

# Продвинутый CSS



Елена Иванова  
@liveldi90  
Front-end developer  
Artec3D

# Содержание

- ➔ Медиа запросы.
- ➔ Flexbox разметка.
- ➔ Градиенты, тени, прозрачность, скругления уголков.
- ➔ Фильтры.
- ➔ 2D трансформации.
- ➔ Анимация: CSS transition.

Медиа запросы

# meta viewport

Обычно браузеры смартфонов имитируют браузеры настольных компьютеров с высоким разрешением, поэтому страницы отображаются в них, как на экране монитора.

```
<head>  
  ...  
  <meta name="viewport" content="width=device-width, initial-scale=1" />  
  ...  
</head>
```

`width` — контролирует размер окна просмотра в зависимости от устройства.

`initial-scale` — уровень масштабирования, можно уменьшать или увеличивать.

# media queries

С помощью `media queries` мы можем отслеживать разрешение экрана пользователя и отображать необходимые стили для каждого разрешения или устройства.

```
@media screen and (max-width:667px) {...}
```

```
@media screen and (min-width:667px) {...}
```

Разрешения apple: <http://stephen.io/mediaqueries/>

# Адаптивная верстка



# CSS

```
.col {  
  width: 33%;  
  float: left;  
}  
  
@media screen and (max-width: 900px) {  
  .col {  
    width: 50%;  
  }  
}  
  
@media screen and (max-width: 667px) {  
  .col {  
    width: 100%;  
  }  
}
```

**Размеры переходов лучше определять опираясь на представление вашего сайта.**



МCS — кодинг для девушек

lessons31.liveldi.ru

iPhone 6 375 x 667 75%

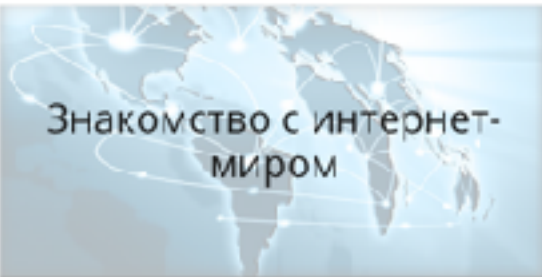
Responsive

- Galaxy S5
- Nexus 5X
- Nexus 6P
- iPhone 5
- ✓ iPhone 6
- iPhone 6 Plus
- iPad

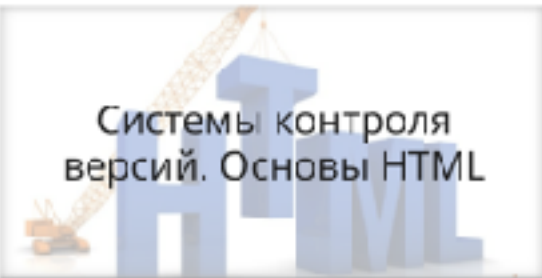
Edit...

# КОДИНГ ДЛЯ ДЕВУШЕК

октябрь 2016



Знакомство с интернет-миром



Системы контроля версий. Основы HTML

**Включение мобильного режима**

Elements Console Sources Network Timeline Profiles Application Security Audits Redux

```
<!DOCTYPE html>
<html lang="ru">
  <head>...</head>
  <body cz-shortcut-listen="true"> == $0
    <div class="container">...</div>
    <footer>...</footer>
    <script src="/app.js?999"></script>
  </body>
</html>
```

html body

Styles Computed Event Listeners

Filter :hov .cls +

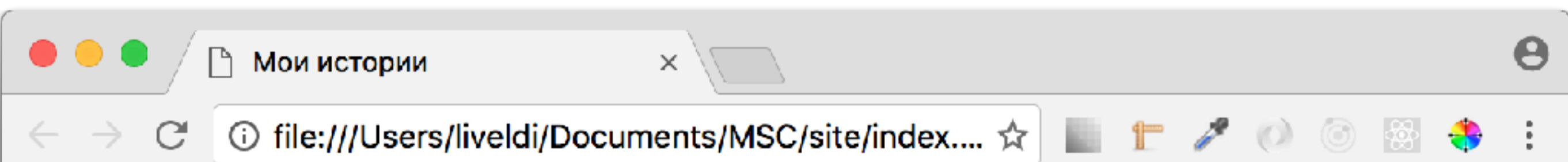
```
element.style {
}

body {
  font: 16px/1.2em 'Open Sans', Arial;
  min-height: 100vh;
}
```



