

HyperText Markup Language

HTML & CSS

Lesson 4

Graphics in Web Design

Contents

Graphics in Web Design.....	4
Images on the Page. The Tag	4
Attributes of the Tag	10
CSS Properties for Images	13
Image File Formats on the Web	16
Layout of an Article with Images	20
The figure and figcaption Tags	31
Related links:	33
Properties of the background Group	33
The background-position Property.....	42
The background-size Property	44
The background-attachment Property	46
The background-origin Property	48
The background-clip Property.....	48
The Generalized background Property	50

Contents

Multiple Backgrounds.....	53
Export Images	59
Copy Text.....	61
Measure Indents	62
Website Header Markeup	64
The Services Section.....	66
About Us Section	68
Checking the Layout for Compliance with the Layout	70
Homework	74

Lesson materials are attached to this PDF file. In order to get access to the materials, open the lesson in [Adobe Acrobat Reader](#).

Graphics in Web Design

Images on the Page. The Tag

Graphics, namely, images added to the page, greatly beautify a website. It is very hard to imagine a web page without them. Surely, you missed images in the examples and homework assignments in the previous lessons. Let's fix this situation.

Images are added to html using the `` tag. This tag has 2 required attributes: `src` and `alt` which you must always specify. The syntax for adding an image to a page is as follows:

```

```

For instance,

```

```

The `src` attribute specifies the image source, that is, file location.

As a rule, all images for the site are placed in a folder with traditional name *images* or *img*, or less often — *assets* so that the structure of the site is ordered. We will also adhere to this rule. The `src` attribute specifies a *relative path* to the image. In this case, the term “relative path” means image file location relative to the HTML file. The images folder lies in the same directory as *image.html*. Therefore, the path to the image file will include the folder name/file name and extension.



Figure 1

Code editors usually prompt the image name after you specify the name of the image folder in `src` quotes:

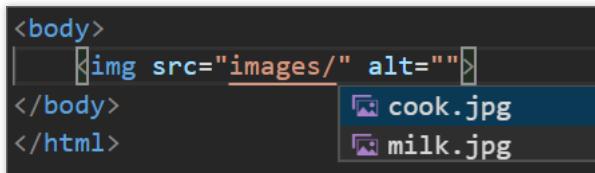


Figure 2. Visual Studio Code

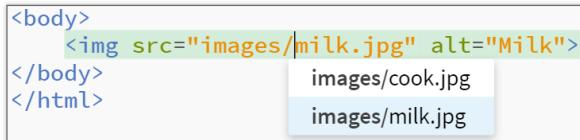
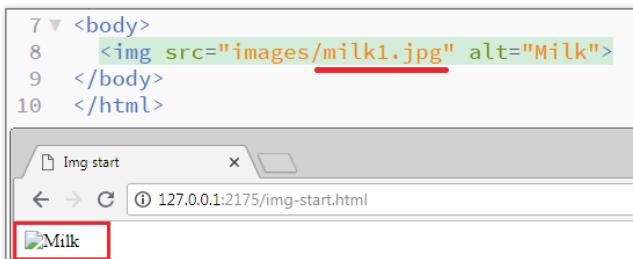


Figure 3. Brackets

You can use the `Enter` key to confirm the selection and the down (\downarrow) or up (\uparrow) arrow keys to move to the next/previous file in the folder. `Ctrl + space` in some editors allows you to list the files in the folder.

The `alt` attribute is necessary to specify *alternative text* that will appear if the path to the image is specified incorrectly, the image file has an unsupported format or has been deleted from the server. This attribute is very important for crawlers and screen readers, which help visually impaired people navigate the site, therefore it is mandatory. You also need to indicate information that corresponds to what is shown in the picture in it.

Figure 4 shows that the code tries to use the image named “*milk1.jpg*” that does not exist in the *images* folder, so the browser shows the text from the *alt* attribute.



The screenshot shows a code editor at the top with the following HTML code:

```
7 ▼ <body>
8   
9   </body>
10  </html>
```

Below the code editor is a browser window titled "Img start". The address bar shows the URL "127.0.0.1:2175/img-start.html". The main content area of the browser displays a small image placeholder with the word "Milk" next to it, indicating that the intended image file "milk1.jpg" was not found.

Figure 4

As for the insertion of images into the content of the page, when they are added to the text of the paragraph, they will be positioned with their bottom edge along the bottom lines of the text. Figure 5 clearly shows that the paragraph text begins right after the image, “sticking” to it at the very bottom.

Image alignment

Lorem 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.*

Figure 5

You can change the text position by using the CSS property **vertical-align** for the image with its values set to **top**, **middle**, **baseline**, or **bottom** (default), although it is unlikely to brighten up the page (Fig. 6).

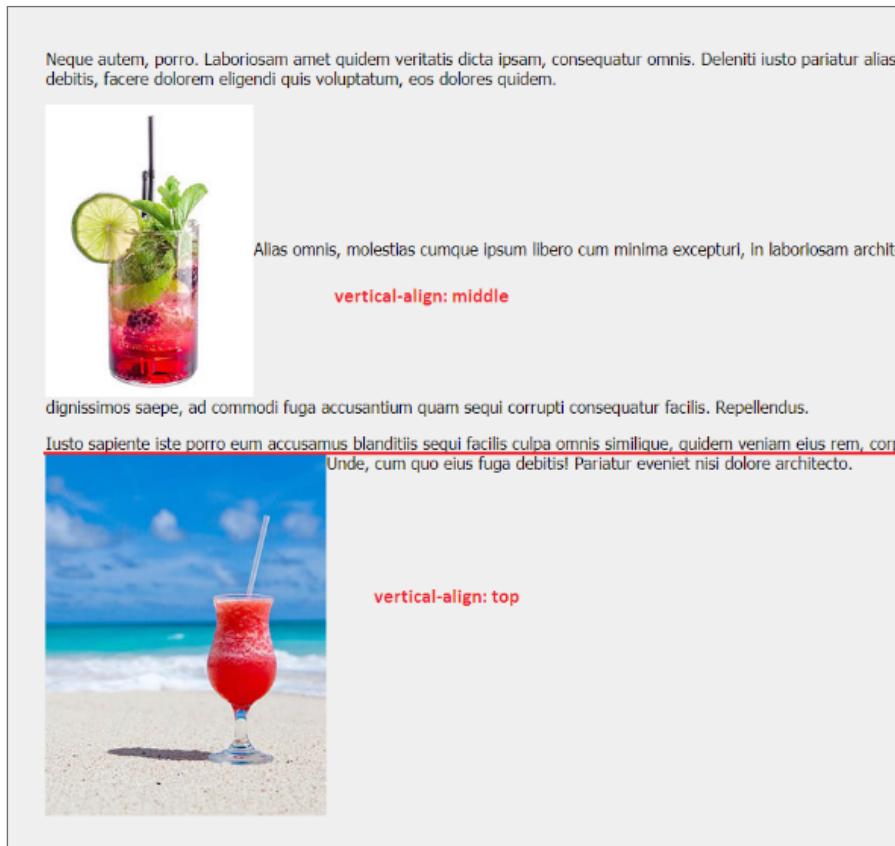


Figure 6

However, you can use the **vertical-align** property to align a small image with text.

