

# HyperText Markup Language

## HTML & CSS

# Урок 4

## Графика в web-дизайне

# Оглавление

<b>Графика в web-дизайне .....</b>	<b>4</b>
Изображения на странице. Тег <img> .....	4
Атрибуты тега <img> .....	10
CSS-свойства для изображений .....	13
Форматы графических файлов в Web .....	17
Верстка статьи с изображениями .....	21
Теги figure и figcaption. ....	34
Свойства группы background .....	35
Свойство background-position .....	45
Свойство background-size .....	47
Свойство background-attachment .....	49
Свойство background-origin .....	50
Свойство background-clip .....	51
Обобщенное свойство background .....	52
Множественные фоны. ....	56

Верстка PSD макета.....	59
Экспорт изображений .....	61
Копирование текста .....	64
Измерение отступов .....	64
Разметка шапки сайта.....	66
Раздел Services.....	68
Раздел About Us .....	70
Проверка верстки на соответствие макету .....	72
Домашнее задание .....	76

Материалы урока прикреплены к данному PDF-файлу. Для доступа к материалам, урок необходимо открыть в программе [Adobe Acrobat Reader](#).

# Графика в web-дизайне

## Изображения на странице. Тег <img>

Графика, а именно изображения, добавленные на страницу, очень украшают внешний вид сайта. Сейчас без них очень сложно представить web-страницу. Наверняка на предыдущих уроках вам не хватало изображений в примерах и домашних заданиях. Исправим эту ситуацию.

Изображения в html-разметку добавляются с помощью тега <img>. У этого тега есть 2 обязательных атрибута — src и alt, которые вы должны указывать всегда. Синтаксис добавления изображения на страницу таков:

```

```

Например,

```

```

В атрибуте src указывается источник изображения, т.е. месторасположение файла.

Как правило, все изображения для сайта складывают в папку с традиционным именем *images* или *img*, или реже — *assets* для того, чтобы структура сайта была упорядоченной. Мы тоже будем придерживаться этого правила. В атрибуте src указывается относительный путь до изображения. В данном случае термин «относительный путь» обозначает расположения файла изображения относительно html-файла. Папка *images* лежит в том же каталоге, что и файл

*image.html*. Поэтому путь к файлу изображения будет включать название папки/название файла и его расширение.

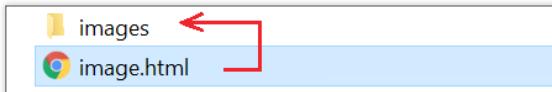


Рисунок 1

Как правило, редакторы кода подскажут вам название картинки после указания имени папки с изображениями в кавычках атрибута src (рис. 2-3):

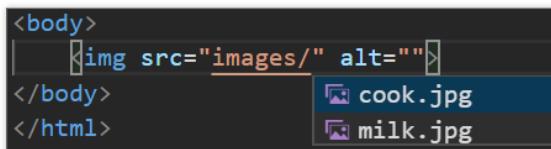


Рисунок 2. Visual Studio Code

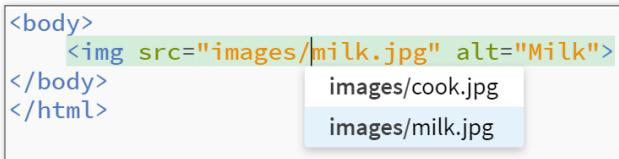


Рисунок 3. Brackets

Вы можете использовать клавишу Enter для подтверждения выбора, а также клавиши стрелок вниз (↓) или вверх (↑) для перехода к следующему/предыдущему файлу в папке. Ctrl+пробел в некоторых редакторах позволяет вывести список файлов в папке.

Атрибут alt необходим для указания альтернативного текста, который появится, если путь к изображению указан неверно, файл картинки имеет неподдерживаемый формат или был удален с сервера. Этот атрибут имеет

важное значение для поисковых роботов и программ для чтения с экрана, которые помогают людям с ограниченным зрением ориентироваться на сайте, поэтому является обязательным. Также нужно указывать в нем информацию, соответствующую тому, что изображено на картинке.

На рисунке 4 видно, что использовано несуществующее в папке *images* изображение *milk1.jpg*, поэтому в браузере показан текст из атрибута alt.

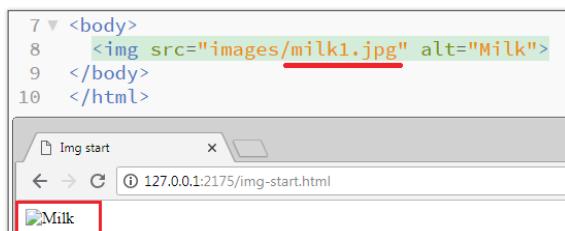


Рисунок 4

### Image alignment

Лorem ipsum dolor sit amet, consectetur adipisicing elit. Provident, nulla numquam sequi praesentium repellend iusto, sapiente velit incidunt nemo repudiandae tempora! Et error, hic! Doloremque.



Totam consequatur eveniet illum commodi pariatur vol  
dolorum omnis, laboriosam facere deleniti distinctio eaque quia. Saepe maxime, dignissimos repudiandae quam

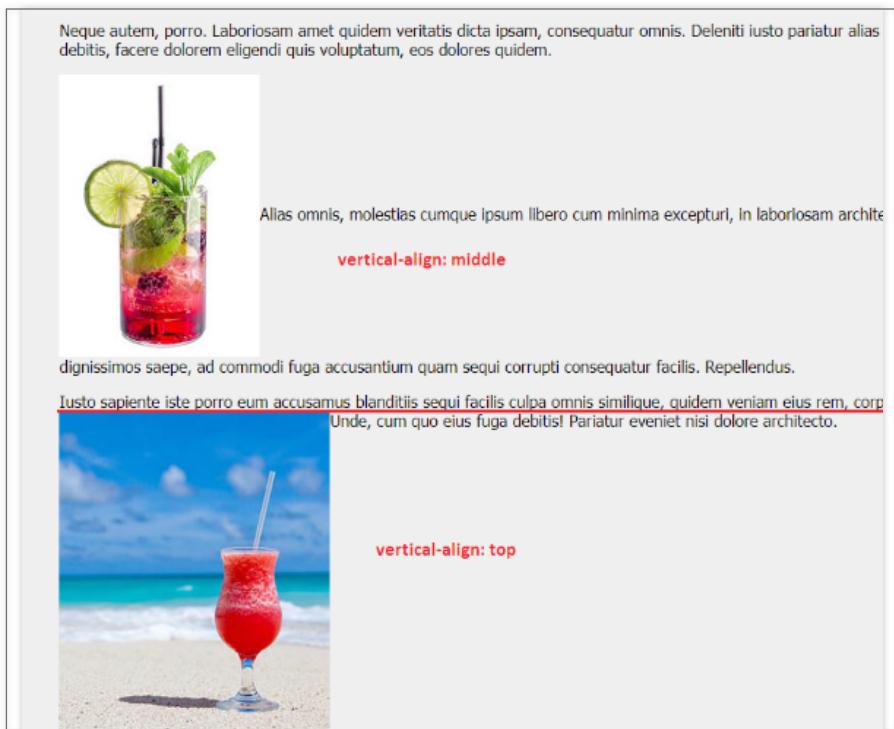
Neque autem, porro. Laboriosam amet quidem veritatis dicta ipsam, consequatur omnis. Deleniti iusto pariatur debitis, facere dolorem eligendi quis voluptatum, eos dolores quidem.

Рисунок 5

Что касается вставки изображений в контент страницы, то будучи добавленными в текст абзаца, они будут

расположены своим нижним краем по нижней линии текста. На рисунке 5 хорошо видно, что текст абзаца начинается сразу после изображения, «влипая» в него в самом низу.

Можно изменять местоположение текста, используя для картинки CSS-свойство vertical-align со значениями top, middle, baseline или bottom (по умолчанию), хотя визуально это страницу вряд ли украсит (рис. 6).



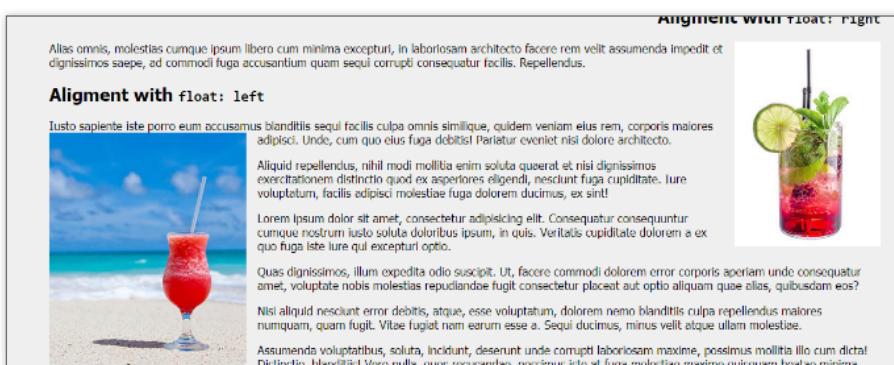
**Рисунок 6**

Тем не менее, вы можете использовать свойство vertical-align для выравнивания небольшого по размеру изображения относительно текста.

Чаще всего для выравнивания изображений применяют свойство float со значениями left или right, чтобы текст обтекал их справа или слева. Это свойство сейчас следует использовать вместо устаревшего атрибута align, добавляя к нему внешние отступы справа или слева, чтобы текст не «влипал» в край картинки. Обычно эти свойства записывают для классов, а уже классы добавляют к различным картинкам в качестве атрибутов:

```
.alignleft { float: left;
    margin-right: 15px; }
.alignright { float: right;
    margin-left: 15px; }
```

К двум изображениям (рис. 7) были применены эти классы. В результате текст разместился более привычным образом. Причем неважно, каково значение css-свойства vertical-align для этих картинок.



**Рисунок 7**

Если же требуется разместить изображение по центру, можно использовать либо вложенность его в блоч-

ный элемент с `text-align: center`, либо превратить изначально строчный элемент в блочный с помощью свойства `display: block` и задать ему `margin: auto` (рис. 8-9).

```
.text-center { text-align: center; }





```

Img in div with `text-align: center`



Рисунок 8

```
.center-block {
    display: block;
    margin: auto;
}

```

Center img with `display: block; margin: auto;`



Рисунок 9

Все примеры вы найдете в файле *image-alignment.html* в папке *examples*.

## Атрибуты тега <img>

Тег <img> может также содержать другие атрибуты, а именно:

1. **width** — ширина изображения в px (не указываются в кавычках).
2. **height** — высота изображения в px (не указываются в кавычках).
3. **sizes** — размеры изображения для разных макетов страницы при разных разрешениях экрана. Используется вместе с медиа-запросами, поэтому будет рассмотрен в дальнейших уроках.
4. **srcset** — позволяет указать одного или несколько через запятую, указывающих набор возможных изображений для отображения в браузере в зависимости от плотности пикселей устройства (обычные экраны и ретина-дисплеи). В атрибуте указываются:
  - ▷ адрес изображения;
  - ▷ ширина изображения в виде целого положительного числа, за которым следует буква w (например: 600w);
  - ▷ плотность пикселей в виде целого положительно-го десятичного числа, за которым сразу следует x (по умолчанию 1x, можно указать 2x).

Например,

```

```

- **ismap** — указывает на то, что картинка является серверной картой-изображением, которая позволяет привязывать ссылки к разным областям одного изображения;
- **usemap** — связывает картинку с картой-изображение, задаваемую с помощью контейнера `<map>`.

Последние два атрибута сейчас используются крайне редко, поэтому на них мы останавливаться не будем.

Наиболее часто используются атрибуты `width` и/или `height`, которые позволяют несколько ускорить загрузку страницы, т.к. заранее сообщают браузеру, сколько места необходимо оставить для отображения каждой картинки после ее получения. Эти атрибуты можно указывать как в соответствии с реальными размерами картинки, так и с уменьшением или с увеличением. В последнем случае изображение будет растянуто, что чаще всего некрасиво смотрится в браузере.

В файле `image-size.html` в папке `examples` вы найдете текст, в котором вставлены несколько изображений. Первое из них было уменьшено в половину за счет изменения ширины и высоты в атрибутах `width` и `height` (рис. 10).

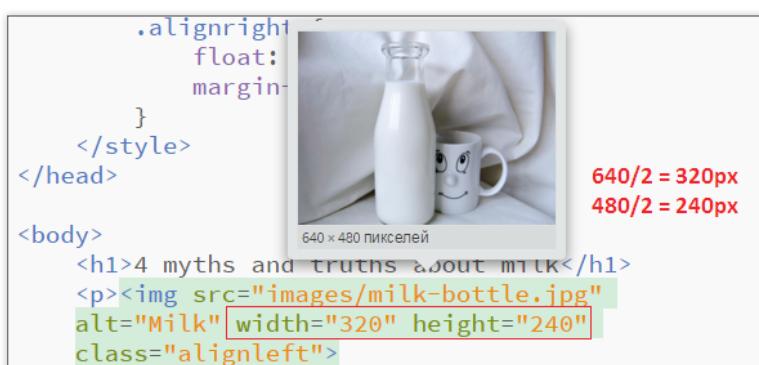


Рисунок 10

Вторая картинка вставлена с теми размерами в атрибутах width и height, которые характерны для этого изображения (рис. 11).

```

<head>
  <style>
    .alignright {
      float: right;
      margin-right: 10px;
    }
  </style>
</head>
<body>
  <h1>4 myths and truths about milk</h1>
  <p> We've heard some interesting things about cow's milk over the years. I am going to tell you four myths about milk and patients' parents.</p>
  <p><strong>Myth 1: Don't give milk to a child with fever, the milk will curdle (or some other variant).</strong></p>
  <p> |</p>

```

Рисунок 11

Если вы перепутаете значения величин в атрибутах или неверно зададите пропорции картинки, она будет исажена в браузере — вытянута по вертикали (рис. 12-13).

```

<head>
  <style>
    .alignleft {
      float: left;
      margin-right: 10px;
    }
  </style>
</head>
<body>
  <h1>4 myths and truths about milk</h1>
  <p> |</p>
  We've heard some interesting things about cow's milk over th

```

Рисунок 12

- ▶ **Примечание:** мы не будем рассматривать такие атрибуты, как align, border, hspace, vspace, longspace, т.к. они устарели в HTML5. Просьба не использовать их при верстке домашних заданий, т.к. на данный момент они заменяются CSS-свойствами.



