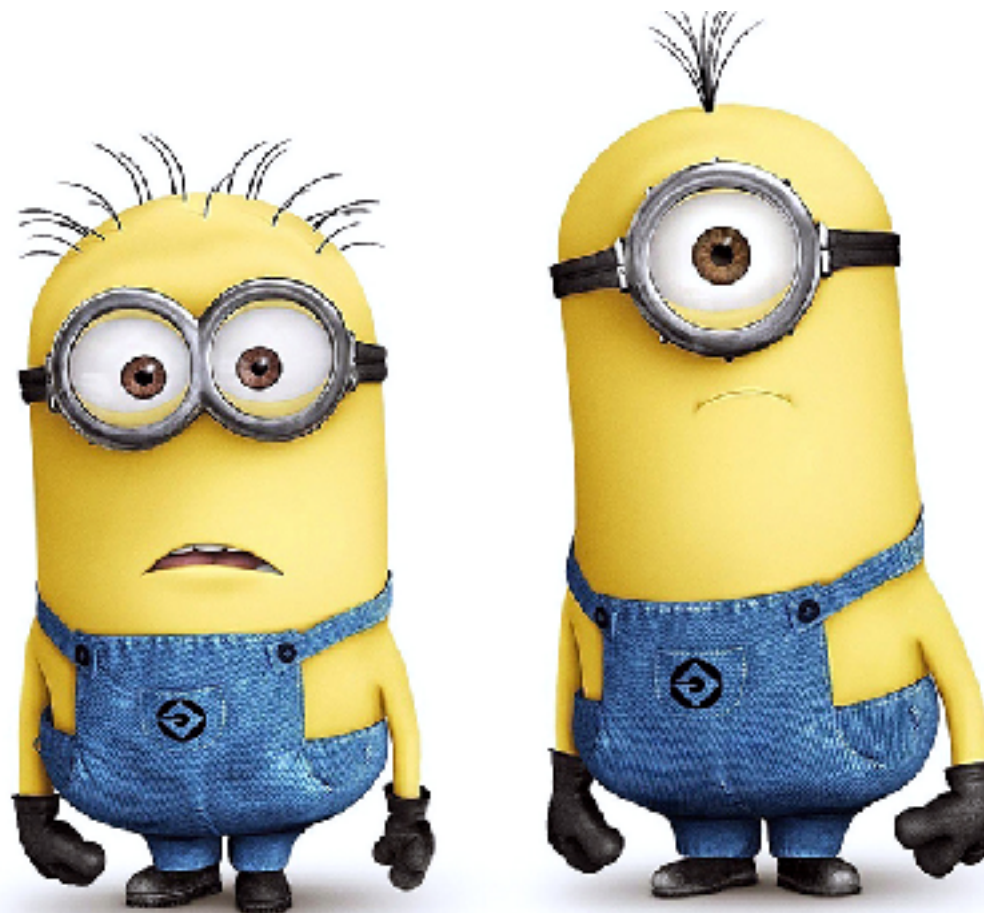
— 100vh (высота = 100 % высоте окна)

```
header {  
  width: 100vw;  
  height: 100vh;  
}
```

Позиционирование элементов на странице

Расположение элементов

В обычном потоке элементы располагаются друг под другом. Как нам это изменить, сделать колонки, фиксированные меню?



Позиционирование

position

position: static

Значение по умолчанию.
Ничего не делает...

position: relative

- Элемент остается на своем месте, но за счет свойств `top`, `left`, `right`, `bottom`, его можно сдвинуть относительно своего первоначального места.
- При этом элемент начнет налезать на соседей, а за собой оставит пустое пространство.