Flexbox разметка

# Фиксированная и резиновая колонка?



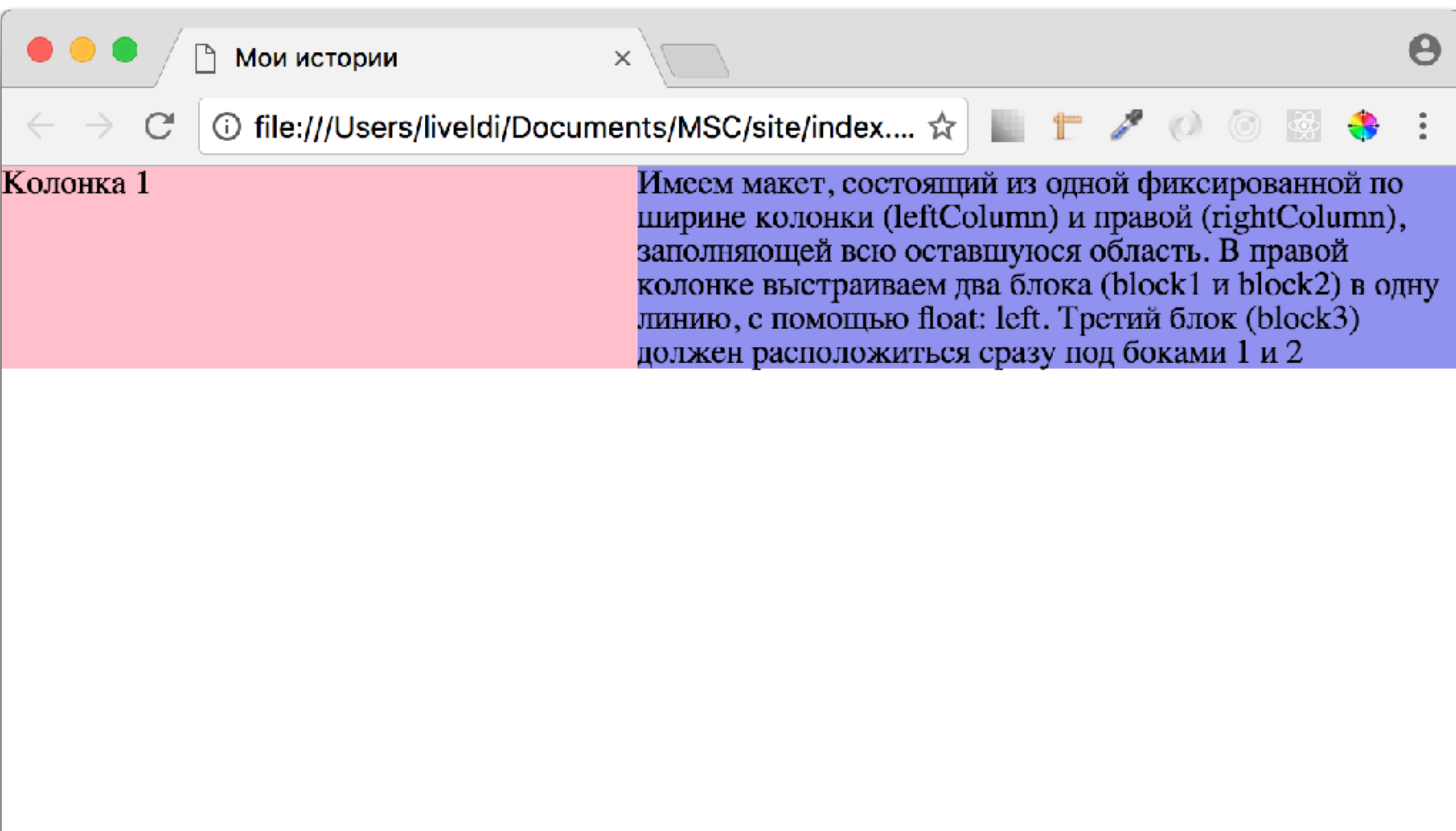
## Колонка 1

Имеем макет, состоящий из одной фиксированной по ширине колонки (`leftColumn`) и правой (`rightColumn`), заполняющей всю оставшуюся область. В правой колонке выстраиваем два блока (`block1` и `block2`) в одну линию, с помощью `float: left`. Третий блок (`block3`) должен расположиться сразу под блоками 1 и 2

```
.cols-1 {  
  background: pink;  
  width: 300px;  
  float: left;  
}
```

```
.cols-2 {  
  background: #9191f0;  
  margin-left: 300px;  
}
```

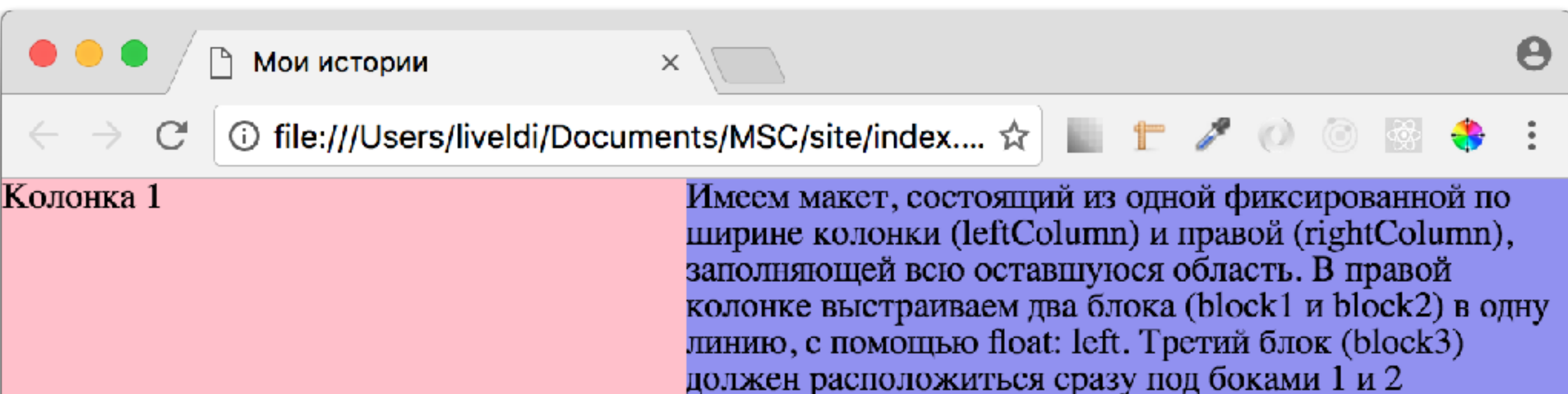
# Одинаковая высота колонок ?



The screenshot shows a web browser window with a single tab titled "Мои истории". The address bar displays the file path: `file:///Users/liveldi/Documents/MSD/site/index....`. Below the browser window, there is a table with two columns. The first column is labeled "Колонка 1" and has a pink background. The second column has a blue background and contains text describing a layout design.

Колонка 1	Имеем макет, состоящий из одной фиксированной по ширине колонки ( <code>leftColumn</code> ) и правой ( <code>rightColumn</code> ), заполняющей всю оставшуюся область. В правой колонке выстраиваем два блока ( <code>block1</code> и <code>block2</code> ) в одну линию, с помощью <code>float: left</code> . Третий блок ( <code>block3</code> ) должен расположиться сразу под боками 1 и 2
-----------	--

# Одинаковая высота колонок ?



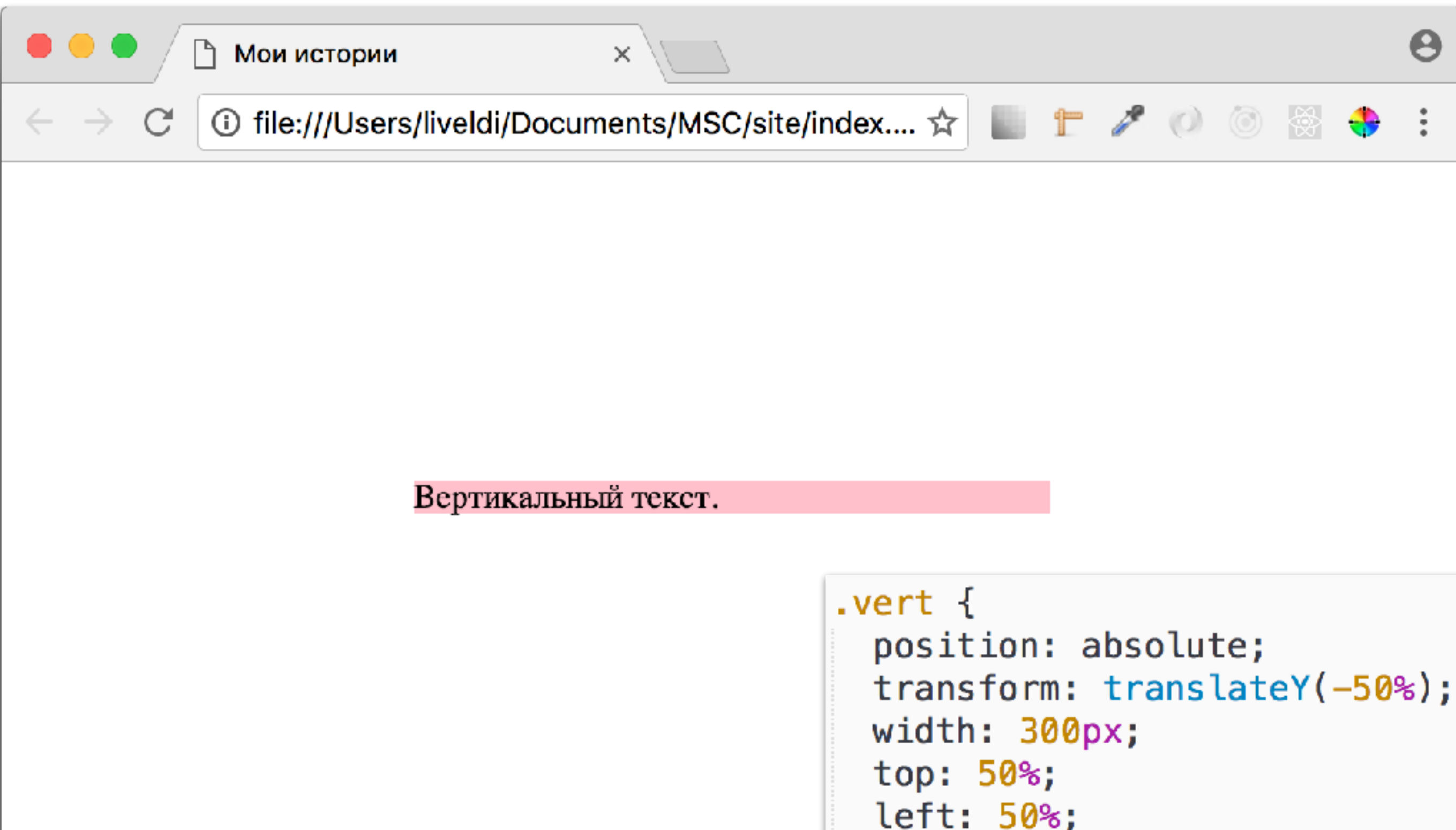
## HTML

```
<div class="cols clearfix">
  <div class="cols-1">
    ...
  </div>
  <div class="cols-2">
    ...
  </div>
</div>
```

## CSS

```
.cols {
  background: pink;
}
```

# Выравнивание по вертикали



Способы вертикального выравнивания

```
.vert {  
  position: absolute;  
  transform: translateY(-50%);  
  width: 300px;  
  top: 50%;  
  left: 50%;  
  margin-left: -150px;  
  background: pink;  
}
```

# Как решить все эти проблемы просто?



# Flexbox

Flexbox позволяет контролировать **размер, порядок, распределение свободного места между элементами** и многое другое.