We've heard some interesting things about from my patients' parents.

**Myth #1:** Don't give milk to a child with a fever.

**Truth:** As long as your child is not vomiting or having diarrhea, it's fine to give your febrile child. In fact, it is superior to giving water, which is very typical of a child with a fever. Refusing to eat, which is very typical of a child with a fever, can lead to dehydration and loss of appetite. So if your child stops eating while you're trying to get them to eat, offer some nutrition. Milk has energy and nutrients (germs). Take milk, add a banana and a little honey (if your child is under one year), and maybe some peanut butter in a blender, and make a nourishing milk shake. Children with fevers need extra hydration anyway. Even if your child is not vomiting, offer breast milk, not plain water. The milk will help keep your child hydrated. If, on the other hand, your child is vomiting, offer small amounts of clear liquids until their stomach settles.

**Myth #2:** Don't give children milk when they have a cold because the milk will make them sicker.

Рисунок 13

Ссылки по теме:

1. [https://www.w3schools.com/tags/tag\\_img.asp](https://www.w3schools.com/tags/tag_img.asp);
2. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img>.

## CSS-свойства для изображений

На замену устаревшим атрибутам пришли CSS-свойства, которые имеют больше значений и позволяют лучше отформатировать изображения. Например, можно использовать различные варианты рамок с помощью свойства border:

```
.bordered {  
    padding: 3px;  
    border: 3px double #cfcfcf;  
}
```

```

```

Результат:



Рисунок 14

Также можно использовать border-radius и box-shadow:

```
.radius {border-radius: 20px; }
.shadow {box-shadow: -6px -6px 6px #a7a7a7; }
.around-shadow {box-shadow: 0 0 8px
                  rgba(0, 0, 0, 0.65); }


```

```
...  

```

Результат применения стилей классов:

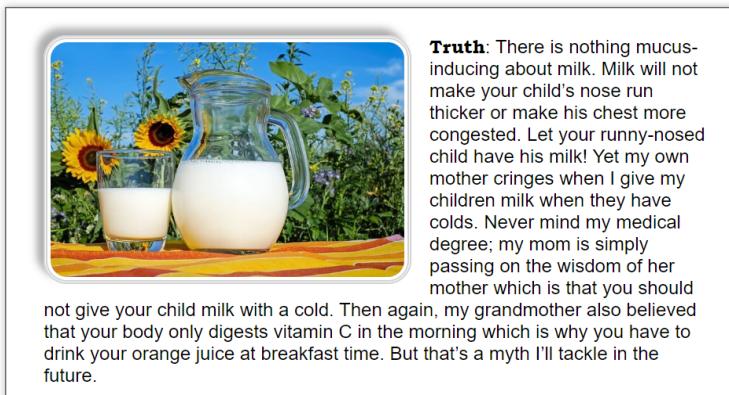


Рисунок 15

Также для изображений часто применяют различные эффекты при наведении курсора мыши. Для этого служит псевдокласс `:hover`:

```
.shadow { box-shadow: -6px -6px 6px #a7a7a7; }  
.shadow:hover {  
    box-shadow: 6px 6px 6px #a7a7a7;  
    cursor: pointer;  
}  
.effect:hover {  
    opacity: .5;  
    cursor: pointer; }
```

```

```

В первом случае псевдокласс :hover применяется к классу shadow, и при наведении курсора мыши меняет направление тени. Во втором случае класс effect сам по себе не имеет правил, они описаны только для псевдокласса :hover, в котором происходит изменение непрозрачности до 50% (opacity: .5) и внешнего вида курсора (cursor: pointer) — вместо стандартной стрелки будет рука с пальцем (рис. 16). О видах курсоров можно почитать в [соответствующей статье](#).

### 4 myths and truths about milk



We've heard some interesting things about cow's milk over the years. I am going to share with you four myths about milk that I continue to hear from my patients' parents.

**Myth #1:** Don't give milk to a child with a fever, the milk will curdle (or some other variant).

**Truth:** As long as your child is not vomiting, milk is a perfectly acceptable fluid to give your febrile child. In fact it is superior to plain water if your child is refusing to eat, which is very typical of a child with a fever. Fevers take away appetite. So if your child stops eating while she is sick, at least she can drink some nutrition. Milk has energy and nutrients, including protein (grams). Take milk, add a banana and a drizzle honey

(if your child is older than one year), and maybe some peanut butter for added protein, pour it into a blender, and make a nourishing milk shake for your febrile child. Children with fevers need extra hydration anyway. Even febrile infants need formula or breast milk, not plain water. The milk will not curdle or upset them in any way. If, on the other hand, your child is vomiting, then stick to clear fluids until her stomach settles.

**Myth #2:** Don't give children milk when they have a cold because the milk will give them more mucus.



**Truth:** There is nothing mucus-inducing about milk. Milk will not make your child's nose run thicker or make his chest more congested. Let your runny-nosed child have his milk! Yet my grandmother believed that you never give children milk when they have colds. Never mind my medical degree, my mom is simply passing on the wisdom of her mother which is that you should not give your child milk with a cold. Then again, my grandmother also believed that your body only digests vitamin C in the morning which is why you have to drink your orange juice at breakfast time. But that's a myth I'll

tackle in the future.

**Myth #3:** You can't overdose a child on milk.

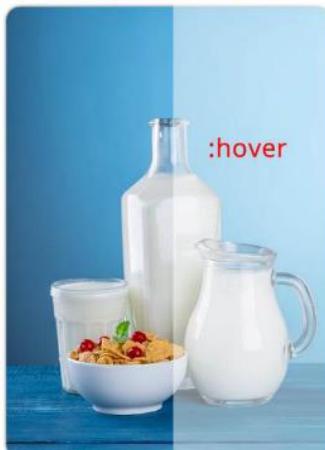


Рисунок 16

## Форматы графических файлов в Web

В прошлом уроке мы рассмотрели несколько вариантов сохранения (экспорта) в различных форматах. Графические программы позволяют сохранить изображения в самых разных форматах — *PSD, AI, CDR, TIFF, SKETCH, XD*, но далеко не все из них можно использовать на html-страницах. Главным требованием к файлам изображений для web является небольшой вес при хорошем качестве. Поэтому на html-страницах используют такие форматы графики:

- **JPEG** (по названию организации-разработчика *Joint Photographic Experts Group*) — формат, поддерживающий миллионы цветов. Основным расширением является *JPG*, хотя вы можете встретить такие варианты, как *JFIF, JPE* или *JPEG*. Этот формат, прежде всего, предназначен для реалистичных фотоизображений, в том числе с градиентными переходами. Поддерживает сжатие с потерями, за счет чего файлы могут иметь малый вес при достаточно большом разрешении. Не поддерживает ни анимацию, ни прозрачность. При сохранении изображений в этом формате нужно выбрать качество (0–100%), в зависимости от которого размер файла будет больше (80–100%) или меньше (50–70%). При сохранении картинок с невысоким качеством сжатие приводит к появлению артефактов, отсутствующих на исходном изображении.
- **GIF** (от англ. *Graphics Interchange Format* — «формат для обмена изображениями») — был долгое время очень популярным форматом в web, т.к. позволяет

сохранять изображения с прозрачностью (логотипы) и с анимацией (аватары, открытки). Полупрозрачность пикселей (технология альфа-канала) не поддерживается, поэтому для изображений с тенью он не подходит. Имеет ограничения в количестве используемых цветов — максимум 256, которые GIF индексирует, поэтому изображения с градиентами и массой переходов между цветами — не для этого формата. Зато изображения с областями сплошного цвета сохраняют очень хорошо. Картинки с анимацией при большом размере в пикселях могут иметь очень приличные размеры в килобайтах. Также формат GIF допускает чересстрочное хранение данных. Это значит, что строки разбиваются на группы, и при загрузке изображение проявляется постепенно, в несколько проходов по горизонтали.

- **PNG** (от англ. *portable network graphics*) — формат, который выполняет сжатие без потерь. Был создан, как замена GIF. Распространен в 3-х вариантах:
  - ▷ **PNG-8** (очень похож по настройкам на GIF), который позволяет сохранить индексированное изображение с максимумом в 256 цветов, в том числе и с прозрачным цветом,
  - ▷ **PNG-24**, который, помимо прозрачного цвета, может сохранить миллионы цветов подобно JPEG-изображениям,
  - ▷ **PNG-32**, в котором помимо миллионов цветов, есть поддержка альфа-канала, позволяющего сохранить полупрозрачные области с тенями, например.

Что касается размера сохраняемого файла, необходимо сравнивать файлы в обоих этих форматах (PNG и JPEG), если не нужна прозрачность.

Разработан также формат APNG для анимированных PNG-изображений, но пока он не поддерживается всеми браузерами, хотя перспективы у него хорошие.



Рисунок 17

**SVG** (от англ. *Scalable Vector Graphics* — масштабируемая векторная графика) — формат хранения векторной, а не растровой (как все предыдущие) картинки, который основан на языке разметки XML. В какой-то степени он приближен к технологии Flash, но, в отличие от Flash, рекомендован к использованию организацией W3C и имеет открытый код. **SVG** — это такой интересный формат, который, с одной стороны, можно редактировать в графических программах типа Adobe Illustrator, Inkscape или Sketch, с другой — в текстовом редакторе кода (Brackets, Sublime Text, Atom, Visual Studio Code и т.п.). Плюсом его является *масштабируемость*, т.е. картинка не искажается при любом увеличении/уменьшении, т.к. представляет собой набор векторов, описываемых математическими

уравнениями. Также это формат позволяет задавать для него специальные CSS-свойства и анимацию.

Относительно недавно появился формат **WEBP**, который разработала и внедрила в web компания Google в 2010 году. Он основан на алгоритме сжатия неподвижных изображений (ключевых кадров) из видеокодека VP8 и предполагает сжатие изображений как с потерями, так и без потерь качества. [Поддержка браузерами](#) пока еще далека от 100%, но в перспективе может такой стать.



Рисунок 18

Браузеры также поддерживают форматы BMP и ICO, но они настолько узкоспециализированы, что в реальной верстке их не используют. Разве что формат ICO применяется для фавиконов (*favicon — favorite icon* — дословно с англ. «любимый значок») — для отображения логотипа сайта или его части на вкладке браузера (рис. 19).



Рисунок 19

Добавляется ссылка на фавикон в блоке head вашей страницы с помощью тега link:

```
<link rel="shortcut icon" href="images/logo/favicon.ico">
```

На данный момент, в основном используются изображения в формате JPG, PNG, SVG, а формат GIF утратил свою былую популярность и используется главным образом для анимации. Вы найдете ряд примеров с GIF-анимацией в папке *examples/images/gifs*, а также сможете посмотреть на пример использования этих изображений в файле *event-with-gif.html*.

Ссылки по теме:

1. <https://developer.mozilla.org/en-US/docs/Web/SVG>;
2. [https://en.wikipedia.org/wiki/Portable\\_Network\\_Graphics](https://en.wikipedia.org/wiki/Portable_Network_Graphics);
3. <https://en.wikipedia.org/wiki/GIF>;
4. <https://en.wikipedia.org/wiki/JPEG>;
5. [https://en.wikipedia.org/wiki/Scalable\\_Vector\\_Graphics](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics);
6. <http://www.patrickhansen.com/2011/02/04/png-8-24-32-what/>.

## Верстка статьи с изображениями

Используем изображения для верстки статьи, которая будет похожа на небольшой рекламный блок. Вы найдете готовый файл с названием *article-with-images.html* в папке *examples*, а заготовку для работы — в файле *article-with-images-start.html*, чтобы вы могли повторить шаги, описанные далее. Внешний вид статьи представлен на рисунке ниже.

The screenshot shows a section of the InkyDeals website with a teal header bar at the top. Below it is a white page with a light gray border. At the top left is the InkyDeals logo, featuring a blue cartoon owl icon and the text "InkyDeals". To the right of the logo is the tagline "Daily Deals for Creative Professionals". The main content area contains several sections with icons and text:

- Average discount:** The average discount for our products was **93.40%**.  
Icon: A teal percentage sign (%) inside a white circle.
- Average user rating:** The average user rating for our deals was **4.5 out of 5 stars**.  
Icon: A teal five-pointed star inside a white circle.
- Total number of deals:** In the last 5 months we've launched **134 new deals**.  
Icon: A teal shopping cart icon inside a white circle.
- Client retention:** More than 78% of our clients return for a second purchase within the next 56 days.  
Text: Don't hesitate to drop us a line whether you'd like to tell us something about you or if you want to know more about a deal, a design bundle, or us – cause we're pretty sociable:).

Рисунок 20

Начнем с разметки:

```
<article>
  <header>
    <div id="logo">
      
    </div>
    <div id="slogan">
      <h1>Daily Deals for Creative Professionals</h1>
    </div>
  </header>

  <div id="main">
    <div class="block">
      <h2>Here is what InkyDeals is all about</h2>
      <p>We've made a lot of friends along ...</p>
    </div>
    <div class="block">
      
      <h3>Average user rating:</h3>
      <p>The average user rating for our deals was
         <strong>4.5 out of 5 stars</strong>
      </p>
    </div>
    <div class="block">
      
      <h3>Total saved by member:</h3>
      <p>InkyDeals members have saved
         <strong>$28,320,804</strong> in 2018 alone.</p>
    </div>

    <div class="block">...</div>
    <div class="block">...</div>
    <div class="block">...</div>
    <div class="block">
```

```
<h3>More than 78% of our ... </h3>
<p>Don't hesitate to drop us ...</p>
</div>
</div> <!-- /#main -->
</article>
```

The screenshot shows a portion of the InkyDeals website. At the top left is a blue octopus logo with the text "InkyDeals". Below it is a section titled "Daily Deals for Creative Professionals". A heading "Here is what InkyDeals is all about" is followed by a paragraph of text. To the left of the text is a circular icon containing a percentage sign (%). Below this icon is the text "Average discount:" and a sentence about the average discount. To the left of the sentence is a circular icon containing a star. Below this icon is the text "Average user rating:" and a sentence about the average user rating.