Most often, images are aligned with the **float** property and its values **left** or **right** to have the text flow around them

to the right or left. This property should now be used instead of the deprecated `align` attribute, but it requires margins to the right or left so that the text does not “stick” to the edge of the image. These properties are usually written for classes, and classes are added to various images as attributes:

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

These classes (Fig. 7) have been applied to the two images at the bottom. As a result, the text is arranged in a more familiar way. And the value of the `vertical-align` does not matter for these images.



Figure 7

If you need to place the image in the center, you can either nest it into a block element with `text-align: center` or turn an inline element into a block one using the `display: block` property and set `margin: auto`.

```
.text-center { text-align: center; }  
<div class="text-center">  
      
</div>
```

Img in div with `text-align: center`



Figure 8

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

```

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



Figure 9

All examples can be found in the *image-alignment.html* file of the *examples* folder.

Attributes of the Tag

The tag can also contain other attributes, namely:

1. **width** — image width in px (no quotes).
2. **height** — image height in px (no quotes).
3. **sizes** — image sizes for different page layouts and screen resolutions. It is used along with media queries, so it will be covered in later lessons.
4. **srcset** — allows specifying one or several comma-separated URLs that indicate a set of images to be displayed in the browser based on the pixel density of the device (regular screens and retina). The attribute should specify:
 - ▷ image address;
 - ▷ image width as a positive integer followed by **w** (e.g., **600w**);
 - ▷ pixel density as a positive decimal integer followed by **x** (**1x** is by default, you can specify **2x**).

For instance,

```

```

- **ismap** — specifies that the image is a server-side image map that allows you to bind links to various areas of the same image;
- **usemap** — binds an image to an image map set with the <map> container.

The latter two attributes are now used extremely rarely, so we will not dwell on them.

The attributes **width** and/or **height** are most commonly used. They allow speeding up page loading because they im-

mediately tell the browser how much space it needs to reserve for each image after the browser receives it. These attributes can match the actual image sizes, or you can make them smaller or bigger. In the latter case, the image will be stretched, which most often looks ugly in the browser.

If you open the file *image-size.html* in the *example* folder, you will find text with several images. The first one was cut in half by changing the **width** and **height** attributes.

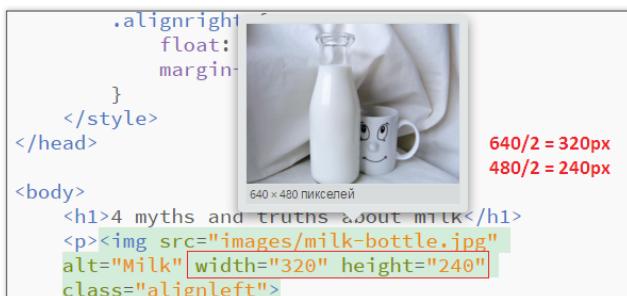


Figure 10

The image was inserted with its natural dimensions specified in the **width** and **height** attributes.



Figure 11

If you mix up the values in the attributes or set incorrect proportions of the image, it will be distorted in the browser - extended vertically (see Figures 12 and 13).

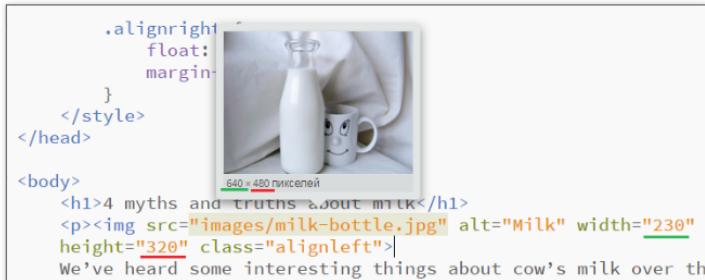
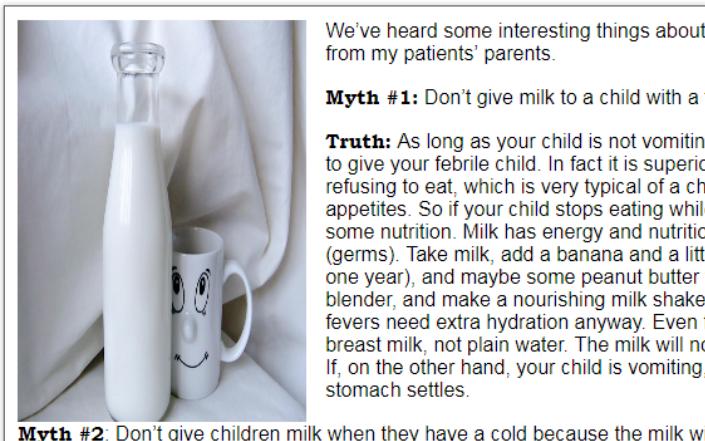


Figure 12



Myth #2: Don't give children milk when they have a cold because the milk wi

Figure 13

- **Note:** we will not consider attributes such as *align*, *border*, *hspace*, *vspace*, *longspace*, because they are deprecated in HTML5. Please do not use them in your homework because they are currently replaced by CSS properties.

Related links:

1. https://www.w3schools.com/tags/tag_img.asp;
2. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img>.

CSS Properties for Images

Deprecated attributes were replaced by CSS properties that have more values and allow for better image formatting. For example, you can use different border options using the `border` property:

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

```

Result:



Figure 14

You can also use `border-radius` and `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); }
```

```

...

```

The result of applying class styles:



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 own mother cringes when I give my 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.

Figure 15

Also, various hover effects are often applied to images when they are hovered by a mouse cursor. This is done with the `:hover` pseudo-class:

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