position: absolute

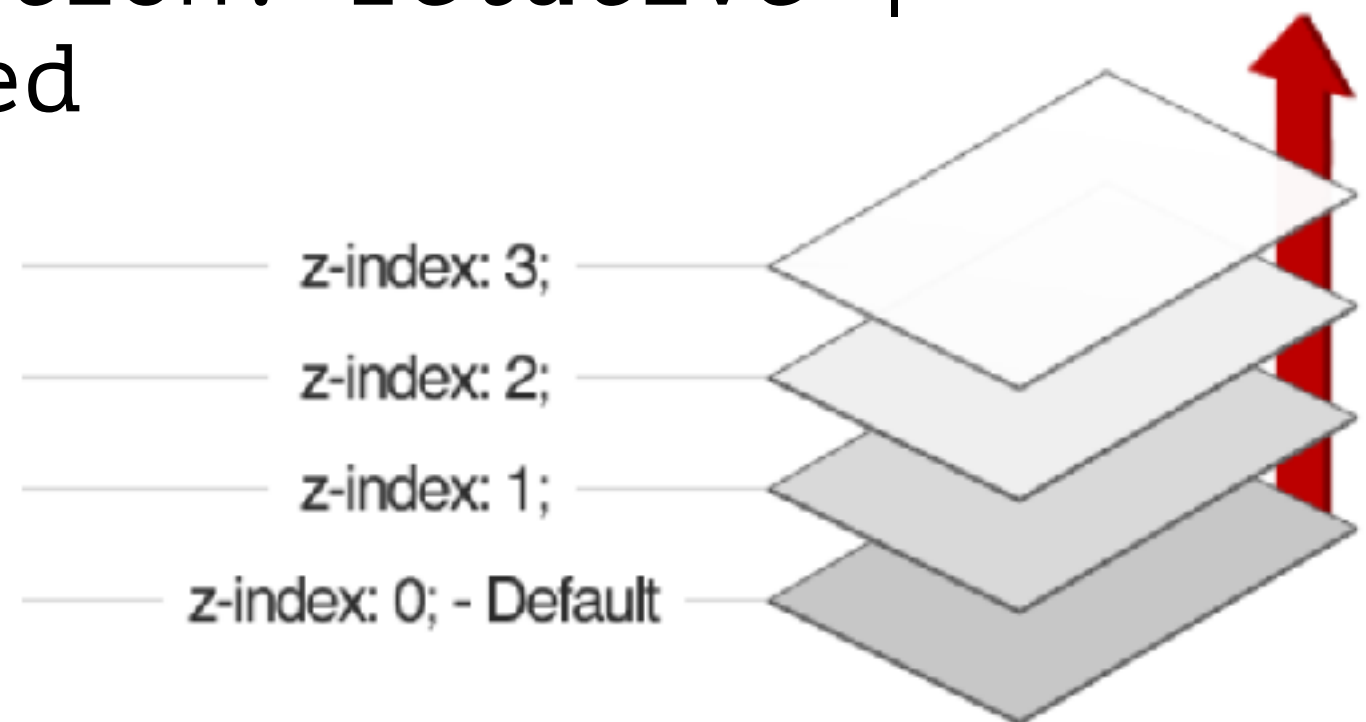
- Элемент полностью выходит из потока.
- `width` для блочных элементов становится по величине контента.
- Можно двигать с помощью `top`, `left`, `right`, `bottom`.
- Если среди родительских элементов, нет никакого со свойством `position: relative`, `absolute` или `fixed`, то `top`, `left`, `right`, `bottom` будут работать относительно окна браузера. Иначе относительно первого родителя.

position: fixed

- Элемент полностью выходит из потока.
- `width` для блочных элементов становится по величине контента.
- Можно двигать с помощью `top`, `left`, `right`, `bottom`. Действуют они всегда относительно окна.
- Элемент всегда виден, даже при скролле.

z-index

- Определяет видимость при наложении.
- Возможные значения положительное число, auto
- Работает для `position: relative | absolute | fixed`



Переполнение

Overflow

visible

— Отображается все содержание элемента (по умолчанию).

hidden

— Отображается только область внутри элемента, остальное скрывается

scroll

— Всегда добавляются полосы прокрутки.

auto

— Полосы прокрутки добавляются только при необходимости.

Стилизуем ссылки,
СПИСКИ

Псевдоклассы ссылок

a:link

— Не посещенная

a:visited

— Посещенная: по которой уже выполнялся переход

a:hover

— Не нажатая: над которой находится указатель мыши

a:active

— Нажатая: которая удерживается мышью

Подчеркивание ссылок

```
text-decoration: none;
```

— нет подчеркивания

```
text-decoration: underline;
```

— линия

```
border-bottom: 1px solid #000000;
```

— линия

```
border-bottom: 1px dashed #000000;
```

























— п р е р ы в и с т а я л и н и я

```
border-bottom: 1px dotted #000000;
```

— т о ч е ч н а я л и н и я

Стилизация курсора

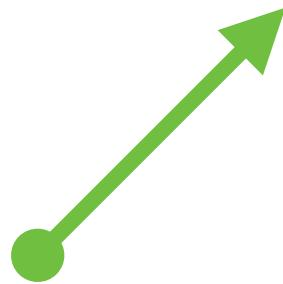
cursor

 auto	 move	 no-drop	 col-resize
 all-scroll	 pointer	 not-allowed	 row-resize
 crosshair	 progress	 e-resize	 ne-resize
 default	 text	 n-resize	 nw-resize
 help	 vertical-text	 s-resize	 se-resize
 inherit	 wait	 w-resize	 sw-resize

Списки

`list-style-type | list-style`

По умолчанию



disc <ul style="list-style-type: none">• item 1• item 2• item 3	circle <ul style="list-style-type: none">◦ item 1◦ item 2◦ item 3	square <ul style="list-style-type: none">■ item 1■ item 2■ item 3
decimal <ol style="list-style-type: none">1. item 12. item 23. item 3	decimal-leading-zero <ol style="list-style-type: none">01. item 102. item 203. item 3	upper-alpha <ol style="list-style-type: none">A. item 1B. item 2C. item 3
lower-alpha <ol style="list-style-type: none">a. item 1b. item 2c. item 3	upper-roman <ol style="list-style-type: none">I. item 1II. item 2III. item 3	lower-roman <ol style="list-style-type: none">i. item 1ii. item 2iii. item 3