InkyDeals

Daily Deals for Creative Professionals

Here is what InkyDeals is all about

We've made a lot of friends along the way and nothing would make us happier than being able to consider you part of c

%

Average discount:

The average discount for our products was 93.40%

★

Average user rating:

The average user rating for our deals was 4.5 out of 5 stars

Рисунок 21

В разметке есть корневой элемент `<article>`, внутри которого есть тег `<header>` и `<div id="main">`. Внутри основного div-а расположены несколько div-ов с классом `block`. Первый и последний из `<div class="block">` содержат

только заголовок и текст, а все остальные — еще и изображения с классами alignleft и alignright, которые нужны для размещения изображений слева или справа от текста.

Пока стили не заданы, внешний вид страницы представлен на рисунке 21.

Приступим к css-форматированию.

```
* { box-sizing: border-box; }

body {
    font-family: Verdana, Geneva, sans-serif;
    background-color: #d4d4d4;
}

article {
    width: 1000px;
    margin: 40px auto;
    padding: 20px 150px 60px;
    background-color: #f0f0f0;
    box-shadow: 0 0 16px rgba(0, 0, 0, 0.7);
}
```

Все правила должны быть вам знакомы, поэтому подробно останавливаться на них не будем. Отметим лишь самое первое — для всех элементов мы задаем box-sizing: border-box, чтобы браузер учитывал рамки (border) и внутренние отступы (padding) в размере ширины (width) элемента. Если посмотреть на значение width: 1000px у article, то этот элемент должен занять большую часть экрана, однако article имеет также довольно приличный padding: 20px 150px 60px. Поэтому за счет свойства box-sizing: border-box на ширину, занимаемую контентом, придется 1000px — 150px\*2 = 700px.

Вид страницы в браузере изменился:

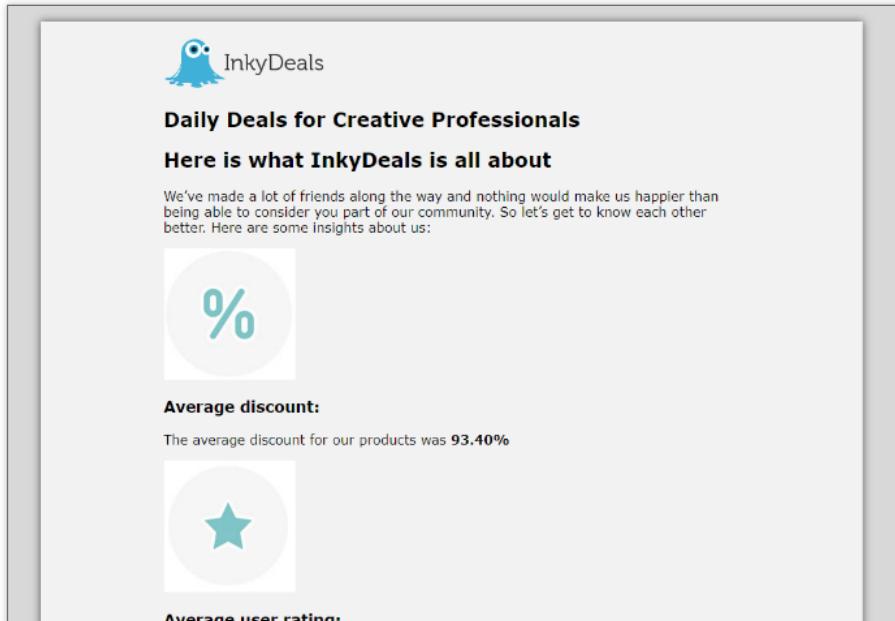


Рисунок 22

Добавим рамку к div id="main" и изменим шрифты для заголовков:

```
#main { border: 2px solid #ddd; }
h1, h2, h3 { font-family: 'Century Gothic', monospace; }
```

Логотип расположим слева, а слоган — справа — для этого у нас в разметке есть div-ы с соответствующими id.

```
#logo { float: left; }
#slogan {
    float: right;
    font-style: italic;
```

```

font-weight: normal;
font-size: .7rem;
text-align: right;
color: #909090;
width: 30%;

}

```

Получим в результате весьма специфичный вид: рамка, заданная для div id="main", «переехала» выше логотипа и слогана и туда же подтянулся заголовок первого блока:

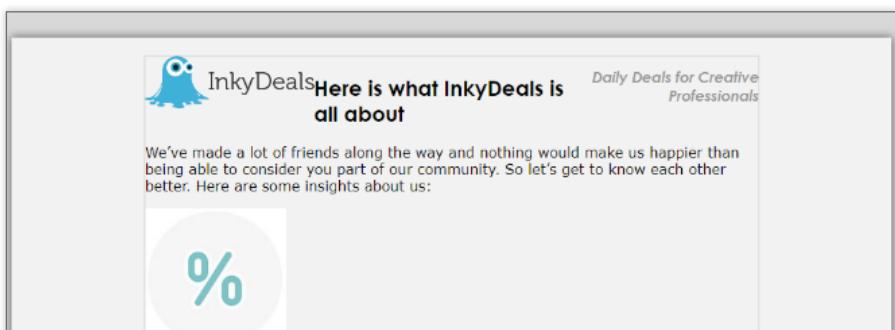


Рисунок 23

Все это произошло из-за использования свойства float: помните о его особенности превращать в 0 высоту родительского элемента (в нашем случае header)? Поэтому используем для header правила класса clearfix:

```

.clearfix::before, .clearfix::after {
    content: "";
    display: table;
}
.clearfix::after { clear: both; }

```

```
<header class="clearfix">
  <div id="logo">
    
  </div>
  <div id="slogan"> ... </div>
</header>
```

Добавив атрибут `class="clearfix"` к `header`, мы приведем внешний вид шапки статьи к нужному виду.

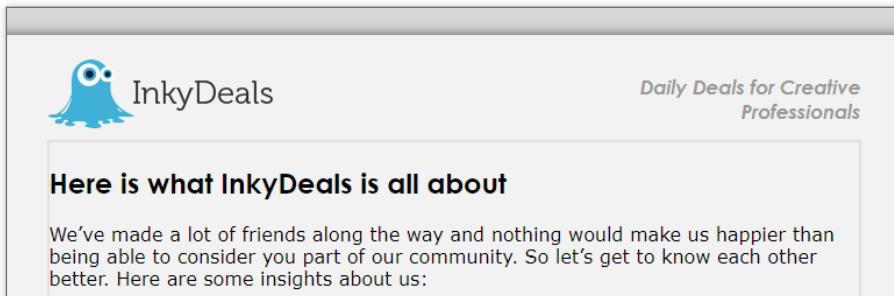


Рисунок 24

Изменим теперь внешний вид блоков, добавив стили классов для `div class="block"` и для изображений.

```
.alignleft {
  float: left;
  margin-right: 15px;
}

.alignright {
  float: right;
  margin-left: 15px;
}

.block { padding: 30px; }
```

Внешний вид блоков улучшился, но им не хватает выделения.

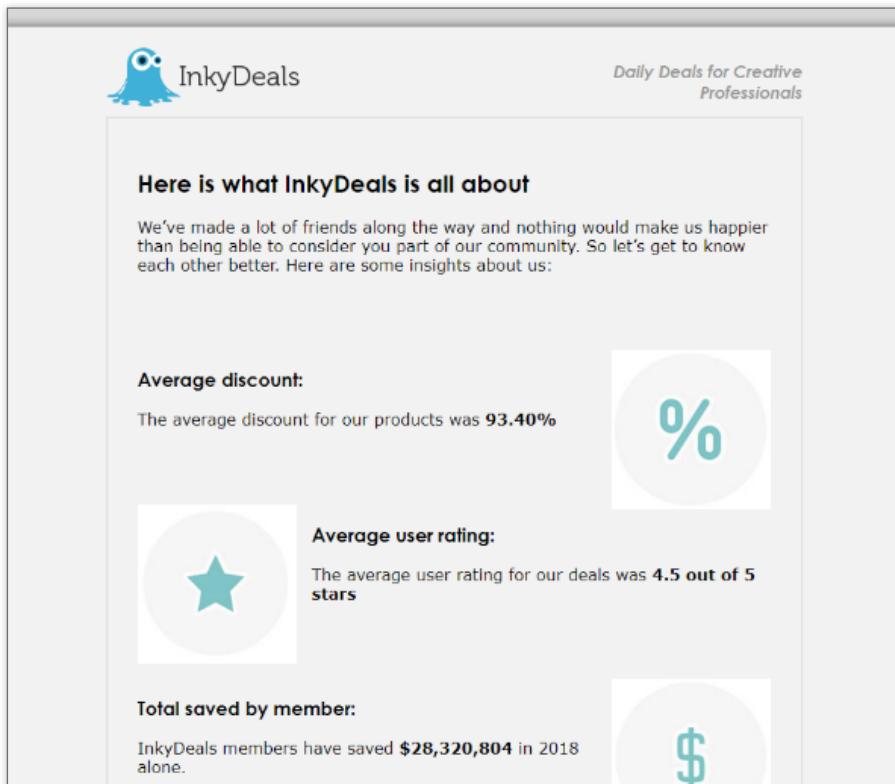


Рисунок 25

Добавим для нечетных блоков светлый цвет фона и рамки сверху и снизу:

```
.block:nth-child(odd) {  
    background-color: #f8f8f8;  
    border-bottom: 2px solid #ddd;  
    border-top: 2px solid #ddd;  
}
```

Однако, и здесь не обошлось без сюрпризов: рамка вверху и внизу увеличены и изображения выходят за пределы блока.

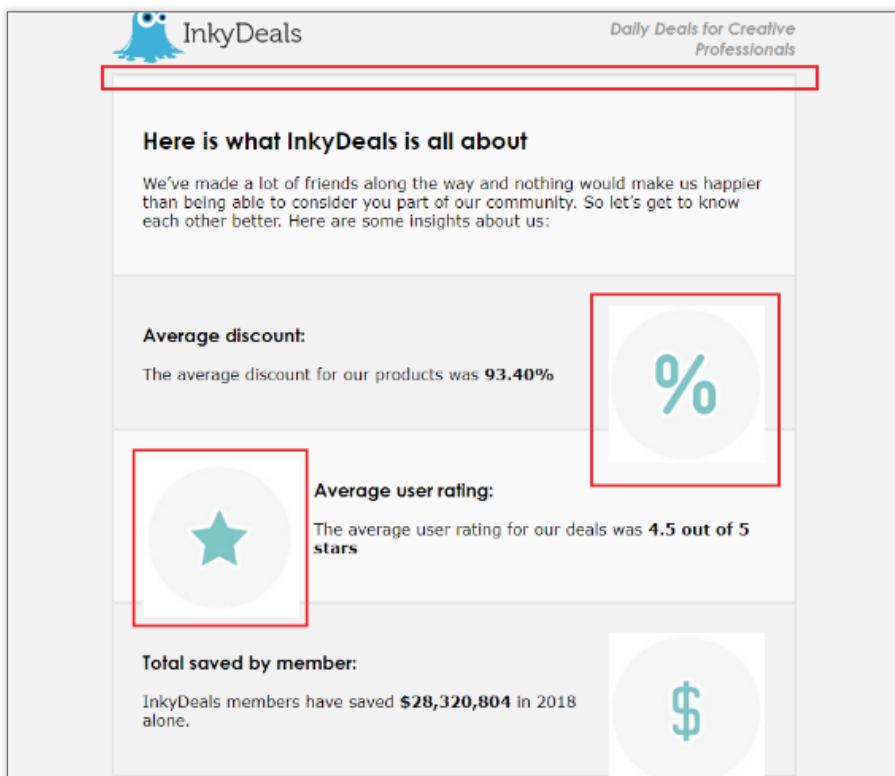


Рисунок 26

Для рамок несколько изменим свойства для div id="main":

```
#main {  
    border-left: 2px solid #ddd;  
    border-right: 2px solid #ddd;  
}
```

А для .block добавим те же правила, что и для класса clearfix:

```
.clearfix::before, .clearfix::after,
.block::before, .block::after {
    content: "";
    display: table;
}
.clearfix::after, .block::after { clear: both; }
```

Теперь изображения не выходят за пределы блоков:

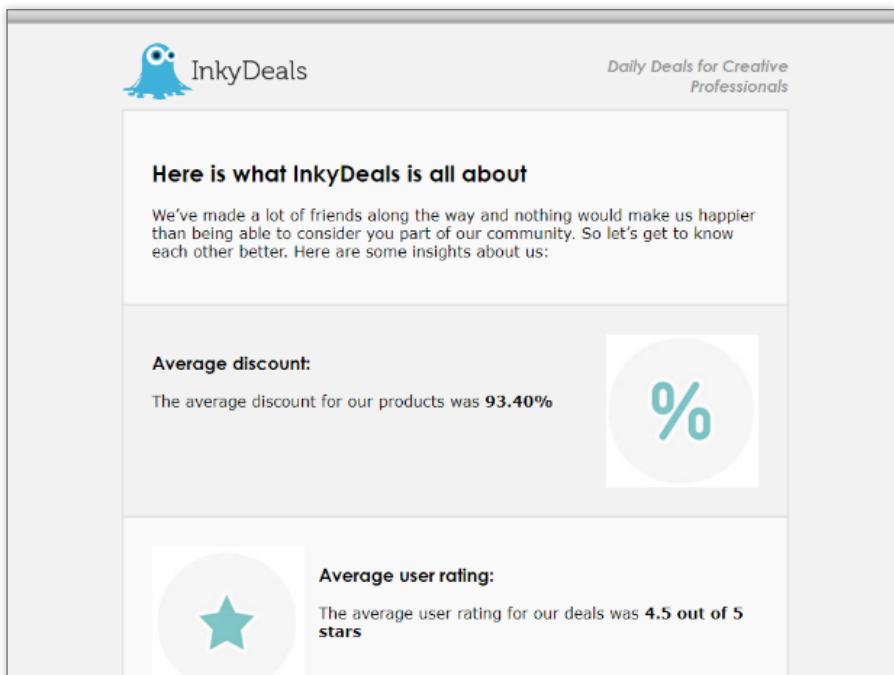


Рисунок 27

Еще одна проблема с изображениями заключается в том, что на фоне блока рамка вокруг изображений,

которые сохранены в формате JPG, выделяется некрасивым квадратом. Можно удалить фон вокруг изображения и повторно сохранить их в формате PNG с прозрачностью. Есть и другой способ — сделать изображения круглыми с помощью border-radius: 50%:

```
.block img {  
    border: 2px solid #ddd;  
    border-radius: 50%;  
}
```

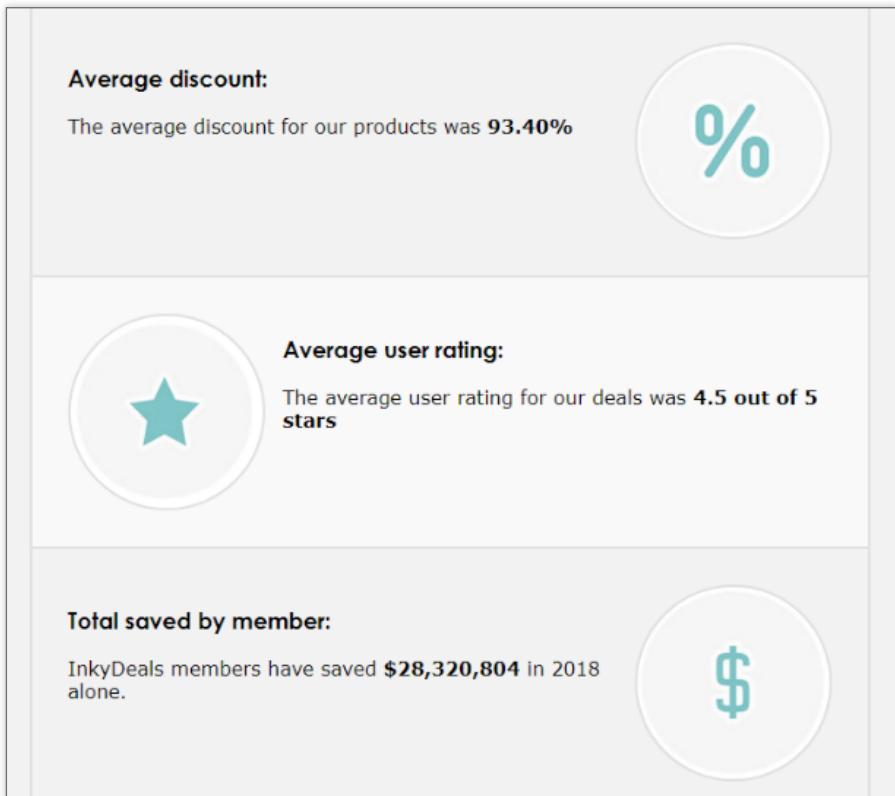


Рисунок 28

Теперь изменим форматирование для первого и последнего div class="block", в котором у нас есть только текст и увеличим отступы в header:

```
.block:first-child, .block:last-child {  
    text-align: center;  
}  
  
header {  
    padding: 30px 15px;  
}
```

Теперь статья приняла нужный вид.

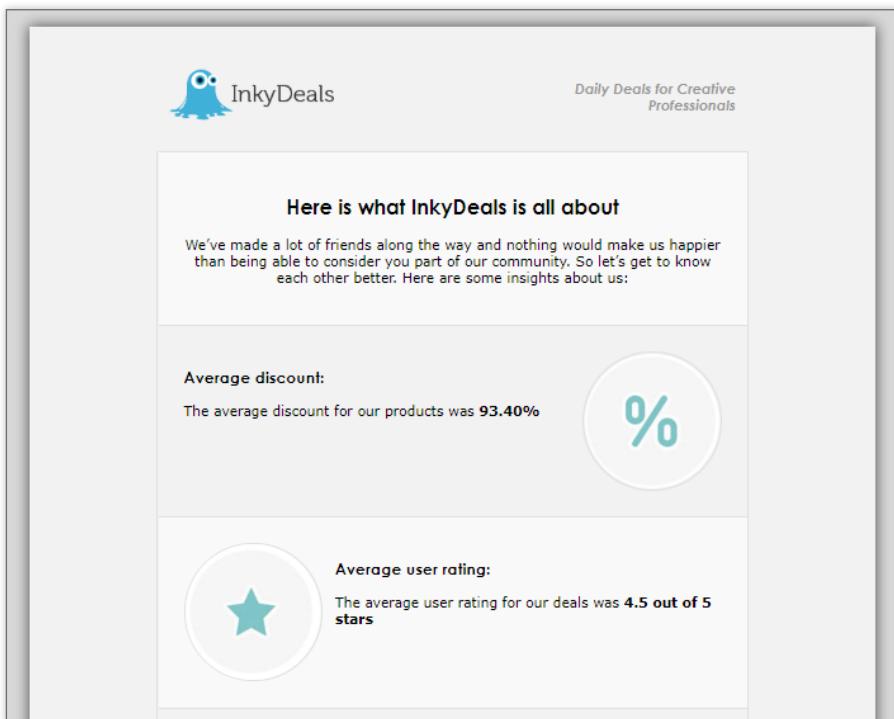


Рисунок 29

## Теги **figure** и **figcaption**.

Достаточно часто изображения сопровождаются заголовками. Если у вас на странице есть такая ситуация, то можно использовать теги `<figure>` и `<figcaption>`.

Тег `<figure>` предназначен для вставки изображений, кода, видео и даже текста, например, стиха или рекламного объявления. Этот элемент представляет собой автономный блок контента. Это означает, что если вы передвинете такой блок в любое место документа, то это не повлияет на смысл разметки всей страницы в целом.

Нас интересует этот тег именно с точки зрения вставки изображений с описанием. В этот тег можно вставить одну или несколько картинок, добавив для них описание внутри тега `figcaption`. Это может быть участок статьи с описанием, например:

```
<figure>
    
    <figcaption>
        Авиакомпания <strong>Pegasus Airlines</strong>
        (Турция) сегодня открыла прямой регулярный
        рейс <strong>Анкара-Одесса-Анкара</strong>.
        Об этом сегодня на пресс-конференции сказал
        директор международного аэропорта «Одесса»
        Виталий Портянко, передает УНИАН.
    </figcaption>
</figure>
```

По умолчанию, тег `figure` имеет внешние отступы вверху и внизу в `1em`, а слева и справа — в `40px`.

**Tags: <figure> & <figcaption>**



Авиакомпания *Pegasus Airlines* (Турция) сегодня открыла прямой регулярный рейс Акапа-Одесса-Акапа. Со этим событием на пресс-конференции оказал спикер международного аэропорта «Одесса» Виталий Портников, передает УНИАН.

```
<body data-brackets-id="80">
  <h1 data-brackets-id="81">Tags: <figure> & <figcaption></h1>
  ...
  <figure data-brackets-id="82"> -- 50
    
    <figcaption data-brackets-id="84">...</figcaption>
  </figure>
</figure data-brackets-id="85">...</figures>

```

Styles Computed Event listeners DOM Breakpoints »  
Filter .hov .cls

```
figure {
  width: 660px;
  padding: 10px;
  border: 3px double #000;
}

figure {
  display: block;
  margin-block-start: 2em;
  margin-block-end: 1em;
  margin-inline-start: 40px;
  margin-inline-end: 40px;
}
```

Inherited from `body`:

**Рисунок 30**

Вы можете задать свои стили для того, чтобы оформить этот блок. Для figcaption также их можно назначить.

Вы найдете этот и еще один пример в файле *figure-figcaption.html* в папке examples.

В Интернете вы можете найти самостоятельно ряд примеров с оформлением изображений и подписей к ним с помощью этих тегов с добавлением CSS-анимации и/или JavaScript, например, на *codepen.io*.

### Ссылки:

- <https://html.spec.whatwg.org/multipage/grouping-content.html#the-figure-element>;
- <https://www.sitepoint.com/quick-tip-the-right-way-to-use-figure-and-figcaption-elements/>.

## Свойства группы background

Помимо вставки изображения на страницу посредством тега *<img>* мы можем управлять графикой помошью CSS. В этом случае мы будем говорить о целой группе свойств, которая начинается со слова *background*.

С одним из свойств этой группы мы уже знакомы — это background-color. Напомним, что задать его можно таким образом:

```
background-color: red | #ff0000 | #f00 |  
      rgb(255, 0, 0) |  
      rgba(255, 0, 0, .7) |  
      transparent (по умолчанию)
```

Следующим, пожалуй, самым важным свойством группы background является background-image. По умолчанию элементы не имеют фонового изображения, т.е. значение background-image установлено как none. С помощью none также можно удалить фоновое изображение, например, с помощью JavaScript при клике на элементе.

Синтаксис свойства background-image таков:

```
background-image: none (по умолчанию) |  
      url(путь_к_изображению.расширение);
```

Например,

```
background-image: url(images/bg-top.jpg);
```

По умолчанию браузер повторяет указанное изображение по горизонтали и по вертикали, укладывая его плитками. Это поведение можно изменить, используя свойство background-repeat с такими значениями:

```
background-repeat: repeat (по умолчанию) |  
      no-repeat | repeat-x | repeat-y |  
      space | round
```

При повторении фоновой картинки важно, чтобы стыки или совмещения соседних изображений были не видны, т.е. нужно использовать бесшовные текстуры. В папке *examples/background* вы найдете файл *bg-repeat.html*, для body которого задано такое фоновое изображение:

```
body { background-image: url(bg/white-bg.jpg); }
```



**Рисунок 31**

Теперь давайте заменим фон другой картинкой — *examples/background/bg/abstract-bg.jpg*.

Смена фонового изображения приводит сразу к 2-м выводам (рис. 32):

- Явно видимые стыки соседних изображений плохо выглядят и по горизонтали, и по вертикали;
- Яркое фоновое изображение делает текст нечитаемым.

```
body { background-image: url(bg/abstract-bg.jpg); }
```

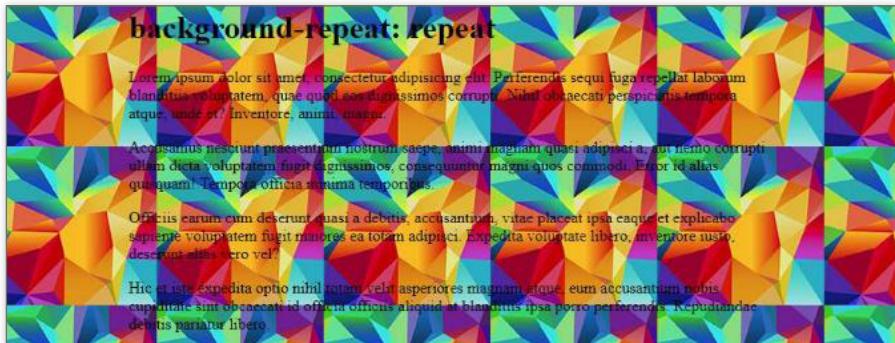


Рисунок 32

Последнее замечание важнее будет для дизайнера, но и вам, как верстальщикам, нужно обращать внимание на такие моменты.

При использовании свойства `background-repeat` со значением `repeat-x` или `repeat-y` изображение будет про-дублировано только по горизонтали или по вертикали соответственно. Например, нам нужно сделать оформление страницы сверху (файл `bg-repeat-x-y.html` в папке `examples/background`). Тогда правила для `body` будут такие:

```
body { width: 70%;  
      margin: 230px auto 10px;  
      background-image: url(bg/paint.jpg);  
      background-repeat: repeat-x; }
```

Обратите внимание, что текст внутри `body` был сдвинут вниз за счет большого `margin-top` (230px) в составном свойстве `margin`. Следует отметить, что с другими элементами использование `margin` не даст нужного эффекта. Чтобы сместить контент от границы элемента следует использовать `padding`.