```
.effect:hover {  
    opacity: .5;  
    cursor: pointer; }  

```

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 good, absorbable fluid to give your febrile child. In fact it is superior plain water if your child is refusing to eat, which is very typical of a child with a fever. Fevers take away appetites. So if your child stops eating while she is sick, at least she can drink some nutrition. Milk has energy and nutrition, which help fight infection (germs). Take milk, add a banana and a little 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 teetle infants need formula or breast milk, not plain water. Then stick to clear fluids until their 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 own mother cringes when I give my 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 when you have a cold. Then again, my grandmother also believed that you only digest 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.

Truth: Actually, while milk is healthy and provides necessary calcium and vitamin D, too much milk can be a bad thing. To get enough calcium from milk, your child's body needs somewhere between 16 to 24 ounces of milk per day. Of course, if your child eats cheese, yogurt, and other calcium-containing foods, she



Figure 16

In the first case, the `:hover` pseudo-class is applied to the `shadow` class that changes the direction of the shadow when hovered. In the second case, the `effect` class has no rules, they are described only for the `:hover` pseudo-class, in which the opacity changes to 50% (`opacity: .five`), and the cursor appearance (`cursor: pointer`) changes from the standard arrow

to a hand (Fig. 16). You can read about the types of cursors in the [related article](#).

Image File Formats on the Web

In the previous lesson, we looked at several saving (exporting) options in various formats. Graphics software products allow you to save images in a wide variety of formats: *PSD*, *AI*, *CDR*, *TIFF*, *SKETCH*, *XD*, but not all of them can be used on HTML pages. The main requirements for image files on the web are lightweight and good quality. Therefore, HTML pages use the following image formats:

- **JPEG** (named after the developer organization, *Joint Photographic Experts Group*) is a format that supports millions of colors. The main extension is *JPG*, although you may come across *JFIF*, *JPE*, or *JPEG*. This format is, first of all, designed for realistic photographic images, including gradients. It supports lossy compression, so files can be lightweight at a high resolution. It does not support animation or transparency. When saving images in this format, you need to select a quality (0-100%) depending on which the file size will be larger (**80-100%**) or smaller (**50-70%**). When saving images with low quality, compression leads to artifacts that are absent in the original image.
- **GIF** (*Graphics Interchange Format*) was a very popular format on the web because it allowed you to save images with transparency (logos) and animation (avatars, greeting cards). Pixel translucency (alpha channel technology) is not supported, so it is not suitable for images with shadow. It has limitations in the number of colors used: a maximum of 256 colors indexed by *GIF*, so images with gra-

dients and many color transitions are not for this format. But solid colors are retained very well. Animated images with a large size in pixels can be really heavy in kilobytes. GIF format also allows interlaced data storage. It means that the lines are split into groups, and the image gradually appears when loaded — horizontally, line by line.

- **PNG (Portable Network Graphics)** is a format that performs lossless compression. It was created to replace GIF. It occurs in 3 variants:
 - ▷ **PNG-8** (similar to GIF in settings) allows saving an indexed image with a minimum of 256 colors, including transparent,
 - ▷ **PNG-24** can, in addition to the transparent color, store millions of colors, as JPEG images do,
 - ▷ **PNG-32** has not only millions of colors but also alpha channel support that allows it to store, for example, transparent areas with shadows.

As for the size of the saved file, it is necessary to compare files in both of these formats (PNG and JPEG) if transparency is not needed.



Figure 17

There is also the APNG format for animated PNGs, but it is not supported by all browsers yet, although it has good prospects (Fig. 17).

SVG (*Scalable Vector Graphics*) is the format for storing a vector and not bitmap (like the previous formats) image which is based on XML. It is close to Flash to some extent, but unlike Flash, it is recommended by the W3C organization and is open source. SVG is an interesting format that, on the one hand, can be edited in graphics editors such as Adobe Illustrator, Inkscape, or Sketch and, on the other hand, in text editors (Brackets, Sublime Text, Atom, Visual Studio Code, etc.). Its advantage is scalability, that is, an image is not distorted when resized because it is a set of vectors described by mathematical equations. Also, this format allows you to set special CSS properties and animations.

A WEBP format has appeared recently. It was developed and introduced by Google in 2010. It is based on the compression algorithm for static images (keyframes) from VP8 video codec and employs both lossy and lossless compression. Support by browsers is far from 100%, but it may become so in the future.



Figure 18

Browsers also support BMP and ICO, but they are highly specialized to the point that they are not used in a real layout. Unless that ICO is used for *favicons* (favorite icon) to display a site logo or part of it on the browser tab.

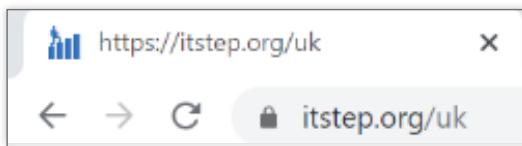


Figure 19

The link to a favicon is added in the `head` block of your page using the `link` tag:

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

As of today, images in JPG, PNG, SVG are used, and GIF has lost its former popularity and is mainly used for animations. You will find a number of examples with GIF animation in this folder: *examples/images/gifs*, and you can also see these images in action in *event-with-gif.html*.

Related links:

1. <https://developer.mozilla.org/en-US/docs/Web/SVG>;
2. 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;
6. <http://www.patrickhansen.com/2011/02/04/png-8-24-32-what/>.

Layout of an Article with Images

Let's use images to lay out an article that will look like a small advertising block. You will find a ready file named *article-with-images.html* in the *examples* folder and a stub to work with in the file named *article-with-images-start.html* so that you could repeat the steps described below. The appearance of the article is shown in Figure 20.

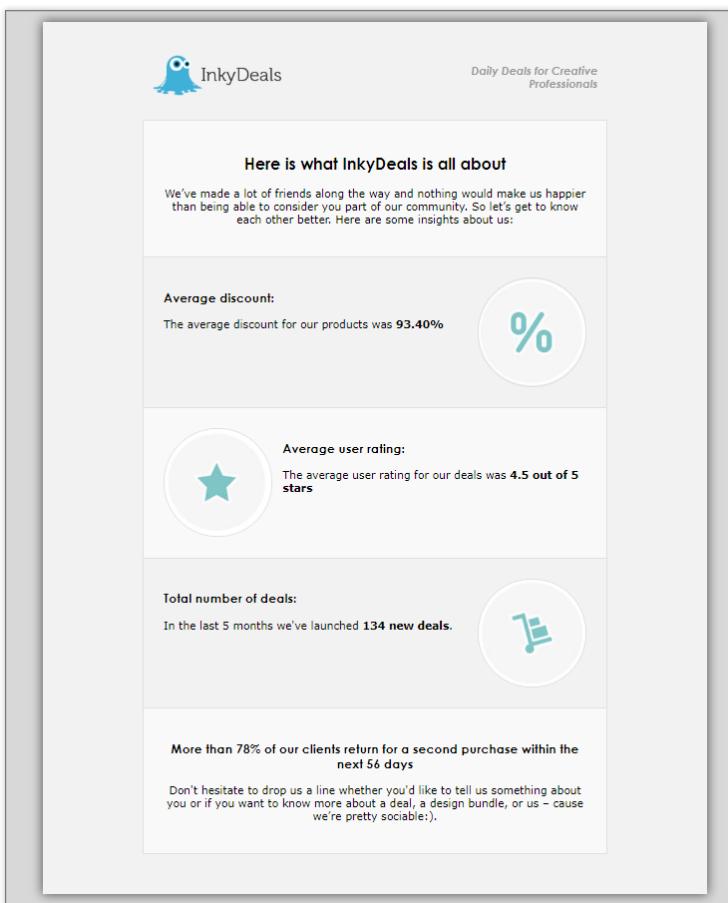


Figure 20

Let's begin with the markup:

```
<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>
```



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 o



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**

Figure 21

The markup has a root element `<article>` with the tags `<header>` and `<div id="main">` inside. There are several

divs with the `block` class inside the main `div`. The first and last of `<div class="block">` contain only the header and text, and the rest have images with the `alignleft` and `alignright` classes that are required to place images to the left and right of the text.

Before the styles are set, the page looks like this (Fig. 21):
Let's get down to CSS formatting.