Спецификация:  
[CSS Flexible Box Layout Module](#)

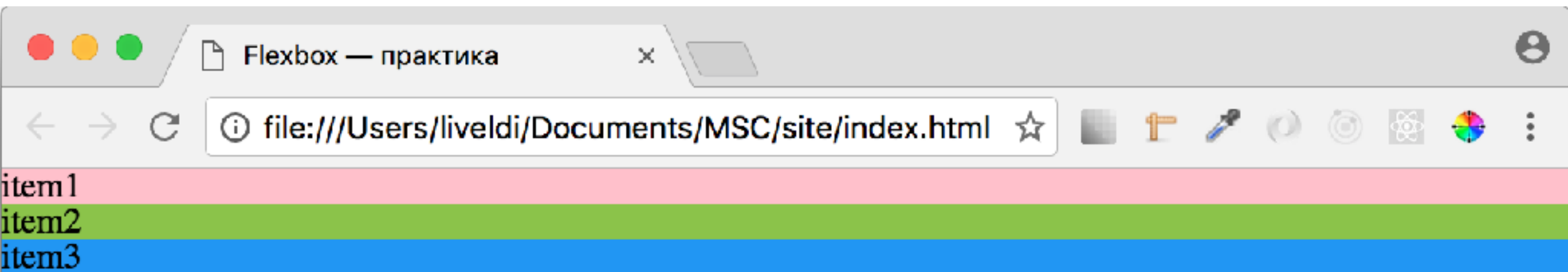


# Преимущества Flexbox

- Все блоки очень легко делаются “резиновым”.
- Легко выравнивать по вертикали и горизонтали.
- Можно менять расположение элементов с помощью CSS.
- Синтаксис CSS правил очень прост и осваивается довольно быстро.

# Создаем flex контейнер

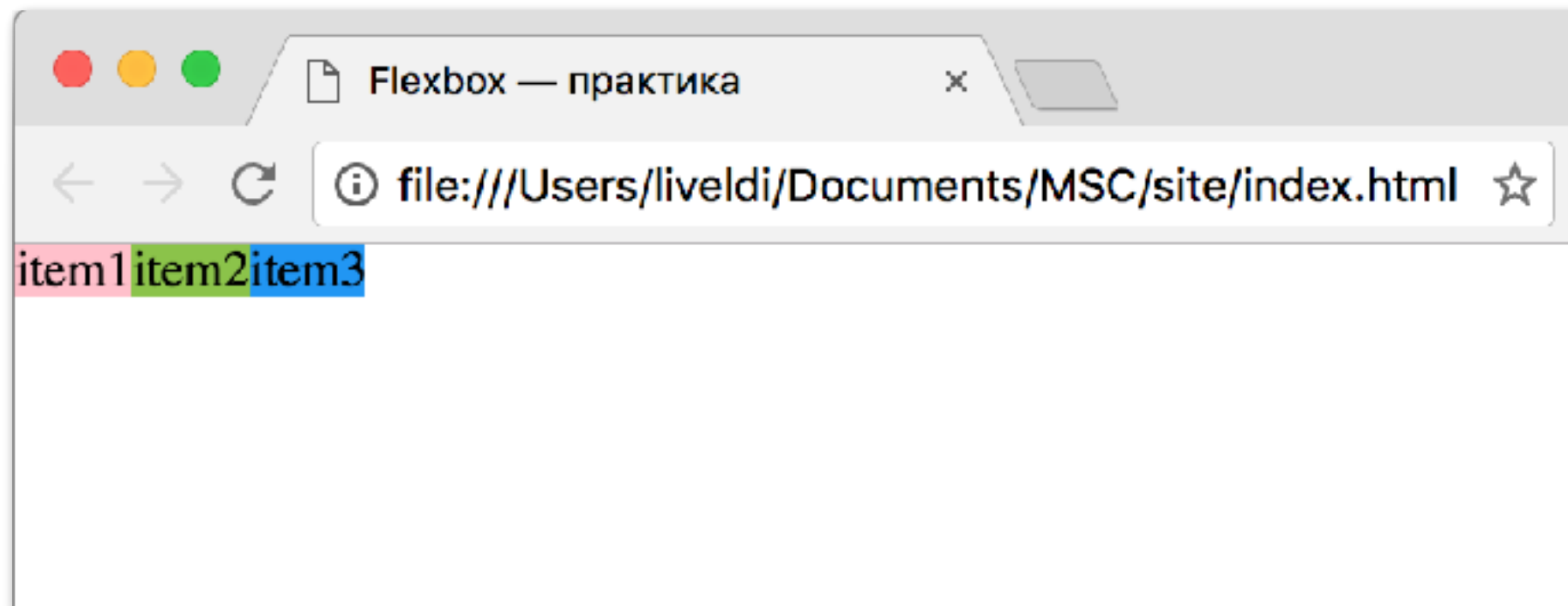
```
<div class="flex-container">  
  <div class="flex-item" style="background: pink;">item1</div>  
  <div class="flex-item" style="background: #8bc34a;">item1</div>  
  <div class="flex-item" style="background: #2196f3;">item1</div>  
</div>
```



# Подключаем Flexbox

Чтобы flex заработал, у основного контейнера, необходимо подключить `display: flex;`

```
.flex-container {  
  display: flex;  
}
```



# Основные свойства Flexbox контейнера

`flex-direction`

— направление главной оси

`justify-content`

— выравнивание по главной оси

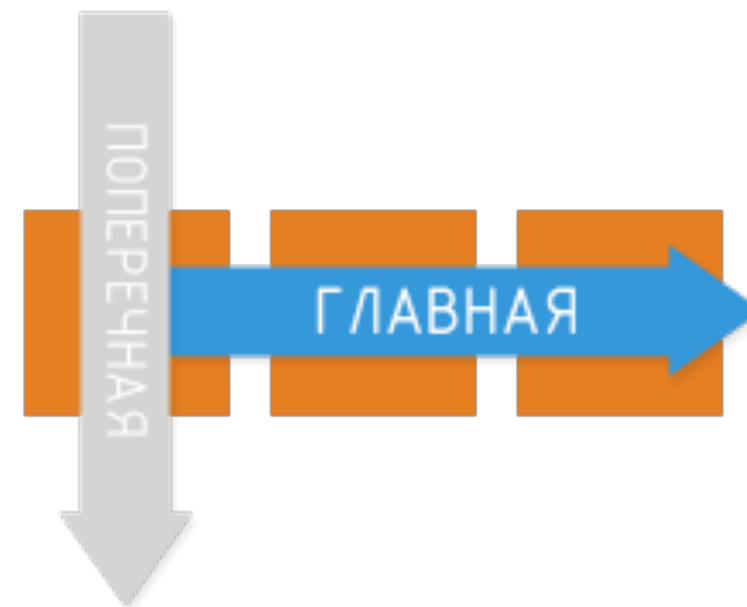
`align-items`

— выравнивание по поперечной оси.