### **background-repeat: repeat-x**

*Lorem ipsum dolor sit amet, consectetur adipisicing elit. Perferendis sequi fuga repellat laborum blanditiis voluptatem, quae quod eos dignissimos corrupti. Nihil obsecrati perspiciat tempora atque, unde et? Inventore, animi, magni.*

*Accusamus nesciunt praesentium nostrum saepe, animi magnam quasi adipisci a, aut nemo corrupti ullam dicta voluptatem fugit dignissimos, consequuntur magni quiscommodi. Error id alias quisquam? Tempora officia minima temporibus.*

*Officia earum cum deserunt quasi a dabitis, accusantium, vitae placeat ipsa eaque et explicabo sapiente voluptatem fugit maiores ea totam adipisci. Expedita voluptate libero, inventore iusto, deserunt alias vero vel?*

*Lorem ipsum dolor sit amet, consectetur adipisicing elit. Mollitia, quam, eos. Voluptatibus, delectus. Autem nesciunt aut dolorum dolor, recusandae iure repudiandae dignissimos, laborum provident sapiente quam. Eaque nulla necessitatibus, praesentium?*

**Рисунок 33**

В тексте страницы есть еще элемент `blockquote`. Используем для него оформление с помощью повторяющегося по вертикали изображения:

```
blockquote {  
    background-image: url(bg/brush.png);  
    background-repeat: repeat-y;  
    margin-left: 0;  
    padding-left: 58px;  
}
```

### **background-repeat: repeat-y**

 *Lorem ipsum dolor sit amet, consectetur adipisicing elit. Mollitia, quam, eos. Voluptatibus, delectus nesciunt aut dolorum dolor, recusandae iure repudiandae dignissimos, laborum provident sapiente Eaque nulla necessitatibus, praesentium?*

 *Iste porro veniam, nulla. Aliquid at accusamus nobis officia, fugiat optio quas deleniti corrupti repe placeat eveniet doloremque quibusdam voluptate eius commodi. Labore dignissimos, beatae vel, ve illum nostrum unde.*

 *Atque dolorum fugit reiciendis provident ducimus simili que consequuntur quasi neque dolorem voluptates, error recusandae ratione ullam corporis exercitationem perspiciat, pariatur nam! Ven sunt molestias, ullam beatae ducimus magni quis debitibus.*

**Рисунок 34**

▶ **Примечание:** изображение *paint.jpg*, которое было использовано для оформления фона *body*, имеет светло-серый цвет фона и выделяется на белом фоне *body*. Поэтому фон всей страницы имеет смысл сделать таким же, как и на изображении. Для того чтобы взять образец цвета с изображения, можно использовать *Adobe Photoshop* и определить цвет, как *#f0f0f0*.

После сохранения стилей разница в оттенках цвета отсутствует.



**Рисунок 35**

Следующие 2 варианта повторения фоновой картинки — это *space* и *round*. Эти значения появились в CSS3, поэтому пользуются пока меньшей популярностью, чем остальные, привычные еще из CSS2.

При использовании значения space для свойства background-repeat фоновое изображение будет повторяться несколько раз, чтобы полностью заполнить область. Однако, в отличие от значения по умолчанию repeat, если изображение полностью не помещается в области элемента, между картинками появляется пустое пространство, причем иногда слишком большое (зависит от размеров фонового изображения и элемента). Повторение происходит только по горизонтали (в этом похоже на repeat-x), поэтому избыток текста может оказаться невидимым, как в файле примера *examples/background/bg-repeat-space-round.html*.

Значение round также помещает в области элемента целое число картинок, но при этом масштабирует их без сохранения пропорций.

```
.space { background-repeat: space; }
.round { background-repeat: round; }
```

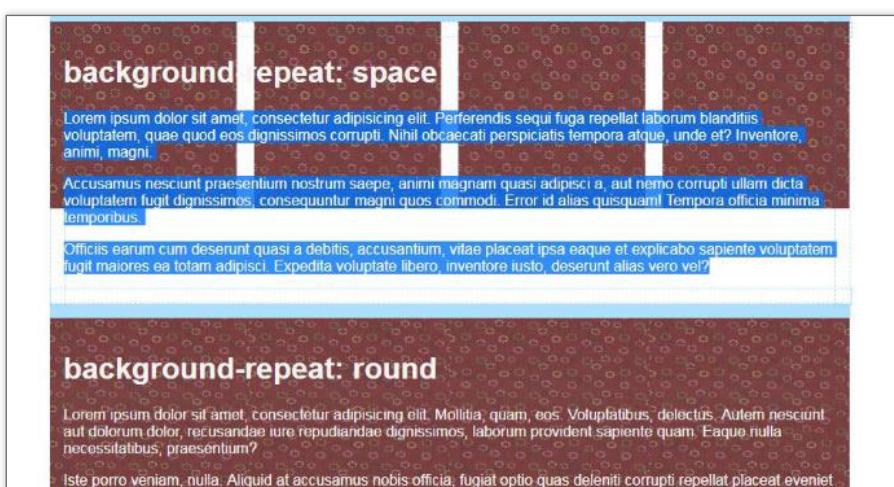


Рисунок 36

- ▶ **Примечание:** чтобы не получилось так, как в блоке с повторением типа space, элементу необходимо задавать не только фоновое изображение, но и цвет фона (background-color). Это поможет пользователю прочитать текст, даже, если изображение не загрузилось.

```
.block { ...;
    color: #f6f6f6;
    background-image: url(bg/flower-texture.jpg);
    background-color: #754143;
}
```



Рисунок 37

Цвет блока можно определить с помощью того же Adobe Photoshop.

Значение no-repeat предполагает, что фоновое изображение не повторяется, т.е. вставляется на страницу 1 раз.

Используем это значение для оформления <div id="wrap"> с текстом из 2-х абзацев.

```
#wrap {
    width: 70%;
```

```

margin: 20px auto;
background-image: url(bg/photo-bg.jpg);
background-repeat: no-repeat;
font-family: sans-serif;
}

```

Получилось не очень красиво.

### **background-repeat: no-repeat**

#### **Зеркальный фотоаппарат — как выбрать и купить**

Среди современных цифровых фотоаппаратов выделяют компактные камеры или цифровые «мыльницы» и зеркальные фотоаппараты. Что касается «цифромыльниц», то они являются оптимальным вариантом для любительской фотографии, поскольку очень компактны и просты в управлении. Они обладают рядом автоматических съемочных режимов для максимального удобства пользователя. Более продвинутые пользователи и профессиональные фотографы выбирают зеркальные фотоаппараты, то есть камеры, использующие для выбора съемочного объекта видосистему с полным выкиривания, в оптическую схему которого входит зеркало.

Зеркальный фотоаппарат позволяет использовать различные съемочные аппараты, которые предлагают фотограф широкие возможности по ручной настройке параметров съемки. Также современные зеркальные аппараты отличаются от обычных компактных фотоаппаратов превосходной работой системы автофокуса и матрицей, обладающей повышенными физическими разрешениями и высокой цветопередачей. Все это, в конечном счете, обеспечивает непревзойденное качество снимков.

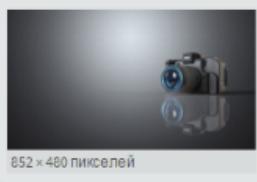
**Рисунок 38**

Исправим ситуацию, воспользовавшись либо подсказками в текстовом редакторе (Brackets), либо открыв изображение в папке:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>background-repeat</title>
    <style>
        #wrap {
            width: 70%;
            margin: 20px auto;
            background-image: url(bg/photo-bg.jpg);
            background-repeat: no-repeat;
            font-family: sans-serif;
        }
    </style>

```



**Рисунок 39**

При наведении на путь к файлу изображения в значении свойства background-repeat мы видим саму картинку и ее размеры. Используем размеры нашего фона для формирования размеров div-a: вместо width: 70% установим ширину с картинки и добавим высоту, а также внутренние отступы и тень text-shadow для улучшения читабельности текста. Сразу вспомним об увеличении размера блока за счет отступов, поэтому укажем еще свойство box-sizing: border-box:

```
#wrap { ...;
  width: 852px;
  height: 480px;
  padding: 10px 30px;
  box-sizing: border-box;
  color: #f0f0f0;
  text-shadow: 1px 1px #444;
  box-sizing: border-box
}
```

Получился такой блок:

### background-repeat: no-repeat

#### Зеркальный фотоаппарат — как выбрать и купить

Среди современных цифровых фотоаппаратов выделяют компактные камеры или цифровые «мыльницы» и зеркальные фотоаппараты. Что касается «цифровых мыльниц», то они являются оптимальным вариантом для любительской фотографии, поскольку очень компактны и просты в управлении. Они обладают рядом автоматических съемочных режимов для максимального удобства пользователя. Более продвинутые пользователи и профессиональные фотографы выбирают зеркальные фотоаппараты, то есть камеры, использующие для выбора съемочного объекта видоискатель прямого визирования, в оптическую схему которого входит зеркало.

Зеркальный фотоаппарат позволяет использовать различные съемные объективы и также предлагает фотографу широкие возможности по ручной настройке параметров съемки. Такие современные «зеркалки» отличаются от обычных компактных фотокамер превосходной работой системы автофокуса и матрицей, обладающей большими физическими размерами и высокой светочувствительностью. Все это, в конечном счете, обеспечивает непревзойденное качество снимков.

Рисунок 40

Пример вы найдете в файле *bg-no-repeat.html* в папке *examples/background*.

## Свойство background-position

Свойство `background-position` отвечает за размещение фонового изображения относительно блока или страницы. Может быть указано с помощью ключевых слов или значений в px, pt, em, % и т. д. для координаты x и y (размещение по горизонтали и вертикали):

```
background-position: [left | center | right | % | px,  
                     em, pt и др.]  
                     [top | center | bottom | % | px,  
                     em, pt и др.]
```

Ключевые слова помогают разместить фоновое изображение по горизонтали (координата x — left, center, right) и по вертикали (координата y — top, center, bottom). Можно указать только одну координату в виде ключевого слова, тогда вторая будет использована по умолчанию как center. Порядок указания ключевых слов неважен. Если вы используете значения в любых единицах измерения, первой указывается координата X, через пробел — координата Y. При указании значения `background-position` в процентах следует учитывать, что ряд значений совпадает с ключевыми словами:

- 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 = center left = 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);

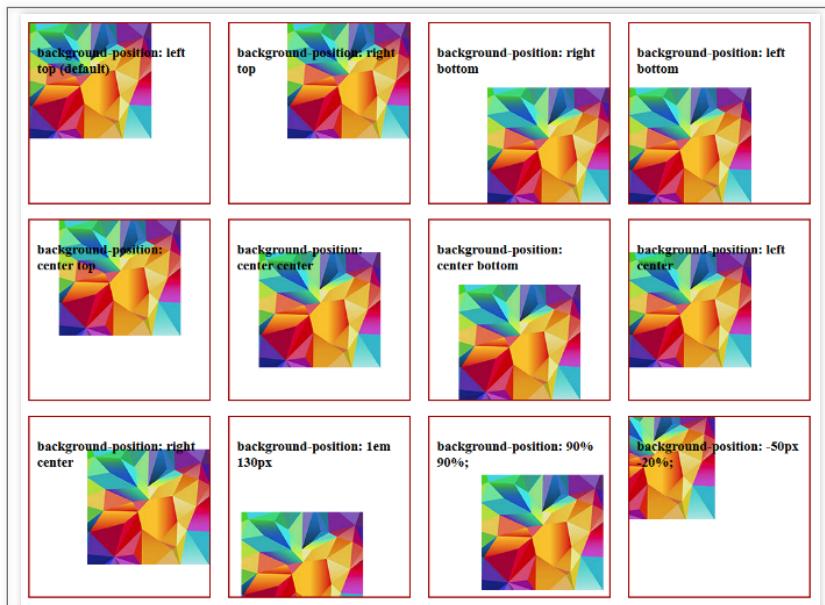


Рисунок 41

Посмотреть на результат размещения фонового изображения с `background-repeat: no-repeat` можно на ри-

сунке 41 и в файле *bg-position.html* в папке *examples/background*.

## Свойство background-size

Свойство `background-size` появилось в стандарте CSS3 и используется для управления размерами фонового изображения. Это очень удобное свойство, которое позволяет растягивать или, наоборот, уменьшать изображение, подгоняя его величину под размеры блока.