```
* { 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);
}
```

All the rules should be familiar to you, so we will not dwell on them. Let's analyze only the first one: we set `box-sizing: border-box` for all elements so that the browser makes allowance for `borders` and `paddings` in the `width` of the element. If you take a look at the value `width: 1000px` in the `article`, this element should take up the bigger part of the screen, the `article`, however, also has fairly big `padding: 20px 150px 60px`. So, due to the `box-sizing: border-box` property, the width occupied by content is accounted for by $1000\text{px} - 150\text{px} \times 2 = 700\text{px}$.

The appearance of the page in the browser has changed:

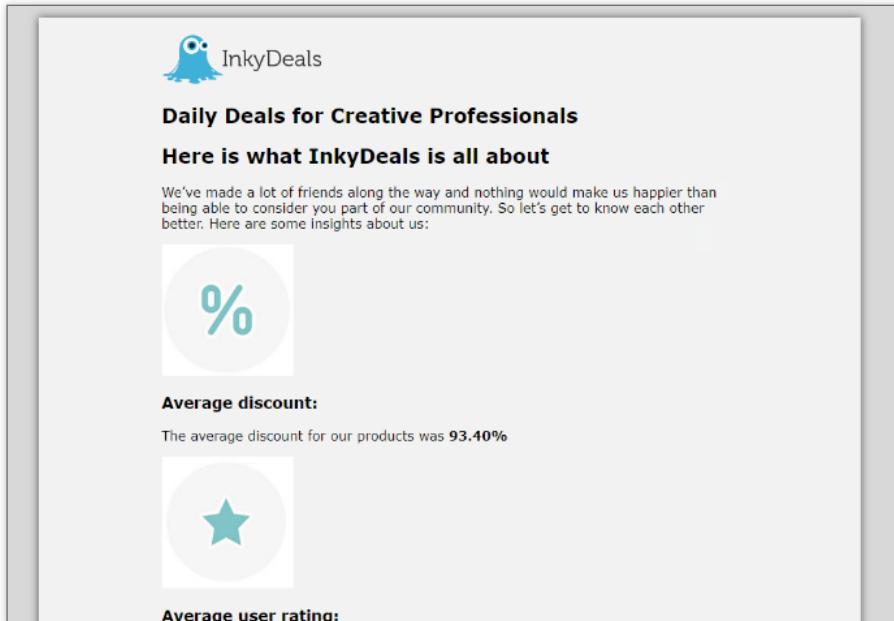


Figure 22

Let's add a border to `div id="main"` and change fonts for the titles:

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

Place the logo on the left and the slogan on the right. We have divs with the relevant ids for this in the markup.

```
#logo { float: left; }
#slogan {
    float: right;
```

```

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

}

```

The result is peculiar: the border set for div id="main" has moved above the logo and slogan, and the title of the first block has come along:

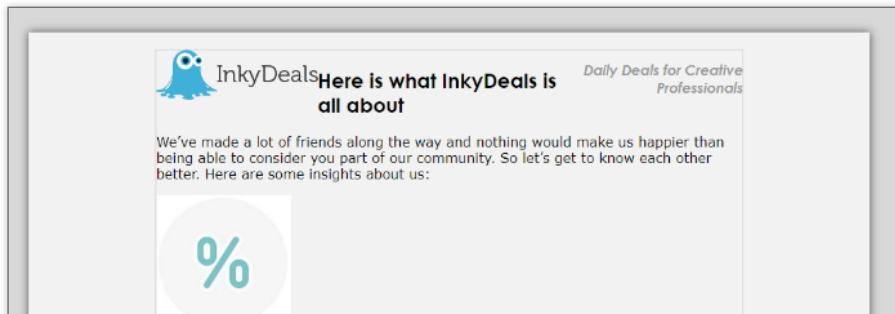


Figure 23

This all has happened due to the use of the `float` property: do you remember about its feature to turn the height of the parent element to `0` (in our case `header`)? This is why we use the rules of the `clearfix` class for the header:

```

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

.clearfix::after { clear: both; }

```

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

  <div id="slogan"> ... </div>
</header>
```

By adding the `class="clearfix"` attribute to the `header`, we will get the desired appearance of the article header:

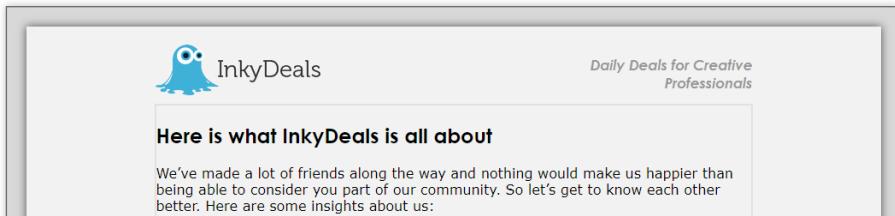


Figure 24

Let's now change the appearance of blocks by adding class styles for `div class="block"` and images.

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

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

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

The appearance of the block has improved, but they lack highlight.

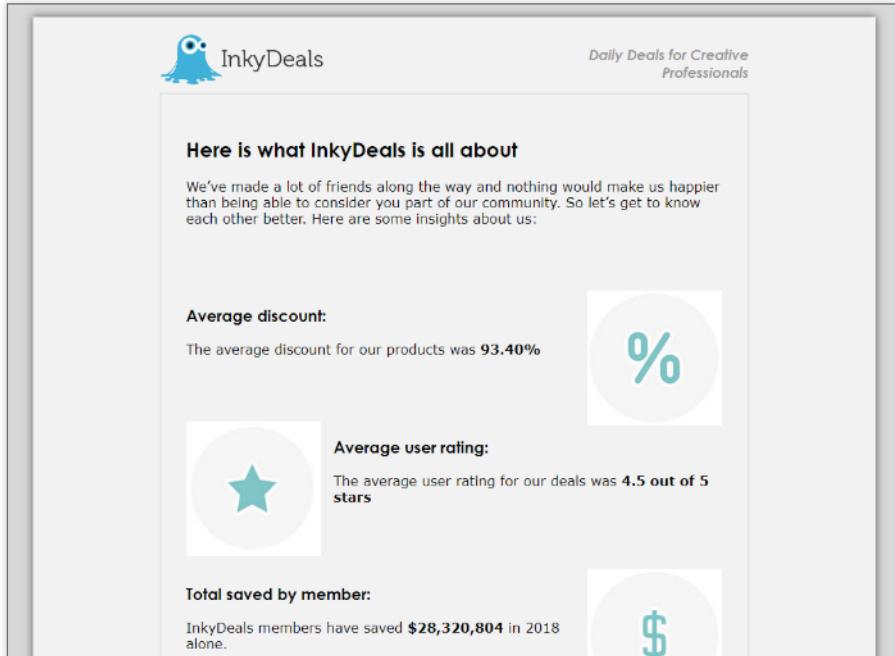


Figure 25

Let's add a light color to the background and frame at the top and bottom for the odd blocks:

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

However, there were some surprises here, too: the frame at the top and bottom is enlarged, and the images go outside the block.