# flex-direction



flex-direction: **column**



flex-direction: **row**

(по умолчанию)

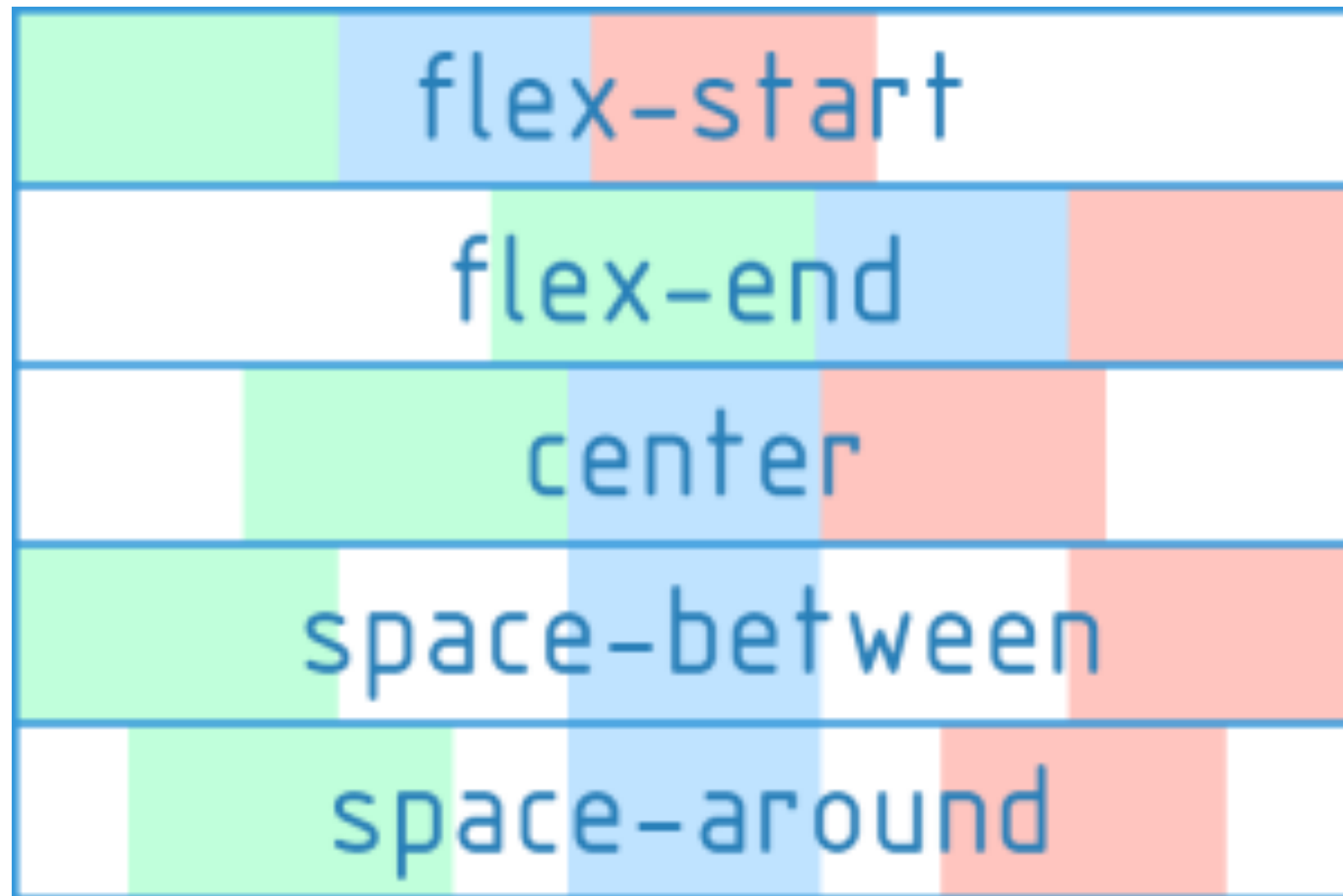
*Дополнительно — переопределяет направление.*

column-reverse

row-reverse

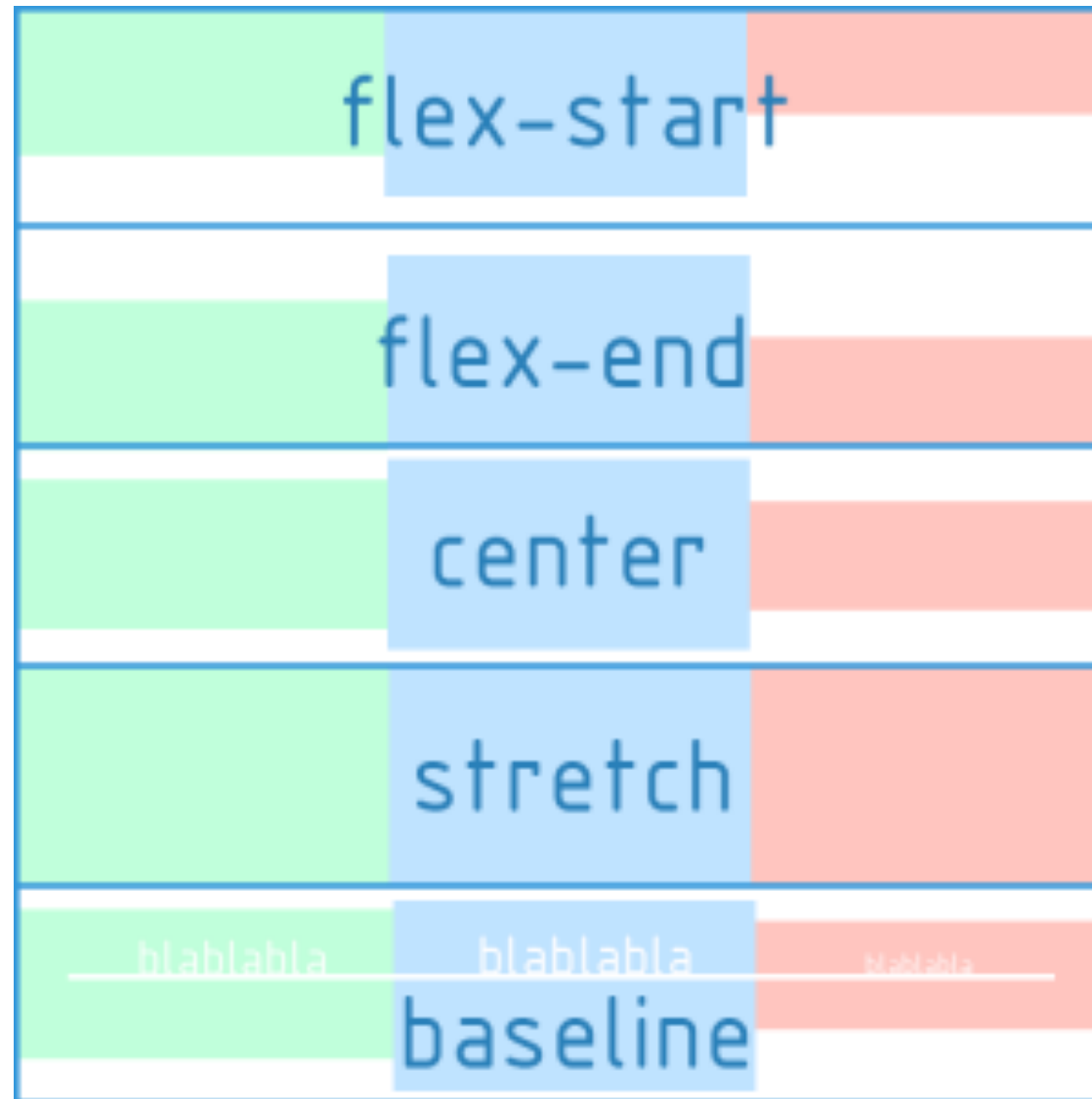
# justify-content

— Выравнивание по главной оси.