```
background-size: [ размер в px, pt, em, rem и др. | % |  
    auto ]{1,2} | cover | contain
```

- **Примечание:** поскольку это свойство появилось в спецификации CSS3, ряд браузеров, устаревших уже на данный момент, используют это свойство с вендорными префиксами. Информацию о них вы можете посмотреть на сайте [caniuse.com](http://caniuse.com).

Рассмотрим свойство `background-size` на примерах. На рисунке 42 (файл примера *bg-size.html*) видно, что по умолчанию (значение `auto`) это свойство не оказывает влияния на размер картинки — он совпадает с ее реальными размерами. Значение `100% 100%` растягивает изображение по размеру контейнера без сохранения пропорций. Значения `100% auto` или `auto 100%` растягивают изображение по горизонтальному или вертикальному размеру блока, сохраняя пропорции. То же делают значения `contain` и `cover` во втором ряду примера, но `cover` изменяет масштаб фона с сохранением пропорций так, чтобы его ширина или высота равнялась ширине

или высоте блока, а `contain` сохраняет пропорции так, чтобы картинка целиком поместилась внутрь блока. Значение `background-size: 50px 50px` привело к тому, что размер фонового изображения стал равен 50x50px, а значение 50% 50% отмасштабировало картинку так, что она заняла ровно половину ширины и высоты элемента без сохранения пропорций.

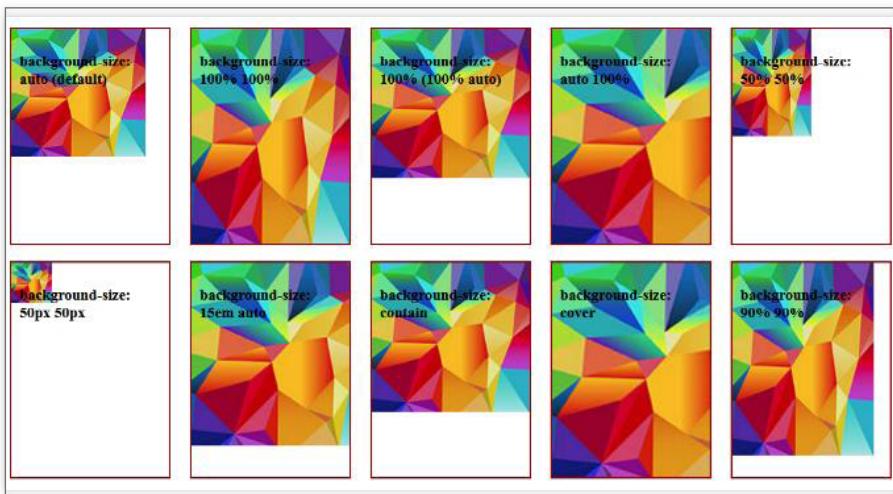


Рисунок 42

После рассмотрения всех предложенных вариантов можно сделать вывод, что значение `auto` для любого из двух параметров сохраняет пропорции фонового изображения. Если же установлено всего одно значение, то оно будет управлять шириной фонового изображения, второе значение при этом имеет значение `auto`, что позволяет сохранить пропорции картинки. Указание двух значений через пробел задаёт ширину и высоту фоновой картинки соответственно.

## Свойство background-attachment

Свойство `background-attachment` необходимо для указания того, будет ли фоновое изображение прокручиваться вместе с содержимым элемента или всей страницы. С помощью этого свойства можно зафиксировать изображение, что позволяет прокручивать контент поверх него. Варианты значений `background-attachment`:

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

Значение `scroll` (по умолчанию) прокручивает фоновое изображение вместе с контентом. Значение `fixed` фиксирует изображение, а контент прокручивается сверху.

Посмотреть на разницу в этих двух значениях можно в файле `bg-attachment.html`, папка `examples/background`. В нем для первого фона указано свойство `background-attachment: scroll`, которое смело можно закомментировать без изменения визуального отображения страницы, т.к. это значение свойства `background-attachment` по умолчанию.



Рисунок 43

Значение local фиксирует фон относительно содержимого элемента, т.е. при прокрутке страницы либо самого элемента фон прокручивается вместе с содержимым элемента. Посмотреть на работу этого значения можно в файле *bg-attachment-local.html* в папке *examples*.

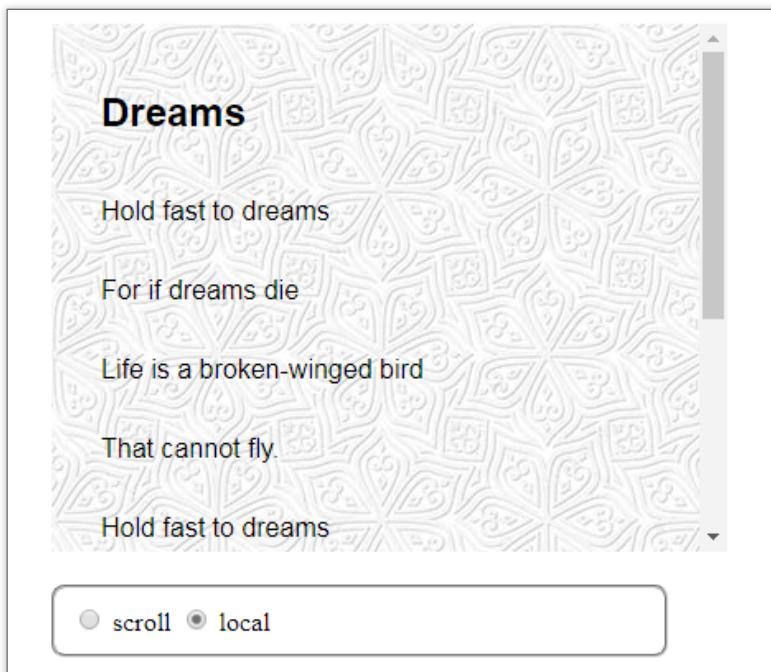


Рисунок 44

### Свойство background-origin

Свойство `background-origin` определяет, в какой области будет позиционироваться фоновое изображение. Особенность использования: нельзя применять `background-origin`, если значение `background-attachment` задано как `fixed`.

Возможные значения `background-origin`:

```
background-origin: padding-box | border-box | content-box
```

Визуально это свойство определяет, будет ли фоновый рисунок занимать пространство, включая внутренние отступы (`padding-box`), внутренние отступы + рамки (`border-box`) или только область контента без отступов и рамок (`content-box`).



Рисунок 45

Пример вы найдете в файле `bg-origin.html` в папке `examples`.

### Свойство `background-clip`

Свойство `background-clip` определяет, как цвет фона или фоновое изображение будет выводиться под рамками элемента. Действие этого свойства можно заметить либо при прозрачных, либо при пунктирных или точечных границах.

Возможные значения `background-clip`:

```
background-clip: padding-box | border-box (по умолчанию)
                | content-box | text
```

Первые 3 значения свойства background-clip подобны по внешнему виду работе свойства background-origin. Значение text позволяет как бы «вырезать» текст из фона. Для этого необходимо к остальным стилям добавить свойство color: transparent:

```
.box4 {
    background-clip: text;
    -webkit-background-clip: text;
    color: transparent;
}
```

Пример можно посмотреть в файле *bg-clip.html* в папке *examples*.

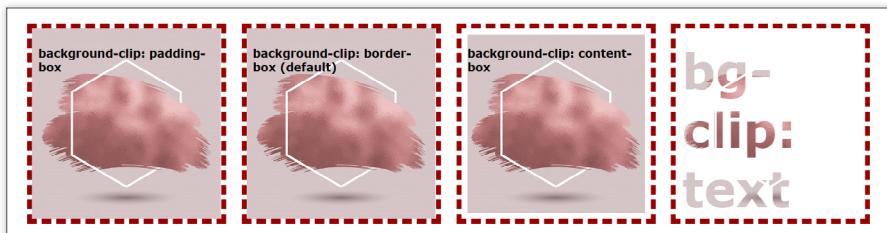


Рисунок 46

- ▶ **Примечание:** свойства *background-origin* и *background-clip* появились в стандарте CSS3 и для ранних версий Android и Opera не поддерживаются. Подробная информация — на сайте [caniuse.com](http://caniuse.com).

## Обобщенное свойство background

Для ряда блоков вам может понадобиться сразу несколько свойств группы background. Для того чтобы сократить запись css-свойств имеет смысл использовать

универсальное, или обобщенное свойство background. Синтаксис его таков:

```
background: background-image ||  
          background-position/background-size ||  
          background-repeat ||  
          background-attachment ||  
          background-origin ||  
          background-clip ||  
          background-color
```

В этом свойстве важно знать значение составляющих его свойств по умолчанию:

```
background: none || top left/auto || repeat ||  
          scroll || padding-box || border-box ||  
          transparent
```

В файле *examples/background/bg.html* использован такой код для элемента с фоном:

```
.example {  
  padding: 50px;  
  background: url(bg/golden-paper.jpg) center/cover  
             no-repeat gold;  
}
```

В нем перечислены не все свойства группы background, а только те, которые необходимы для данного div. Все остальные свойства принимают значения по умолчанию. Внешний вид этого примера изменяется в зависимости от размера открытого окна браузера, т.к. в составе обобщенного свойства background использовано свойство background-size: cover.



Рисунок 47

Увидеть значение всех остальных свойств можно в Инспекторе свойств (клавиша F12) на вкладке Computed.

Styles	Computed	Event Listeners	DOM Breakpoints	Properties	Accessibility
Filter					<input type="checkbox"/> Show all
► background-attachment	scroll				
► background-clip	border-box				
► background-color	rgb(255, 215, 0)				
► background-image	url( <a href="http://127.0.0.1:1640/bg/golden-paper.jpg">http://127.0.0.1:1640/bg/golden-paper...</a> )				
► background-origin	padding-box				
► background-position-x	50%				
► background-position-y	50%				
► background-repeat-x					
► background-repeat-y					
► background-size	cover				

Рисунок 48

Примечание: можно использовать сначала обобщенное свойство `background` с перечислением ряда свойств, а затем отдельной строкой записывать дополнительное свойство, но нельзя делать наоборот — сначала записать какое-либо из свойств, а затем записать `background`. В этом случае все неуказанные свойства будут сброшены до значений по умолчанию.

В примере (файл `examples/background/bg2.html`) сначала заданы обобщенное свойство `background` для класса `bg`, а затем добавлены разные значения свойства `background-image` для классов `.bg1` и `.bg2`:

```
.bg {  
    background: no-repeat center/cover #0a4c8e;  
    padding: 100px 50px;  
    background-image: url(bg/sky-bg.jpg)  
}
```

Для последнего блока с классом my-bg просто поменяем порядок свойств:

```
.my-bg {  
    background-image: url(bg/sky-bg.jpg);  
    background: no-repeat center/cover #0a4c8e;  
    padding: 100px 50px;  
}
```

В результате таких действий мы увидим внизу текст на синем фоне без картинки, т.к. по умолчанию значение свойства background-image: none.

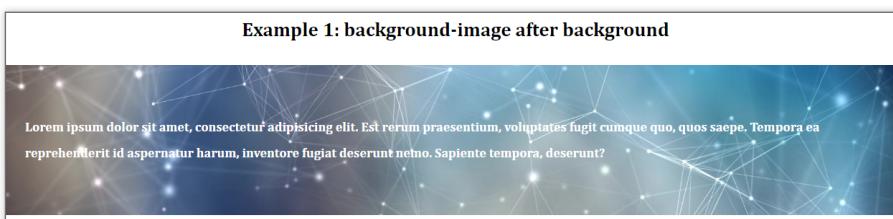


Рисунок 49

Именно такое значение показывает нам на вкладке Computed Инспектора свойств.

► background-attachment	scroll
► background-clip	border-box
► background-color	■rgb(10, 76, 142)
► background-image	none
► background-origin	padding-box
► background-position-x	50%
► background-position-y	50%
► background-repeat-x	
► background-repeat-y	
► background-size	cover

Рисунок 50

Ссылки по теме:

- [https://www.w3schools.com/cssref/css3\\_pr\\_background.asp](https://www.w3schools.com/cssref/css3_pr_background.asp);
- <https://developer.mozilla.org/en/docs/Web/CSS/background-clip>;
- <https://codepen.io/Jintos/pen/crlxk>.

## Множественные фоны

В стандарте CSS3 стало возможным добавление нескольких фонов к одному и тому же html-элементу. Для этого в каждом из свойств группы background нужно поставить запятую и написать нужное значение. Рассмотрим пример, в котором нам необходимо разместить внизу после основного текста 3 изображения детей + основной фон для всего текста:

```
#content {
    background-image: url(bg/baby-girl-crying.png),
                      url(bg/baby-girl-drinking.png),
                      url(bg/baby-girl-laughing.png),
                      url(bg/tetr5.gif);
    background-repeat: no-repeat,no-repeat,no-repeat,
                      repeat;
```

```

background-position: left bottom, center bottom,
                    right bottom, left top;
}

```

Обратите внимание на свойство `background-image`: в нем сначала размещены изображения с детьми, и только в самом конце — повторяющийся фон в виде тетрадки в клетку. Если бы мы сделали наоборот, клетчатый фон перекрыл бы изображения детей, и смысла в их добавлении бы не было. Поэкспериментируйте со стилями самостоятельно.

Файл примера *many-bg.html* из папки *examples* выглядит так:

#### Как играть с ребенком

Четырехмесячный ребенок превращается из пассивного наблюдателя окружающего мира в активного участника событий. Теперь, умев доставать и хватать предметы, он ищет любую возможность использовать приобретенные навыки. Если двигающаяся игрушка висит над кроваткой, но до нее нельзя дотянуться, малыш расстраивается. Ему хочется ударить, схватить ее, положить в рот и посмотреть, что из этого получится.

Если понаблюдать за тем, как ребенок играет с движущейся игрушкой, можно обнаружить, что он делает это, следуя определенному порядку. Он хватает ее за кольцо, отпускает, смотрит, как игрушка вернется, и снова пытается ее схватить. Та последовательность действий - схватить, осмотреть и снова схватить - относится уже не только к игрушке, висящей над кроваткой, ребенок также ведет себя по отношению и к другим вещам. Занимаясь с каким-нибудь предметом, находящимся поблизости, малыш как бы задает себе вопрос: "Что я держу в руках и что с этим можно сделать?"

В этом возрасте дети начинают активно изучать окружающую обстановку и одновременно проявлять инициативу в общении с людьми. Вместо того чтобы ждать, когда кто-нибудь обратит на него внимание, ребенок, как только увидит входящего, первым начинает "гузить" и лепетать. И поскольку нет ничего привлекательнее улыбающегося, лепечущего малыша, вы не сможете не ответить на его приветствие.



Рисунок 51

Вы можете использовать для назначения множественных фонов сокращенную форму записи с помощью свойства `background`. Причем, если вам нужны значения по умолчанию, вы просто не указываете их.

В том же файле `many-bg.html` из папки `examples` второй блок текста оформлен аналогично первому, но с помощью обобщенного свойства `background`:

```
#dream {  
    background: url(bg/baby-girl-sleeping.png)  
              no-repeat bottom right,  
              url(bg/tetr5.gif) 0% 0%;  
}
```

**Здоровый детский сон**

Вырастить одного-единственного ребенка и превратить выращивание в тяжкий труд, себя в мать-героиню, а жизнь семьи в подвиг — совсем не сложно. Для этого всего-навсего надо не высыпаться.

Поэтому поймите и запомните главное: здоровый детский сон — это такой сон, когда сладко и комфортно всем — и взрослым, и детям!

Организация детского сна — это организация здорового сна всех членов семьи.

Самое простое в вышеупомянутой организации — приобрести качественный одноразовый подгузник, и это, по сути, единственный друг детского сна, который реально существует без вашего родительского участия — его за вас и для вас уже придумали и сделали.

Все остальное исключительно в ваших руках.

Ведь сон теснейшим образом связан с другими составляющими образа жизни — с питанием, прогулками, параметрами воздуха, одеждой, гигиеническими процедурами и т.д.

Правильное питание, игры на свежем воздухе, уборка помещений, купание, мягкая, чистая, сухая постель — все это требует времени, желания, знаний, умений.



Рисунок 52

Обратите внимание, что для обоих div-ов в этом файле был задан padding-bottom: 270px, чтобы картинки размером 256x256px свободно поместились под текстом