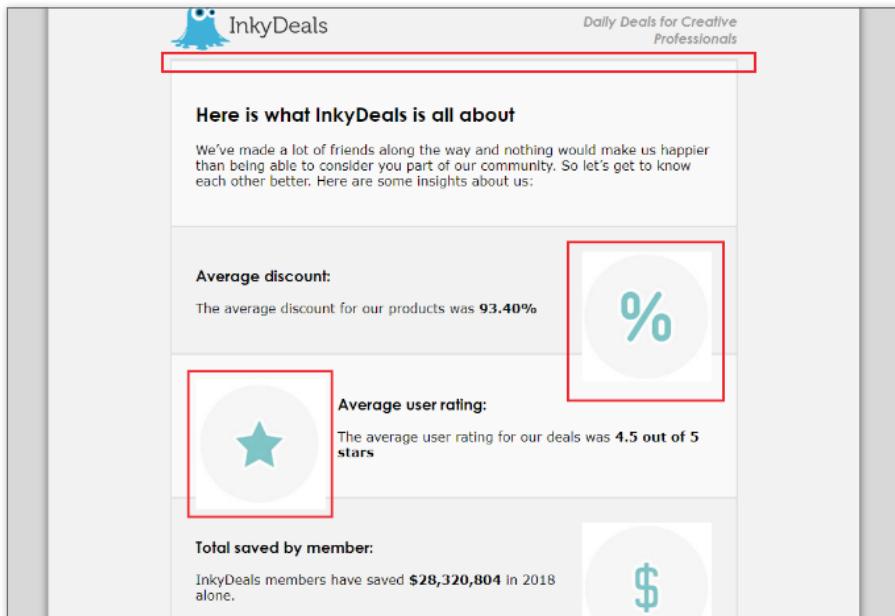


Figure 26

Change the `div id="main"` properties for the frames:

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

And add the rules for `.block` like we did for the `clearfix` class:

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

Now the images do not go outside the blocks:

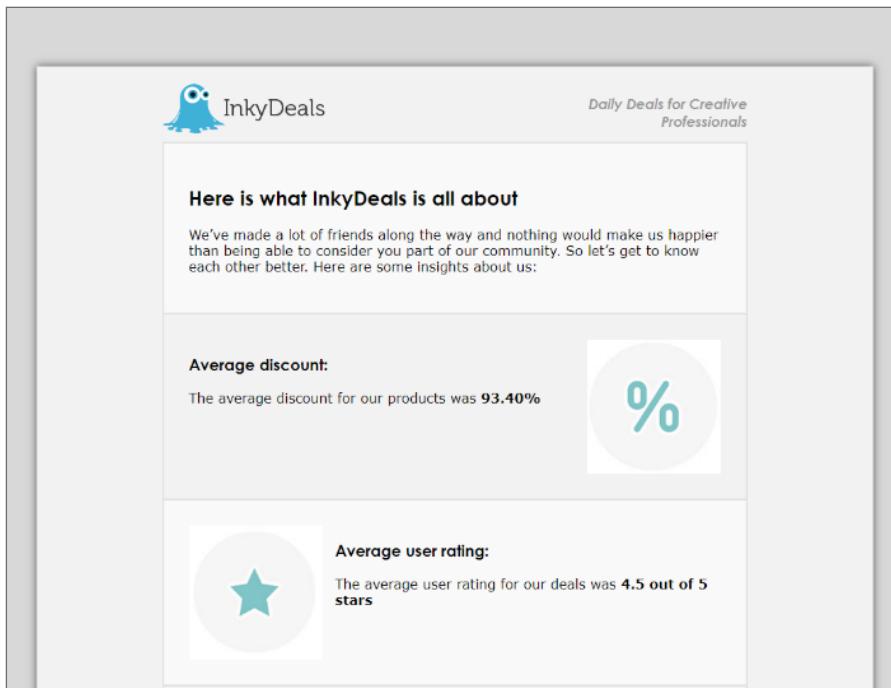


Figure 27

Another problem with images is that the frame around the JPG images is highlighted with an ugly square with the block in the background. You can remove the background around the image and resave them in the PNG format with transparency. Another way is to make the images round with border-radius: 50%:

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

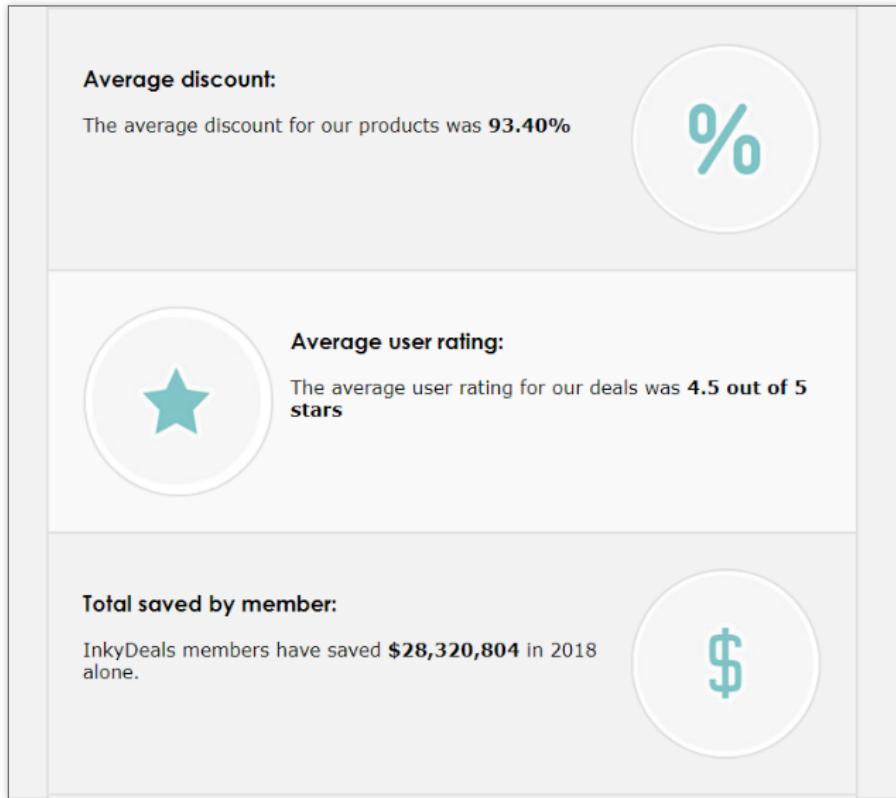


Figure 28

Now let's change the formatting for the first and last div class= "block", wherein we only have text, and increase the paddings in header:

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

The article looks as intended now.