# align-items

— Выравнивание по поперечной оси



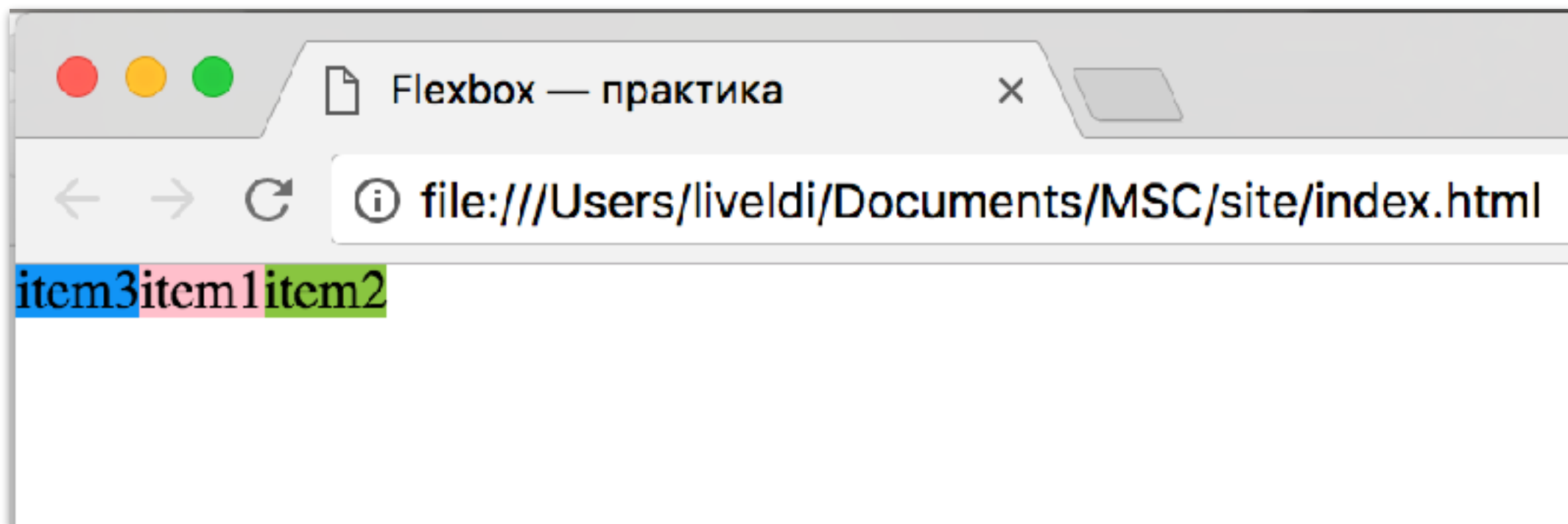
Статья по flexbox



# order

— Порядок следования отдельно взятого flex-блока внутри flex-контейнера.

```
<div class="flex-container">  
  <div class="flex-item" style="background: pink; order: 2;">item1</div>  
  <div class="flex-item" style="background: #8bc34a; order: 2;">item1</div>  
  <div class="flex-item" style="background: #2196f3; order: 1;">item1</div>  
</div>
```



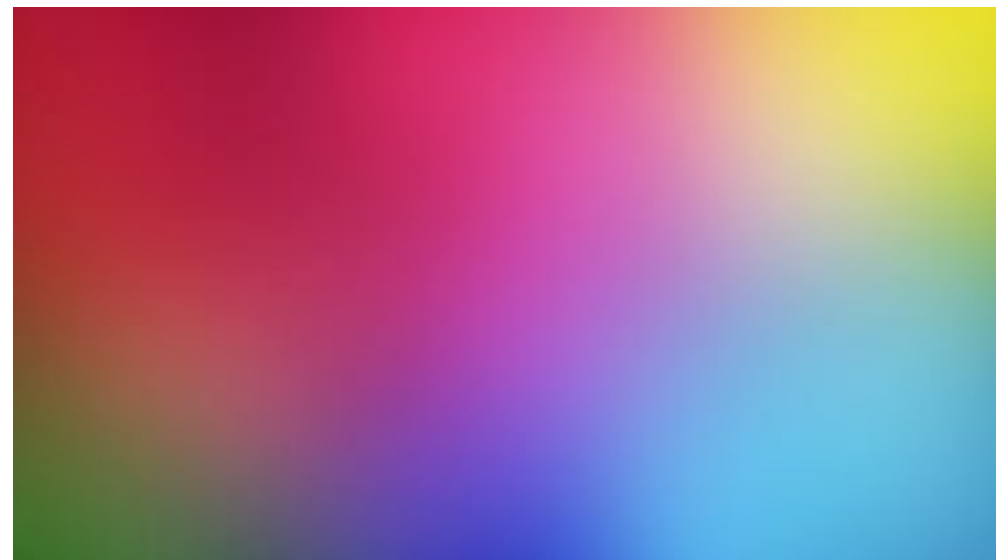
[Статья по flexbox](#)