```

margin: 20px auto;
border: 2px solid #ccc;
padding: 10px;
padding-bottom: 270px;
}
#content {
background: url(bg/baby-girl-crying.png), url(bg/baby-girl-drinking.png),
url(bg/tetr5.gif);
background-size: no-repeat, no-repeat, repeat;
background-position: top center, bottom right, left top;
}
#dream {
background: url(bg/baby-girl-sleeping.png) no-repeat bottom right,url(bg/tetr5.gif) 0% 0%;
}

```

Рисунок 53

## Верстка PSD макета

Верстка предполагает наличие некоего дизайна. Можно, конечно, верстать, используя PNG или JPG-изображение, но тогда вы столкнетесь с проблемами копирования текста и сохранения изображений. Кроме того, там не будет такого удобного инструмента, как направляющие, который позволяет измерить расстояния между блоками.

На данный момент дизайнеры предпочитают работать с такими программами, как Adobe XD, Adobe Photoshop, Figma, Sketch, Affinity Designer (Photo) и др. Выбор программы обычно зависит от предпочтений конкретного дизайнера или программного обеспечения, приобретенного и установленного в дизайн-студии. Поскольку использование Adobe Photoshop пока еще остается очень популярным среди дизайнеров, да и в сети Интернет можно скачать большое количество бесплатных PSD-макетов для наработки опыта верстки, мы сегодня рассмотрим, каким образом можно сохранять (экспорти-

тировать) изображения, рассчитывать отступы и копировать текст из PSD-файла.

Макет был [скачан с одного из сайтов](#), предлагающих ряд бесплатных psd-макетов для верстки. В данной части урока мы разберем верстку только половины макета (на скриншоте ниже). Остальное вам придется сделать дома.

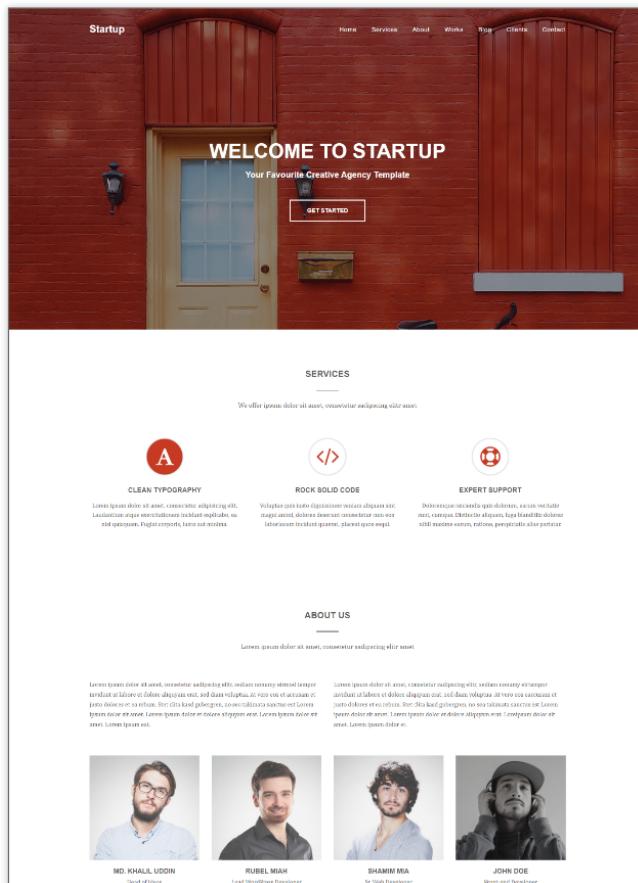
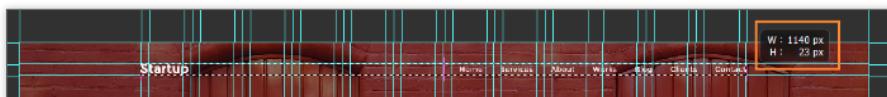


Рисунок 54

Для начала проведем небольшой анализ блоков сайта:

- Ширина макета 1600px, но в середине с помощью направляющих выделен контейнер шириной 1140px, который ограничивает весь контент по ширине. Для него создадим класс .container.



**Рисунок 55**

- Цвет шрифта в шапке сайта — белый, а во всех остальных блоках — #555.
- Основной шрифт на странице — с засечками, а для заголовков — без засечек. Размер основного шрифта (font-size) 14px, высота строк (интерлиньяж, line-height) 24px.
- Большая часть заголовков имеет символы в верхнем регистре, заголовки разделов имеют линию ниже текста, которая при выделении слоя с ней с CTRL имеет размеры 50x2px.
- Часть колонок имеет центральное выравнивание текста, для чего имеет смысл добавить отдельный класс .text-center.
- Информация в макете разбита на 2, 3, 4 одинаковые по размеру колонки, и только в блоке Resent Blog Posts на левую часть приходится 5 из 12 колонок (41.67%), а на правую — 7 из 12, т.е. 58.33%.

## Экспорт изображений

В верхней части макета мы видим 3 слоя, которые управляют фоновым изображением для шапки сайта.

Для начала необходимо склеить эти 3 слоя, выделив их и нажав **CTRL+E**. А затем выполнить экспорт изображения в формат *JPG*.



Рисунок 56

Размеры изображения будут размерами нашего header-a. Если по ширине (1600px) изображение может быть несколько меньше современных экранов, то по высоте мы будем использовать 770px.

Примерно аналогичную операцию необходимо выполнить для экспорта изображений из блока Services и About Us. Только в этом случае сложно обойтись без склеивания слоев, выполнив экспорт группы (папки) слоев в изображение в формат PNG или PNG-8.

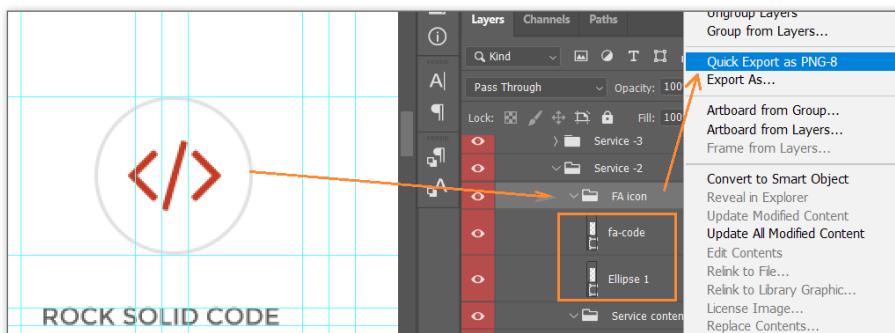


Рисунок 57

Для первого фото блока About Us необходимо отключить видимость слоев с серым прямоугольником и блоком со стрелкой влево.

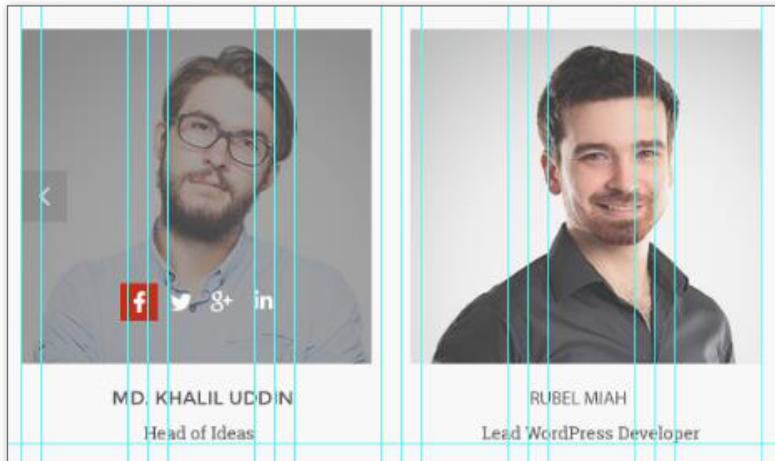


Рисунок 58

После этого можно выполнить экспорт группы слоев в формат JPG, т.к. прозрачность здесь не нужна.

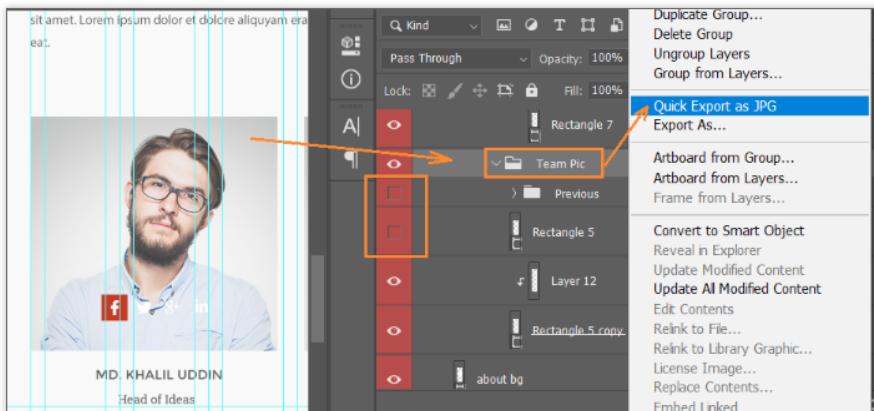


Рисунок 59

## Копирование текста

Для создания html-разметки нам понадобится также текст каждого блока, а также его характеристики. Поскольку шрифтов, которые использованы в макете, у нас нет, и подключать мы их не будем, то у нас есть 2 варианта “добытия” текста.

1. Скопировать его из названия слоя



Рисунок 60

2. Сделать двойной клик на иконке Т с треугольником, скопировать текст из блока и посмотреть его характеристики в палитре Character: размер шрифта, межстрочное расстояние, жирность и регистр символов.



Рисунок 61

## Измерение отступов

Отступы можно измерять инструментом прямоугольного выделения, обращая внимание на высоту (H) выделенного фрагмента или на его ширину (W). Здесь

вы оцените наличие направляющих, т.к. выделение автоматически притягивается к ним.

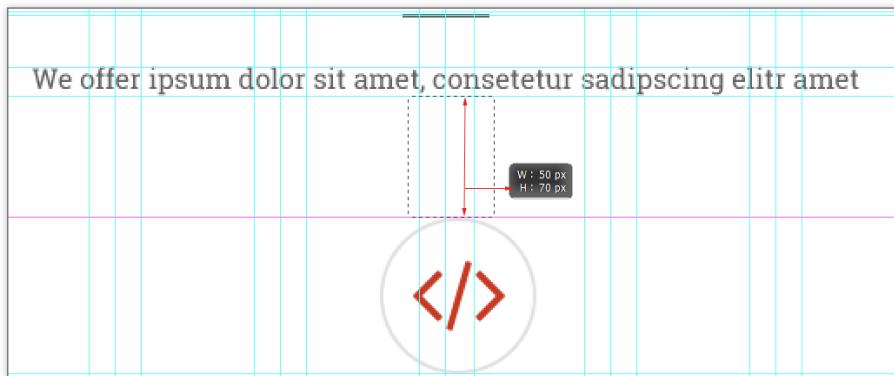


Рисунок 62

Также вы можете выделить в макете 2 и более объекта с зажатой клавишей Shift, а затем зажать клавишу Ctrl и посмотреть на величины отступов. Минус этого способа — не всегда с первого раза вы получите величины нужных вам отступов.

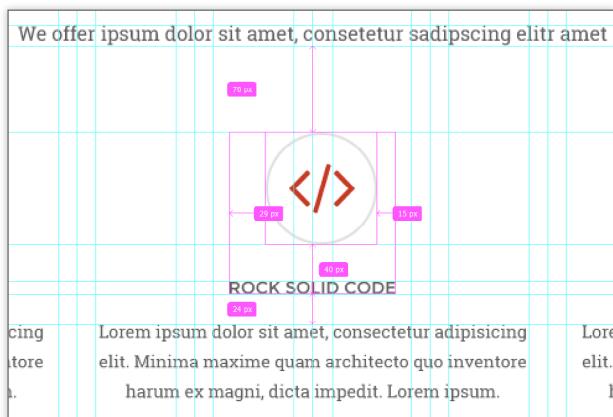


Рисунок 63

Отступы до и после текста придется корректировать из-за величины высоты строки, которая в CSS также сформирует некий отступ выше и ниже текстового блока.

## Разметка шапки сайта

```
<header class="text-center">
  <div class="container">
    <div class="row">
      <div class="column logo">Startup</div>
      <nav class="column menu">
        <a href="#">Home</a>
        <a href="#services">Services</a>
        <a href="#about-us">About</a>
        <a href="#">Works</a>
        <a href="#">Blog</a>
        <a href="#">Clients</a>
        <a href="#">Contact</a>
      </nav>
    </div>
  </div>
  <h1 class="text-uppercase">Welcome to STARTUP</h1>
  <h2>Your Favourite Creative Agency Template</h2>
  <a href="#services"
    class="btn-outline">Get Started</a>
</header>
```

Общие стили для страницы и для шапки:

```
* { box-sizing: border-box }
body { margin: 0;
      font-family: Cambria, serif;
      font-size: 14px;
      color: #555;
      line-height: 24px;
      position: relative; }
```

```
.container { width: 1140px; margin: auto; }
.text-center { text-align: center; }
header { height: 770px;
    background: url(..../images/top-bg.png)
        no-repeat #6c1d14;
    background-size: cover;
    color: #fff;
    padding-top: 37px; }

h1, h2, h3, h4, h5 { font-family: sans-serif;
    text-transform: uppercase; }

h1 { padding-top: 233px;
    font-size: 3rem;
    margin-bottom: 31px; }

header h2 { font-size: 20px;
    font-family: Cambria, serif;
    margin: 0 0 65px;
    text-transform: none; }

.btn-outline { text-transform: uppercase;
    font-weight: bold;
    text-decoration: none;
    font-family: sans-serif;
    background-color: transparent;
    border: 2px solid #fff;
    padding: 16px 40px;
    color: #fff;
    transition: .5s; }

.btn-outline:hover,
.btn-outline:focus { background-color: #fff;
    color: #6c1d14; }

.logo { font-size: 24px;
    font-weight: bold;
    font-family: sans-serif; }
```

```
/*-- Menu --*/
.menu {
    text-align: right;
    float: right;
    margin: 2px 0;
    font-family: sans-serif; }

.menu a { display: inline-block;
    color: #fff;
    text-decoration: none;
    letter-spacing: 1px;
    margin-left: 32px; }

.menu a:hover { text-decoration: underline; }
```

## Раздел Services

Раздел Services начинается с заголовка с линией и имеет отступы снизу и сверху, а также 3 колонки с изображением, заголовком и текстом с выравниванием по центру. Сделаем разметку.