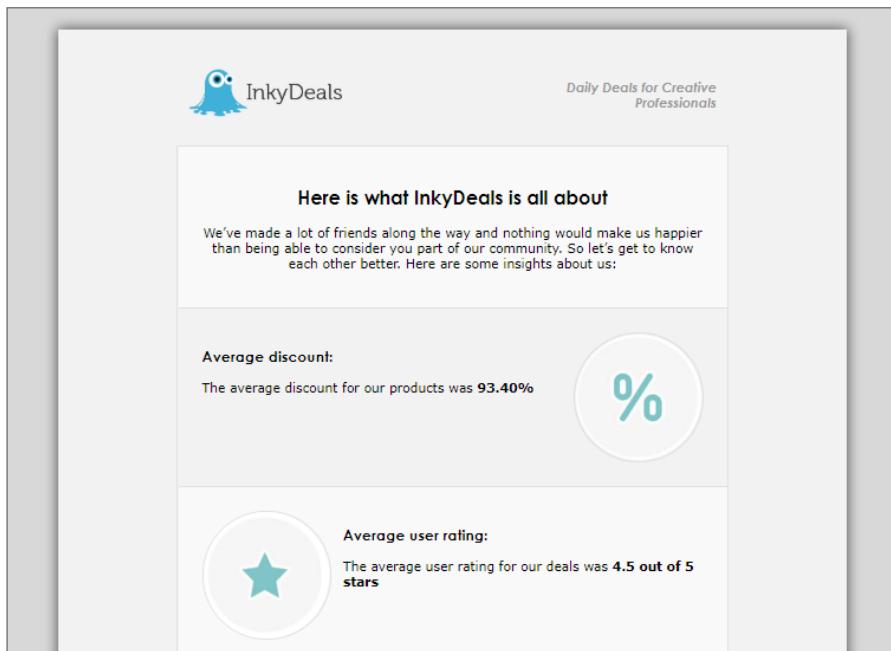


Figure 29

The figure and figcaption Tags

Quite often, images are accompanied by headings. If this is true for your page, you can use the `<figure>` and `<figcaption>` tags.

The `<figure>` tag is designed for inserting *images*, *code*, *videos*, and even *text*, e.g., a verse or advertisement. This element is a self-contained block of content. This means that if you move such a block anywhere in the document, it will not affect the meaning of the markup of the entire page as a whole.

We are interested in this tag precisely from the point of view of inserting images with a description. You can insert one

or more images into this tag by adding a description for them inside the `figcaption` tag. This can be a section of an article with a description, for example:

```
<figure>
    <img src = "images/airplane.jpg"
        alt = "Airline pilots report person on
                jetpack near LAX">
    <figcaption>
        There is a <strong>mystery</strong>
        unfolding in Los Angeles.
        It wasn't Buzz Lightyear or Robert Downey
        Jr. in "Iron Man," but pilots landing
        at Los Angeles International Airport
        on Sunday evening reported seeing
        a <strong>man wearing a jetpack</strong>
        flying near their planes.
    </figcaption>
</figure>
```

The `figure` tag has margins equal to `1em` at the top and bottom, and `40px` on the left and right.

Tags: <figure> & <figcaption>



Авиакомпания Pegasus Airlines (Турция) сегодня открыла прямой регулярный рейс Анкара-Одесса-Анкара. Себ этом сеюдня на пресс-конференции сказал директор международного аэропорта «Одессы» Виталий Портико, передает УНИАН.

Elements
Console
Sources

body
#80
#81
#82
#83
#84
#85

html
body
figure
img

Styles
Computed
Event listeners
DOM Breakpoints

Filter
figure
.figure-figcaption

Figure
width: 660px;
padding: 10px;
border: 3px double #cde;

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

Inherited from body
body
.figure-figcaption

Figure 30

32

You can set your own styles in order to design this block. You can also set them for **figcaption**.

You will find this and another example in the file named *figure-figcaption.html* of the *examples* folder.

You can search the web on your own to find a number of examples of images and captions with these tags and CSS animation and/or Javascript, for instance, on codepen.io.

Related links:

1. <https://www.w3.org/TR/html5/grouping-content.html#the-figure-element>;
2. <https://www.sitepoint.com/quick-tip-the-right-way-to-use-figure-and-figcaption-elements/>.

Properties of the background Group

In addition to inserting an image into the page using the **** tag, we can manage graphics with the help of CSS. In this case, we will talk about a whole group of properties, which begins with the **background** word.

You are already familiar with one of the properties in this group — **background-color**. Recall that it can be set as follows:

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

The next and perhaps the most important property of the background group is **background-image**. By default,

elements have no background image, that is, the `background-image` is set to `none`. You can also use `none` to delete a background image, for instance, using JavaScript when an element is clicked.

The syntax of the `background-image` is as follows:

```
background-image: none (by default) |  
    url(path_to_image.extension);
```

For instance,

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

By default, the browser repeats the specified image horizontally and vertically, in the shape of tiles. This behavior can be changed using the `background-repeat` property with the following values:

```
background-repeat: repeat (by default) |  
    no-repeat | repeat-x | repeat-y |  
    space | round
```

When repeating a background image, it is important that the joints or alignments of adjacent images are not visible, that is, you need to use *seamless textures*. Go to the folder named *examples/background* and find the *bg-repeat.html* file in it. The body of this file has the following background image:

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

Let's now replace the background with another picture: *examples/background/bg/abstract-bg.jpg*.

background-repeat: repeat

Lorem ipsum dolor sit amet, consectetur adipisciing elit. Perferendis sequi fuga repellat laborum blanditiis voluptatem, quae quod eos dignissimos corrupti. Nihil obcaecati perspicatis 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 quos commodi. Error id alias quisquam! Tempora officia minima temporibus.

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

Hic et iste expedita optio nihil totam velit asperiores magnam atque, eum accusantium nobis cupiditate sint obcaecati id officia officis aliquid at blanditiis ipsa porro perferendis. Repudiandae debitis pariatur libero.

Figure 31

Changing the background image immediately leads to 2 conclusions:

- Clearly visible joints of adjacent images look bad both horizontally and vertically;
- The bright background image makes the text unreadable.

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

background-repeat: repeat

Lorem ipsum dolor sit amet, consectetur adipisciing elit. Perferendis sequi fuga repellat laborum blanditiis voluptatem, quae quod eos dignissimos corrupti. Nihil obcaecati perspicatis 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 quos commodi. Error id alias quisquam! Tempora officia minima temporibus.

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

Hic et iste expedita optio nihil totam velit asperiores magnam atque, eum accusantium nobis cupiditate sint obcaecati id officia officis aliquid at blanditiis ipsa porro perferendis. Repudiandae debitis pariatur libero.

Figure 32

The last remark is more important for a designer, but you, as a layout designer, also need to pay attention to such things.

When using the `background-repeat` property set to `repeat-x` or `repeat-y`, the image will be duplicated only horizontally or vertically, respectively.

For example, we need to add design to the top of the page (the file `bg-repeat-xy.html` in the `examples/background` folder). In this case, the rules for `body` are as follows:

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

Notice that the text inside the `body` was pushed down by a large `margin-top` (`230px`) in the composite `margin` property. It should be noted that using `margin` with other elements will not produce the desired effect. To offset the content from the border of the element, use `padding`.