Градиенты, тени,  
прозрачность,  
скругления уголков

# Градиент

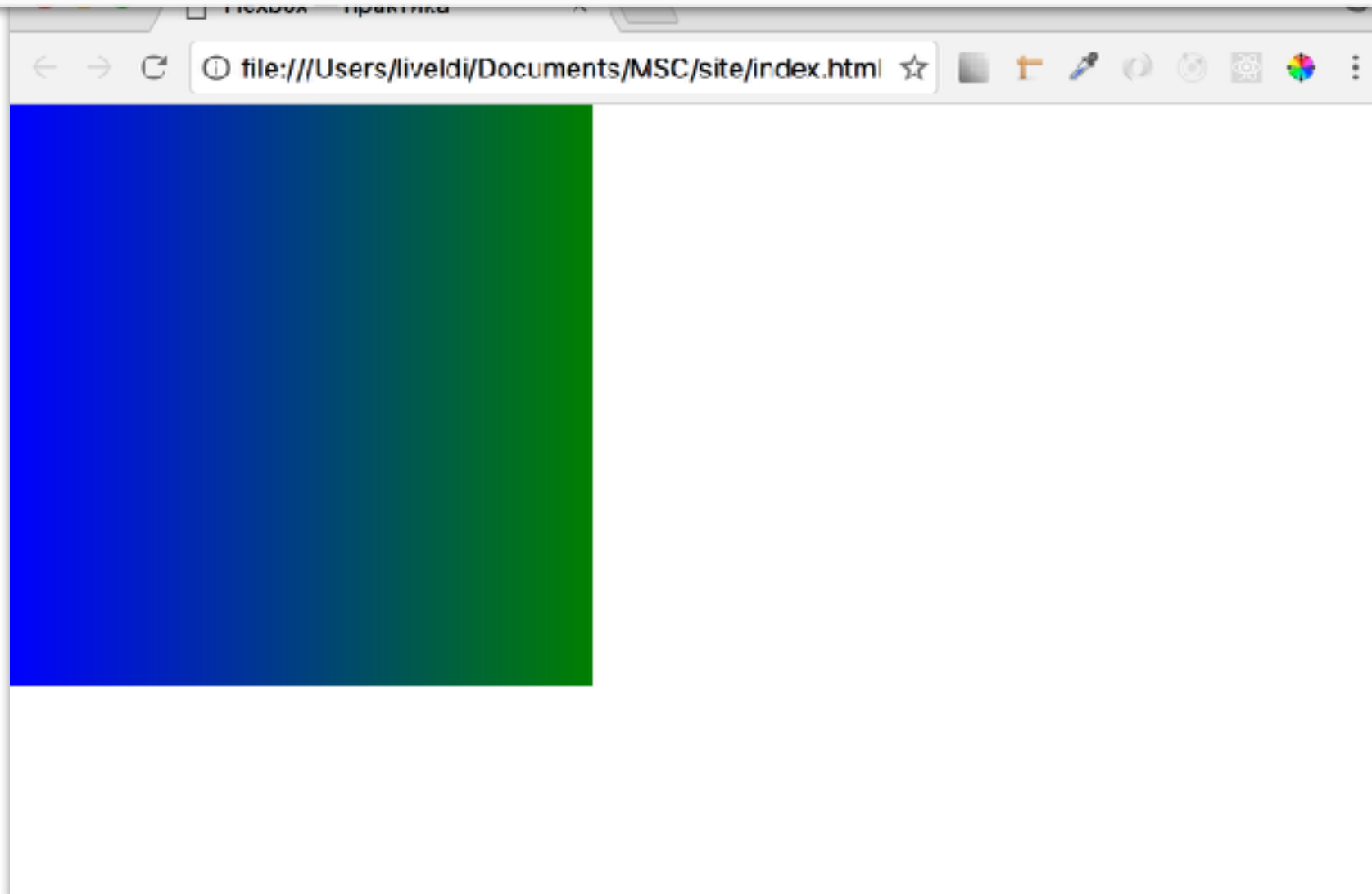
— представляет собой переход от одного цвета к другому.

- Линейный градиент: `linear-gradient()`
- Радиальный градиент: `radial-gradient()`



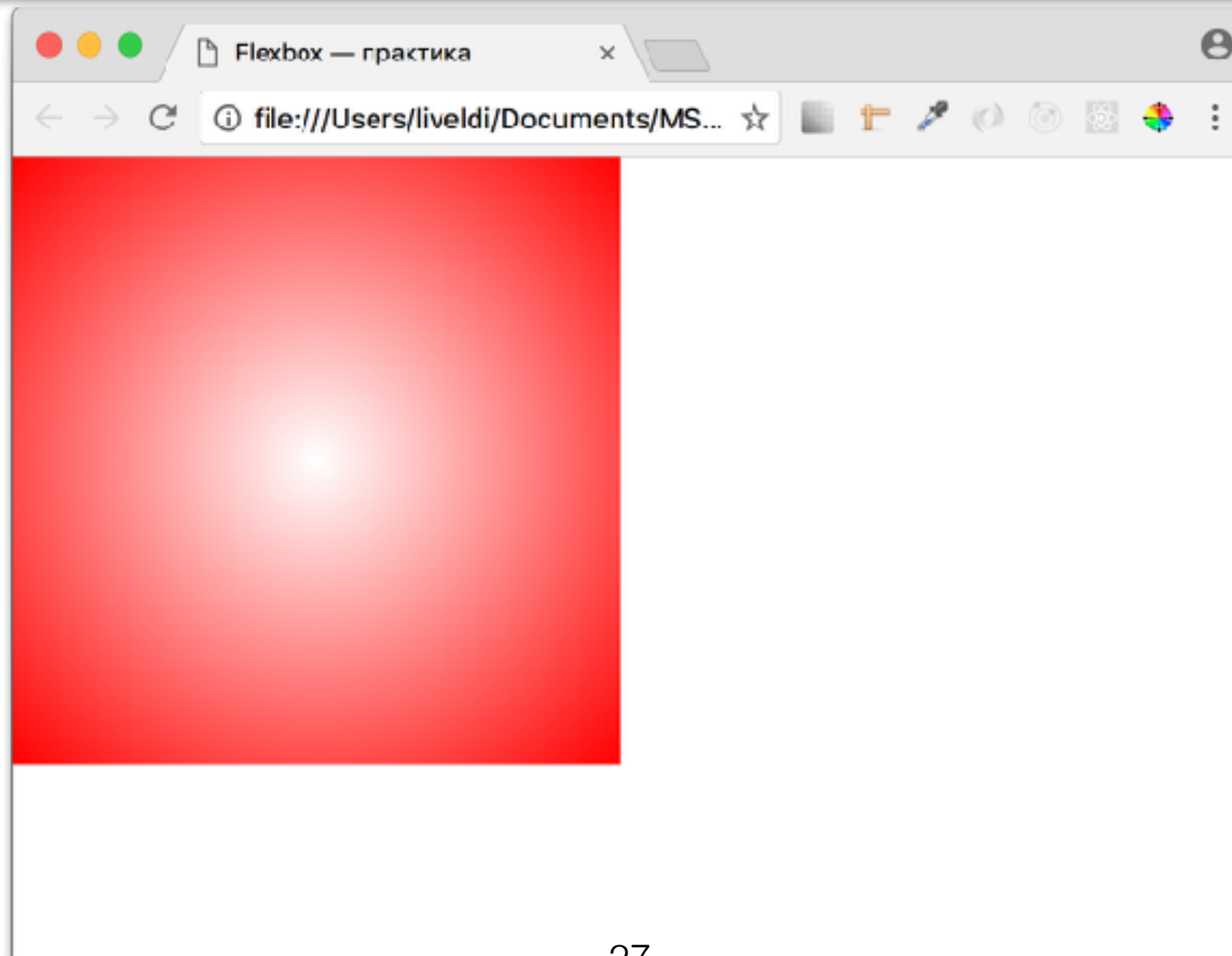
# Линейный градиент

```
.container {  
  width: 100px;  
  height: 300px;  
  background: linear-gradient(90deg, blue, green);  
}
```



# Радиальный градиент

```
.container {  
  width: 100px;  
  height: 300px;  
  background: radial-gradient(white, red);  
}
```



# Разные способы задания градиентов

Статья: CSS3 градиент

**Готовые решения: Ultimate CSS Gradient Generator**

# Тени

`box-shadow` — добавляет тень к элементу.

`box-shadow: [сдвиг по x] [сдвиг по y]  
[размытие] [растяжение] [цвет]`

```
.container {  
  width: 300px;  
  height: 300px;  
  box-shadow: 10px 10px 3px #000;  
}
```

**Готовые решения: *The ultimate CSS tools***



# Прозрачность

`opacity` — Определяет уровень прозрачности элемента веб-страницы.

`opacity:` [от 0 до 1]

0 — полностью прозрачное.

0.5 — 50% прозрачности.