`list-style: none`

Генерируемое
содержимое

:before, :after

С помощью псевдоклассов, можно сгенерировать различные дополнительные декораторы к элементам

:before

— Вставка перед элементом

:after

— Вставка после элемента

content: ‘ ’ — обязательное свойство



Любой стиль СПИСКОВ

list-style: none — убираем стиль по умолчанию

```
li {  
    padding-left: 17px;  
}  
  
li:before {  
    content: '-';  
    position: absolute;  
    margin-left: -18px;  
    margin-top: -2px;  
}
```

Уголки

```
a:before {  
  content: '';  
  display: block;  
  width: 0;  
  height: 0;  
  border-width: 15px 10px;  
  border-style: solid;  
  
  border-color: #000 transparent transparent transparent;  
  border-color: transparent #000 transparent transparent;  
  border-color: transparent transparent #000 transparent;  
  border-color: transparent transparent transparent #000;  
}
```



Работа с фоном и рамками

background

— ЦВЕТ

- background-color
- background-image
- background-repeat
- background-position
- background-attachment
- background-size
- background



background-color

— Определяет цвет фона элемента.

`background-color: <цвет> | transparent`

— Цвет задается в 16-ой системе счисления (`#000000`).

— В системе RGB(A) — red green blue (alpha)

`rgb(0, 0, 0)` и с прозрачностью 50% `rgba(0, 0, 0, 0.5)`

Таблица цветов: <http://www.stm.dp.ua/web-design/color-html.php>

background-image

— Устанавливает фоновое изображение для элемента.

`background-image: url(путь к файлу) | none`

```
body {  
  background-image: url('../images/pattern.jpg');  
}
```

background-repeat

— Определяет, как будет повторяться фоновое изображение.

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

```
body {  
  background-image: url('../images/pattern.jpg');  
  background-repeat: no-repeat;  
}
```

background-position

— Задаёт начальное положение фонового изображения.

`background-position:`

`[left | center | right | <проценты> | <значение>] ||`

`[top | center | bottom | <проценты> | <значение>]`

```
body {  
    background-image: url('../images/pattern.jpg');  
    background-position: center;  
    background-position: center center;  
    background-position: 50%;  
    background-position: right top;  
}
```


background-attachment

— Устанавливает, будет ли прокручиваться фоновое изображение вместе с содержимым элемента

`background-attachment: fixed | scroll`

```
body {  
  background-image: url('../images/pattern.jpg');  
  background-attachment: fixed;  
}
```

Пример:

http://feelingthelife.com/stories/16_dubai-br-whisper-of-the-sea

background-size

— Масштабирует фоновое изображение согласно заданным размерам.

`background-size: [<значение> | <проценты> | auto]{1,2} | cover | contain`

```
body {  
  background-image: url('../images/pattern.jpg');  
  background-size: cover;  
  background-size: 100% 100%;  
  background-size: auto 100px;  
}
```

background

— Универсальное свойство `background` позволяет установить одновременно несколько характеристик фона.

```
background: [  
    background-attachment ||  
    background-color ||  
    background-image ||  
    background-position ||  
    background-repeat ]
```

```
body {  
    background: fixed #000000 url('../images/1.jpg') center top no-repeat;  
}
```

border

— Граница

- border-width
- border-style
- border-color
- border



border-width

— Задаёт толщину границы элемента..

`border-width: [значение]`

```
ul {  
  border-width: 4px;  
}
```

border-style

1 пиксел	3 пиксела	5 пикселей	7 пикселей
dotted	dotted	dotted	dotted
dashed	dashed	dashed	dashed
solid	solid	solid	solid
double	double	double	double
groove	groove	groove	groove
ridge	ridge	ridge	ridge
inset	inset	inset	inset
outset	outset	outset	outset

border-color

— Устанавливает цвет границы.

`border-color: [цвет]`

```
ul {  
    border-color: #000000;  
}
```

border

— Универсальное свойство `border` позволяет одновременно установить толщину, стиль и цвет границы вокруг элемента.

`border: [border-width || border-style || border-color]`

```
ul {  
  border: 4px solid rgba(0, 0, 0, 0.5);  
  border-top: 4px solid rgba(0, 0, 0, 0.5);  
  border-right: 4px solid rgba(0, 0, 0, 0.5);  
  border-bottom: 4px solid rgba(0, 0, 0, 0.5);  
  border-left: 4px solid rgba(0, 0, 0, 0.5);  
}
```


Домашнее задание

Домашнее задание

1. Знакомство с CSS (пройти до конца).
2. Селекторы, часть 1
3. Наследование и каскадирование

Вопросы?



Автор курса: Елена Иванова
lessons4.liveldi.ru