Figure 33

There is also the **blockquote** element in the text. Let's use a vertical repeating image for it:

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



background-repeat: repeat-y

Ipsum dolor sit amet, consectetur adipisciing elit. Molitia, quam, eos. Voluptatibus, delectus. Autem nesciunt aut dolorum dolor, recusandae iure repudianda dignissimos, laborum provident sapiente quam. Eaque nulla necessitatibus, praesentium?



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



Atque dolorum fugit reiciendis provident ducimus similiique consequuntur quasi neque dolorem voluptates, error recusandae ratione ullam corporis exercitationem perspiciat, pariatur nam! Veniam, sunt molestias, ullam beatae ducimus magni quis debitis.

Figure 34

- **Note:** the *paint.jpg* image, which was used to decorate the *body* background, has a light gray background and stands out against a white background of the *body*. Therefore, it makes sense to make the background of the entire page the same as in the image. To sample a color from an image, you can use Adobe Photoshop and define the color as *#f0f0f0*.

There is no difference in color tones after you save the styles.



background-repeat: repeat-x

Lorem ipsum dolor sit amet, consectetur adipisciing elit. Perferendis sequi fuga repellat laborum blanditiis voluptatem, quae quod eos dignissimos corrupti. Nihil obcaecati perspicatis 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 quois commodi. Error id alias quisquam! Tempora officia minima tem poribus.

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



background-repeat: repeat-y



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



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

Figure 35

The next 2 options for repeating the background image are **space** and **round**. These values appeared in CSS3, so they are still less popular than others familiar since CSS2.

When using the **space** value for **background-repeat**, the background image will be repeated several times to completely fill the area. However, unlike the default **repeat**, if the image does not completely fit in the element area, white space appears between the images, and sometimes it is too large (depending on the size of the background image and the element). The repetition occurs only horizontally (similar to **repeat-x**), so the excess text may not be visible, as in the example file *examples/background/bg-repeat-space-round.html*.

The **round** value also places an integer number of images in the area of the element, but at the same time, scales them without maintaining proportions.

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

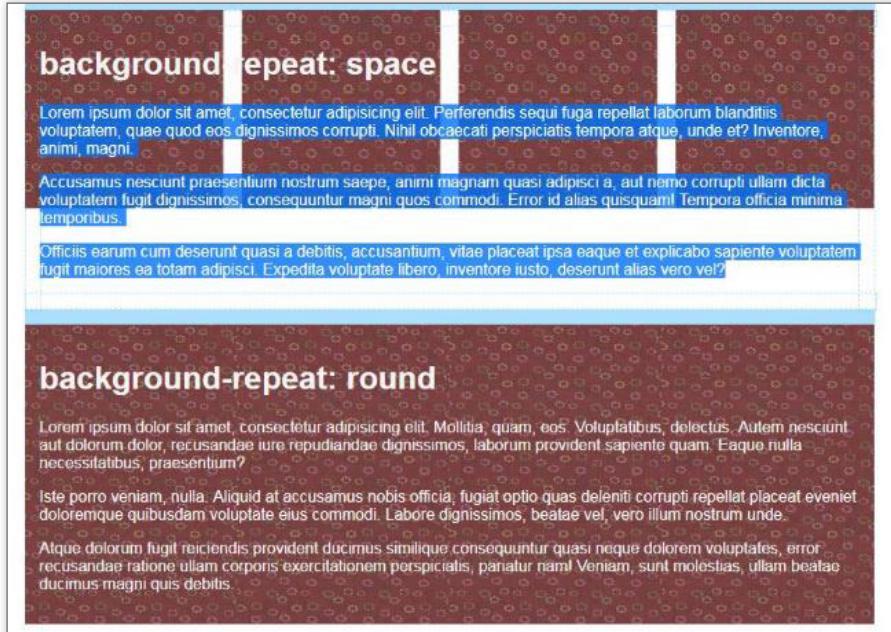


Figure 36

- ▶ **Note:** to avoid the repetition of the space type, it is necessary to set not only the background image to the element, but also the `background-color`. This will help the user read the text, even if the image has not loaded.

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



Figure 37

The block color can be determined using the same old Adobe Photoshop. The `no-repeat` value assumes that the background image will not repeat, that is, it is inserted only once into the page. Let's use this value to design `<div id="wrap">` with the text consisting of 2 paragraphs.

```
#wrap {
    width: 70%;
    margin: 20px auto;
    background-image: url(bg/photo-bg.jpg);
    background-repeat: no-repeat;
    font-family: sans-serif;
}
```

It does not look very nice.



Figure 38

Let's fix this situation by either using tips in the text editor (**Brackets**) or opening the image in the folder:



The screenshot shows a code editor window with the following CSS code:

```
<!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>
```

To the right of the code, there is a preview window showing a camera icon with the text "852 × 480 пикселей" (852 × 480 pixels) below it.