1 — 100% прозрачности.

```
.container {  
  width: 300px;  
  height: 300px;  
  background: green;  
  opacity: 0.3;  
}
```

# Скругления углов

`border-radius` — устанавливает радиус скругления уголков рамки.

```
.container {  
  width: 50px;  
  height: 50px;  
  
  border: 50%;  
  
  border-radius: 3px;  
  
  border-radius: 50% 0;  
  
  border-radius: 0 50%;  
  
  border-radius: 0 4px 15px 50%;  
}
```



# Фильтры

# Фильтры

— воспроизводят в браузере визуальные эффекты, похожие на фильтры Photoshop



# Фильтры

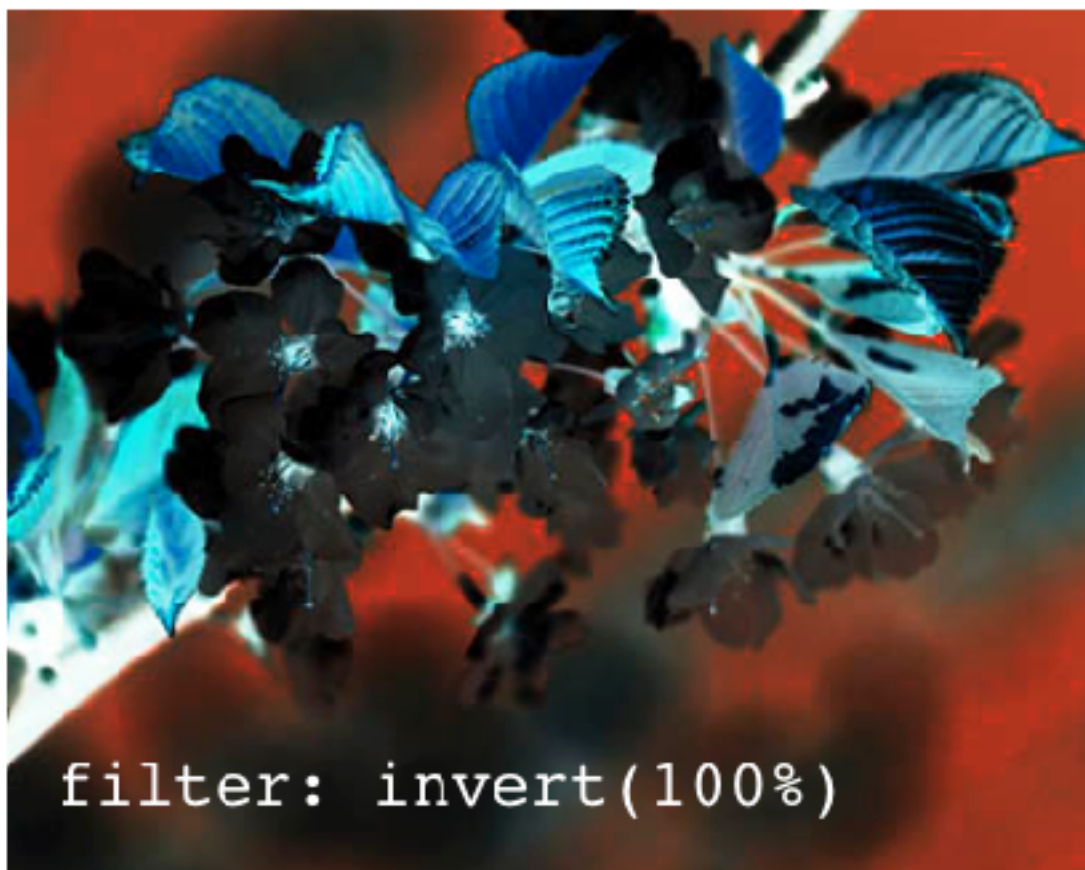


`filter: blur(3px)`



`filter: brightness(40%)`

# Фильтры



filter: invert(100%)



filter: contrast(10%)



# Фильтры



filter: grayscale(.75)



filter: saturate(300%)



# Фильтры



filter: hue-rotate(270deg)



filter: sepia(100%)

# Фильтры



`filter: opacity(50%)`

Статья: [CSS3 фильтры](#)

# 2D трансформации

# Трансформации

— изменяют размер, форму и положение элемента на веб-странице с помощью свойства `transform`

`translate`

— Сдвигает элемент на новое место.

`scale`

— Масштабирует элемент

`rotate`

— Поворачивает элемент

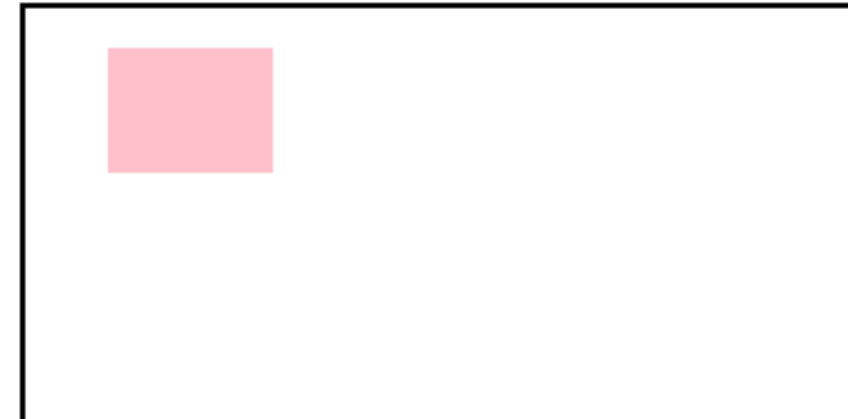
`skew`

— Деформирует стороны элемента

# translate

— Сдвигает элемент на новое место.

`transform: translate(x,y)`



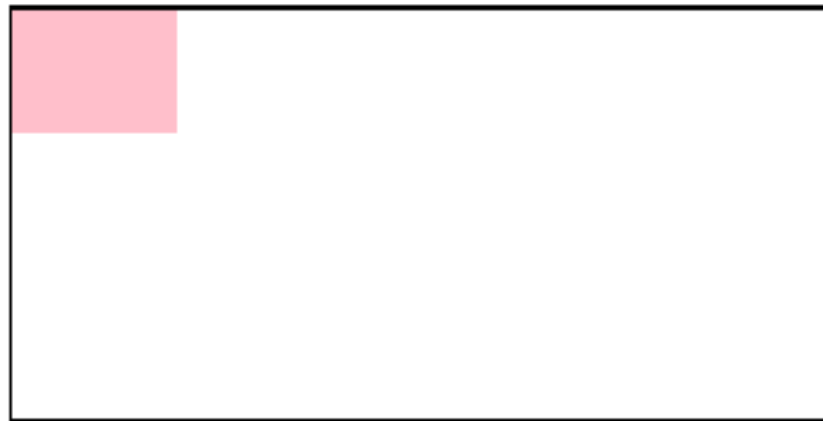
```
.inner {  
  width: 40px;  
  height: 30px;  
  background: pink;  
}
```

```
.inner {  
  width: 40px;  
  height: 30px;  
  background: pink;  
  transform: translate(20px, 10px);  
}
```

# scale

— Функция масштабирует элемент по ширине, делая его шире или уже.

`transform: scale(x,y)`



```
.inner {  
  width: 40px;  
  height: 30px;  
  background: pink;  
}
```