```
<section id="services">
    <div class="container">
        <h2 class="line-header">Services</h2>
        <p class="intro-text text-center">We offer
            ipsum dolor sit amet, consetetur sadipscing
            elitr amet
        </p>
        <div class="row">
            <div class="column-3 text-center service">
                
                <h3 class="text-center">Clean Typography</h3>
                <p>Lorem ipsum dolor sit amet, consectetur
```

```

adipisicing elit. Laudantium atque
exercitationem incident explicabo, ea nisi
quisquam. Fugiat corporis, iusto aut minima.
</p>
</div>
<div class="column-3 text-center service">
    
    <h3 class="text-uppercase">Rock Solid Code</h3>
    <p>Voluptas quis iusto dignissimos veniam
        aliquam sint magni animi, dolores deserunt
        consectetur cum eos laboriosam incident
        quaerat, placeat quos sequi.
    </p>
</div>
<div class="column-3 text-center service">
    
    <h3 class="text-center text-uppercase">
        Expert Support
    </h3>
    <p>Doloremque reiciendis quis dolorum, earum
        veritatis sunt, cumque. Distinctio aliquam,
        fuga blanditiis dolores nihil maxime earum,
        ratione, perspiciatis alias pariatur.
    </p>
</div>
</div>
</div>
</section>
```

Стили:

```

.row { margin-left: -15px; margin-right: -15px; }
.row::before, .row::after { content: '';
    display: table; }
.row::after { clear: both; }
```

```
[class^="column"] {
    padding-left: 15px;
    padding-right: 15px;
    float: left; }
.column-2 { width: 50%; }
.column-3 { width: 33.33%; }
.column-4 { width: 25%; }
section { padding: 94px 0 100px; }
.line-header { text-align: center; margin: 0; }
.line-header::after {
    content: '';
    display: block;
    width: 50px;
    margin: 25px auto;
    border-bottom: 2px solid; }
.intro-text { font-size: 16px; margin-bottom: 70px; }
.service h3 { font-size: 1em; margin: 12px 0 14px; }
.service p { line-height: 1.6; }
```

Класс .row нужен для отмены обтекания, схлопывающего высоту родительского элемента для колонок с классом .column- со свойством float: left.

Отступы для всех элементов взяты с макета и подкорректированы в соответствии с высотой строк элементов.

## Раздел About Us

В этом разделе текстовые блоки вверху размещены в 2 колонки, а информация о команде — в 4, поэтому в разметке использованы разные классы: .column-2 и .column-4 внутри класса .row:

```
<section id="about-us">
    <div class="container">
        <h2 class="line-header">About Us</h2>
```

```
<p class="intro-text text-center">Lorem ipsum  
dolor sit amet, consetetur sadipscing elitr  
amet  
</p>  
<div class="row">  
    <div class="column-2">  
        Lorem ipsum dolor sit amet, consetetur  
        sadipscing elitr, sediam nonumy eirmod ...  
        Lorem ipsum eat.  
    </div>  
    <div class="column-2">  
        Lorem ipsum dolor sit amet, consetetur  
        sadipscing elitr, sediam nonumy eirtempor ...  
        dolor et.  
    </div>  
</div>  
  
<div class="row mt-team">  
    <div class="column-4 text-center team-member">  
          
        <h3 class="text-uppercase">  
            Md. Khalil Uddin  
        </h3>  
        <p>Head of Ideas</p>  
    </div>  
  
    <div class="column-4 text-center team-member">  
          
        <h3>Rubel Miah</h3>  
        <p>Lead WordPress Developer</p>  
    </div>  
  
    <div class="column-4 text-center team-member">  
        
```

```
<h3>Shamim Mia</h3>
<p>Sr. Web Developer</p>
</div>
<div class="column-4 text-center team-member">
    
    <h3>John Doe</h3>
    <p>Front-end Developer</p>
</div>
</div>
</div>

</section>
```

Часть стилей для заголовков и section у нас уже есть, остальные предполагают отступы между элементами (класс .mt-team), а также использование псевдокласса :hover для изображений, чтобы создать эффект затенения фото с помощью [css-свойства filter](#).

```
.mt-team { margin-top: 57px; }
.team-member img { transition: filter .5s; }
.team-member img:hover { filter: brightness(70%); }
.team-member h3 { font-size: 1em; margin: 6px 0 2px; }
```

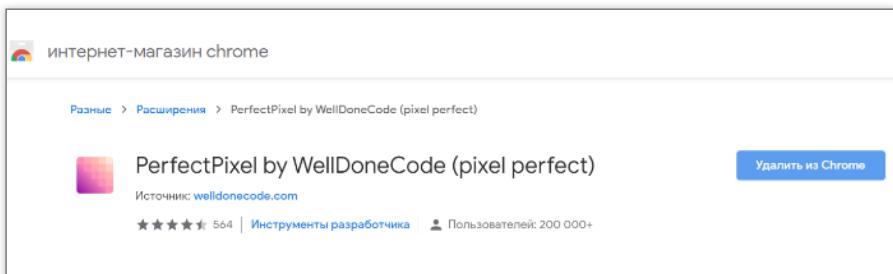
### Проверка верстки на соответствие макету

Хороший PSD-макет предполагает, что между всеми похожими элементами (разделами, заголовками, колонками) дизайнер делает одинаковые отступы и таким образом помогает верстальщику обходится минимумом классов и стилей. В предлагаемом макете это условие в достаточной мере соблюдено. Обратите внимание, что система на-

правляющих в PSD-файле очень четко расположена, имеет одинаковые отступы по вертикали + много горизонтальных направляющих, облегчающий процесс измерения расстояний и размеров. Однако так в вашей практике будет не всегда. Да и дизайнеры для создания макетов не ограничиваются только одним Adobe Photoshop. Достаточно часто бывает так, что отступы у многих элементов разные, настройки шрифтовых блоков тоже. Все достаточно благополучно обычно только с цветовой схемой.

Поэтому неплохо было бы иметь инструмент, позволяющий понять, насколько вы «попали» в макет, т.е. определить визуально, находятся ли дизайнерские элементы на своих местах в верстке. Для этого можно использовать такое расширение для Chrome и Firefox, как [Perfect Pixel](#) или аналогичные.

Вы устанавливаете плагин из каталога расширений браузера.



**Рисунок 64**

Далее необходимо загрузить изображение всего вашего макета или его части, указать прозрачность и расположение изображения макета по X и Y (0, 0), а также масштабирование (0.8 в нашем случае).

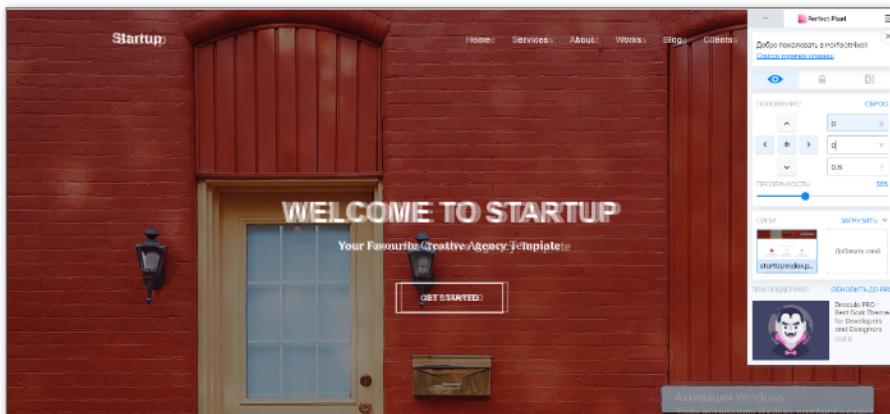


Рисунок 65

Затем вы проверяете с помощью инспектора кода, насколько ваши элементы совпадают с дизайном, добавляя или переопределяя стили.

Плюсы такого плагина — вы управляете прозрачностью и положением скриншота, а также можете отключить его нажатием на кнопку с глазиком. Минусы заключаются в том, что при разных значениях разрешения вашего монитора и использованного в макете горизонтального разрешения/размера вам придется подгонять масштабирование изображения макета под размеры браузера в вашем мониторе.

Второй вариант — вы можете создать отдельный элемент в вашей разметке, например:

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

В файле *style.css* нужно добавить такие стили:

```
body { ...;
    position: relative; }
```

```
.screenshot { height: 100%; width: 100%;  
position: absolute; top: 0; left: 0; z-index: 200;  
opacity: .5; background: url(../HW+Layout+PSD/  
StartUp%20Layout.jpg) center top;  
}
```

В результате этих действий над элементами вашей верстки появляется блок с фоновым изображением, которое размещается сверху и центрируется относительно браузера. Это плюс, т.к. подгонять размеры не нужно, а сместить изображение можно с помощью свойства background-position.



Рисунок 66

Минус этого подхода, в том, что у вас нет под рукой ползунка с изменением прозрачности, вы не сможете отключить этот элемент, нажав на иконку глазика — для этого нужно править html- или css-код.

Ссылки:

- [Perfect Pixel для Chrome](#);
- [Perfect Pixel для Firefox](#).

## Домашнее задание

Вам необходимо будет доверстать макет, используя за основу разметку и стили из папки *startup-page* и файл *PSD*. Разделы Контактов и отзывов верстать не нужно, т.к. элементы форм мы еще не рассматривали, а отзывы предполагают использование JavaScript или jQuery-плагина.

Экспортировать изображения вам нужно будет самим. Обратите внимание на первое изображение из блока Latest Works. Вам нужно будет отключить группу слоев с именем *Work*, чтобы экспортировать изображение. Делать эффект появления красного полупрозрачного фона при наведении на фото необязательно, т.к. это предполагает некоторые дополнительные знания по позиционированию элементов или их трансформации.

Все остальные изображения экспортируйте, как группу слоев.

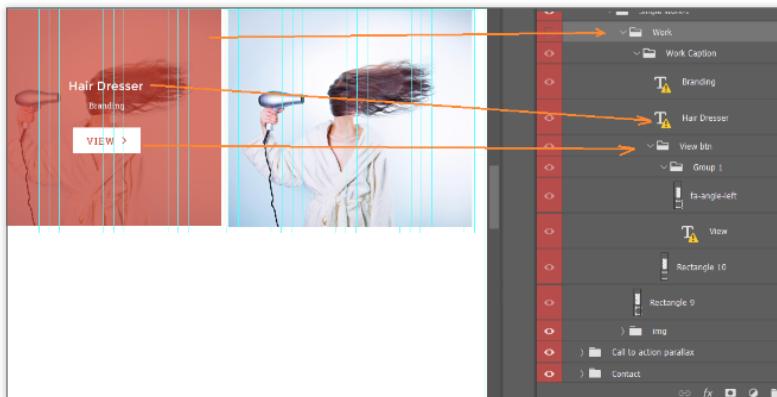


Рисунок 67

Скриншот задания вы найдете ниже и в папке *startup-page/HW+Layout+PSD*.

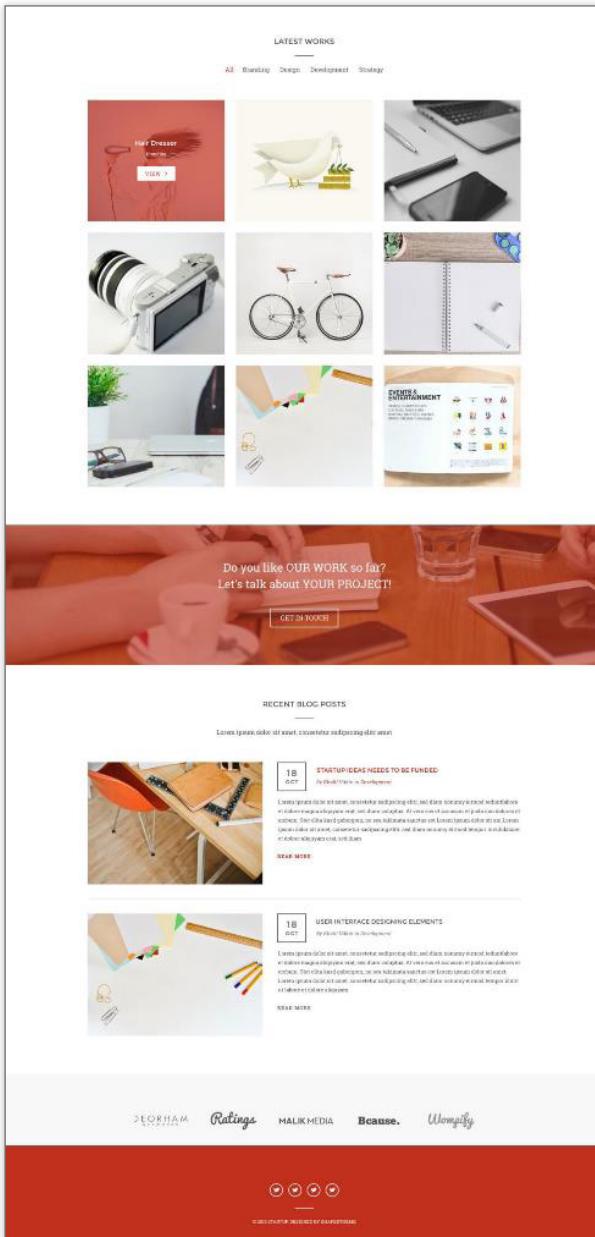


Рисунок 68