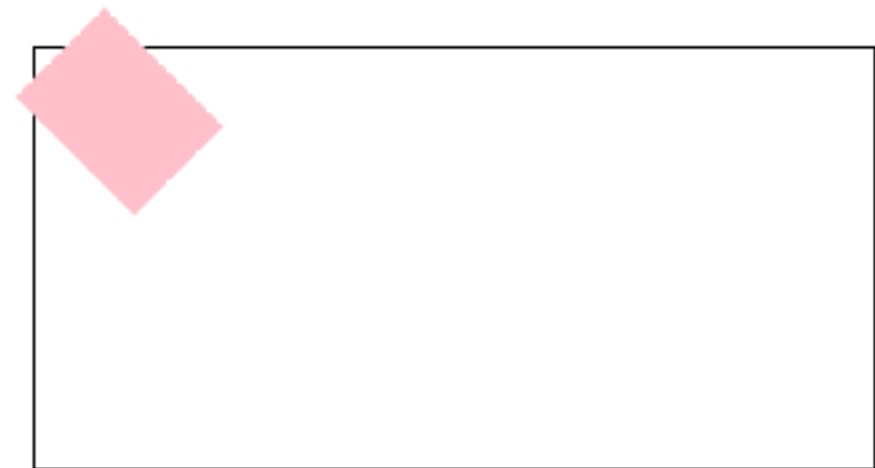
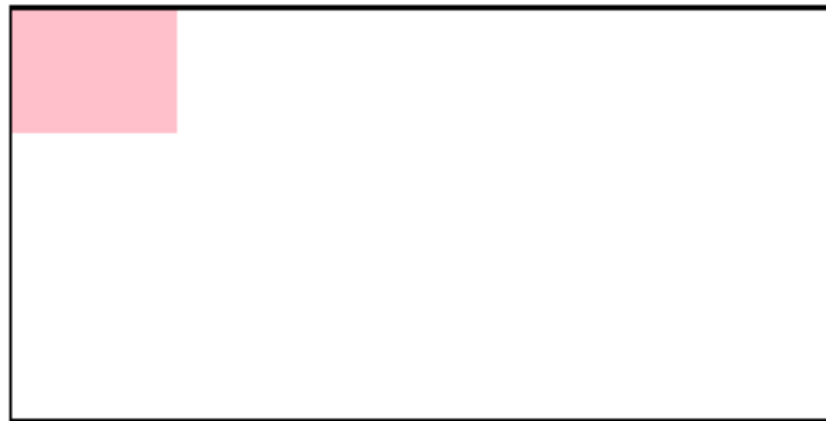


```
.inner {  
  width: 40px;  
  height: 30px;  
  background: pink;  
  transform: scale(2);  
}
```

# rotate

— Поворачивает элементы на заданное количество градусов.

`transform: rotate(x,y)`



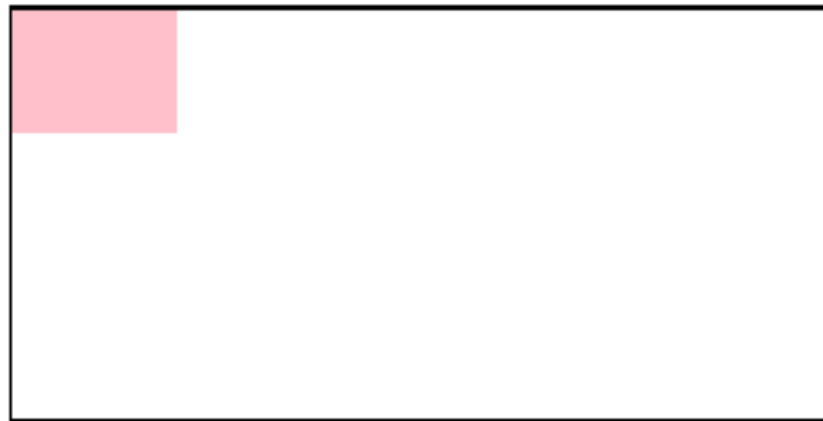
```
.inner {  
  width: 40px;  
  height: 30px;  
  background: pink;  
}
```

```
.inner {  
  width: 40px;  
  height: 30px;  
  background: pink;  
  transform: rotate(45deg);  
}
```

# skew

— Используется для деформирования сторон элемента относительно координатных осей.

`transform: skew(x, y)`



```
.inner {  
  width: 40px;  
  height: 30px;  
  background: pink;  
}
```

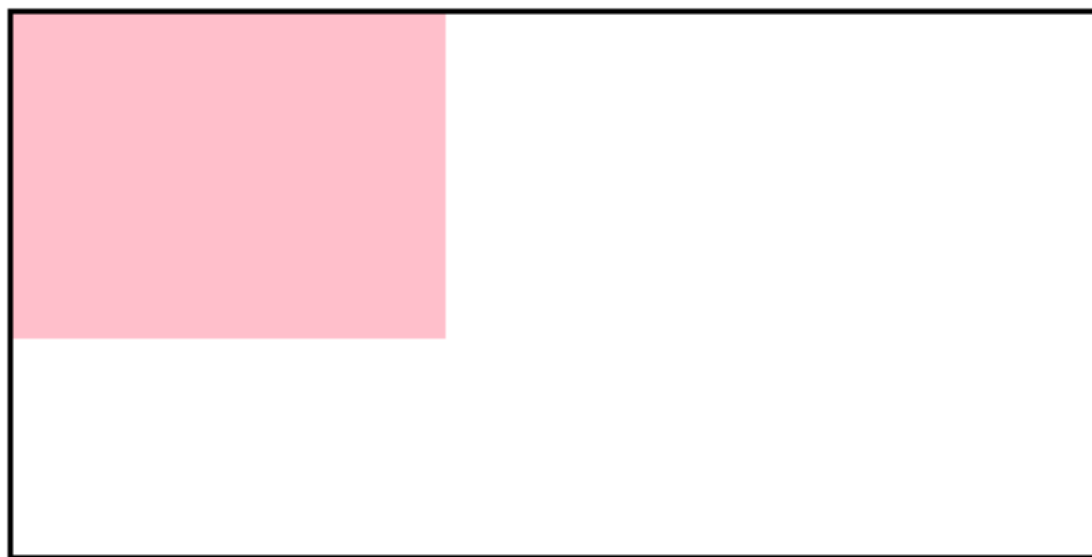
```
.inner {  
  width: 40px;  
  height: 30px;  
  background: pink;  
  transform: skew(-20deg, 0);  
}
```



# transform-origin

— Позволяет сместить центр трансформации, относительно которого происходит изменение положения/размера/формы элемента.

`transform-origin: [0%-100%, 0%-100%];`



```
.inner {  
  width: 40px;  
  height: 30px;  
  background: pink;  
  transform-origin: 0 0;  
  transform: scale(2);  
}
```

Статья: [2D-трансформации](#)

**Вместо % можно использовать любое значение величины.**

# Анимация: CSS transition

# CSS transition

Мы указываем, что некоторое свойство будет анимироваться при помощи специальных CSS-правил. Далее, при изменении этого свойства, браузер сам обрабатывает анимацию.

`transition-property` — Анимлируемое свойство.

`transition-duration` — Количество времени.

`transition-timing-function` — Тип анимации.

`transition` — Общее свойство.

# transition-property

— Любое CSS СВОЙСТВО, либо all для всех свойств.

```
transform-property: background-color;
```

```
transform-property: box-shadow;
```

```
transform-property: all;
```

# transition-duration

— Продолжительность анимации.

`transform-duration: 0.3s;`

`transform-duration: 2s;`

# transition-timing-function

— Тип анимации.

`linear` — линейная

`ease` — замедление в конце (по-умолчанию)

`ease-in` — замедление в начале и далее линейно

`ease-out` — замедление в конце и далее линейно

`ease-in-out` — замедление и в начале, и в конце. В середине линейно

# transition

— Общее свойство.

```
transform: background 0.3s ease-out;
```



# Transition на практике

```
.container {  
  margin: 40px;  
  width: 200px;  
  height: 100px;  
  background: red;  
  transition: background 0.3s;  
}  
  
.container:hover {  
  background: green;  
}
```

Примеры: CSS3 переходы



**Домашнее задание**

# Домашнее задание

1. Знакомство с CSS (пройти до конца).
2. Селекторы, часть 1
3. Наследование и каскадирование.
4. **Стилизовать свой сайт. Сбросить стили с помощью `reset.css`, подключить шрифт и использовать возможности `css`, чтобы сделать ваш сайт красивым!**



# Вопросы?



Автор курса: Елена Иванова  
[lessons7.liveldi.ru](https://lessons7.liveldi.ru)