Figure 39

When hovering over the image file path, we can see the image itself and its dimensions in the value of the `background-repeat` property. Let's use the dimensions of our background for the dimension of the div: instead of `width: 70%`, we set the width of the image and add height, as well as paddings and `text-shadow` to improve the text readability. Let's immediately recall the increase in the block size due to paddings, so we will also add `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
}

```

The resulting block is:

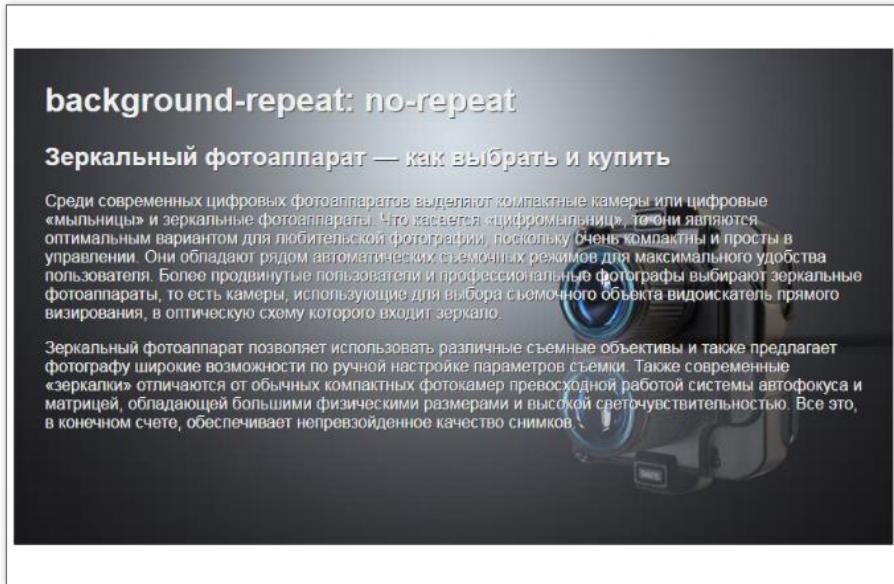


Figure 40

You will find this example in the *bg-no-repeat.html* file of the *examples/background* folder.

The **background-position** Property

The **background-position** property is responsible for placing the background image relative to a block or page. It can be specified with keywords or values in **px**, **pt**, **em**, **%**, etc. for x and y coordinates (horizontal and vertical placement):

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

The keywords help place the background image horizontally (the coordinate x — `left`, `center`, `right`) and vertically (the coordinate y — `top`, `center`, `bottom`). You can specify only one coordinate as a keyword, then the second one will be used by default as `center`. The order of the keywords is not important. If you use values in any unit of measurement, the `X` coordinate is indicated first, then space, and the `Y` coordinate. When specifying the `background-position` value in percents, it should be borne in mind that a number of values coincide with the keywords:

- `top left` = `left top` = `0% 0%`;
- `top` = `top center` = `center top` = `50% 0%`;
- `right top` = `top right` = `100% 0%`;
- `left` = `left center` = `center left` = `0% 50%`;
- `center` = `center center` = `50% 50%`;
- `right` = `right center` = `center right` = `100% 50%`;
- `bottom left` = `left bottom` = `0% 100%`;
- `bottom` = `bottom center` = `center bottom` = `50% 100%`;
- `bottom right` = `right bottom` = `100% 100%`.

You can see the result of placing the background image with `background-repeat: no-repeat` in Figure 41 and in the file named `bg-position.html` in the folder `examples/background`.

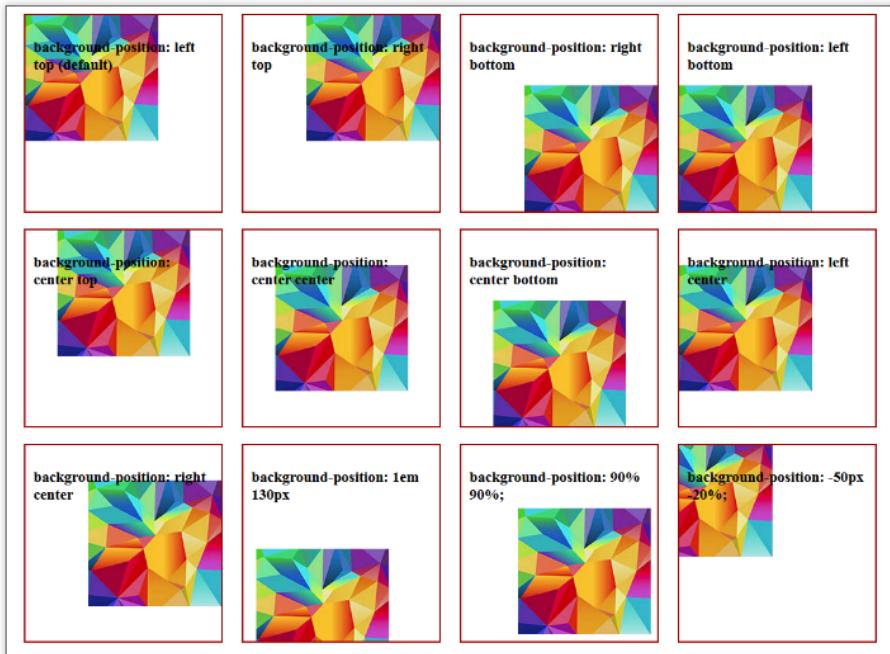


Figure 41

The background-size Property

The `background-size` property first appeared in the CSS3 standard and was used to control the size of the background image. This is a very convenient property that allows you to stretch or, vice versa, reduce the image size, adjusting it to fit the block size.

```
background-size: [ size in px, pt, em, rem etc. | % |  
               auto ]{1,2} | cover | contain
```

- **Note:** since this property was introduced in the CSS3 specification, a number of legacy browsers have been using this property with vendor prefixes. Go to caniuse.com to read more about them.

Let's consider the `background-size` property through examples. Figure 42 (example file is *bg-size.html*) shows that this property does not influence the image size by default (the value is `auto`) — it matches its real size. The value `100% 100%` stretches the image to fit the container without maintaining the image proportions. The values `100% auto` or `auto 100%` stretch the image to the horizontal or vertical size of the block while preserving the proportions. The values `contain` and `cover` in the second row of the example do the same, but `cover` changes the background size while maintaining the proportions so that its width or height is equal to the width or height of the block, and `contain` preserves the proportions so that the image fits entirely inside the block. The value `background-size: 50px 50px` has made the size of the background image equal to `50x50px`, and the value `50% 50%` has scaled the image so that it occupied exactly half of the width and height of the element without maintaining the proportions.

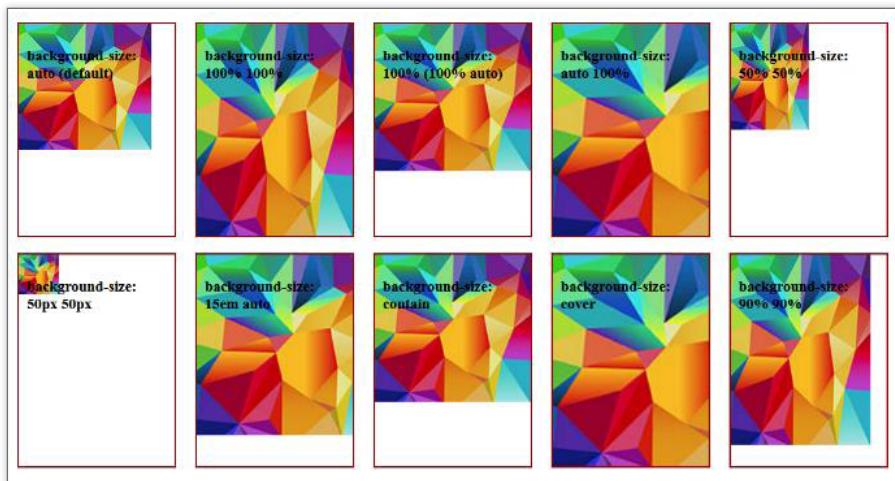


Figure 42

After considering all the proposed options, we can conclude that the `auto` value maintains proportions of the background image for any of the two parameters. If only one value is set, then it will control the width of the background image, the second value, in this case, is `auto`, which allows you to maintain the proportions of the image. Specifying two values separated by a space sets the `width` and `height` of the background image, respectively.

The `background-attachment` Property

The `background-attachment` property allows you to specify whether the background image will scroll along with the content of the element or the entire page. Using this property, you can fix the position of the image, which allows you to scroll the content above it. The `background-attachment` values are:

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

The `scroll` value (by default) scrolls the background image along with the content. The `fixed` value fixes the image, and the content is scrolled above it.

You can see the difference in these two values in the `bg-attachment.html` file of the `examples/background` folder. It specifies the `background-attachment: scroll` property for the first background, which you can safely comment out without changing the visual display of the page, since this value of the `background-attachment` property is default (Fig. 43).

The `local` value fixes the background relative to the element content, that is, when scrolling the page or the element itself, the background is scrolled along with the website element. You can see how this value works in the file named `bg-attachment-local.html` in the `examples` folder (Fig. 44).



Figure 43

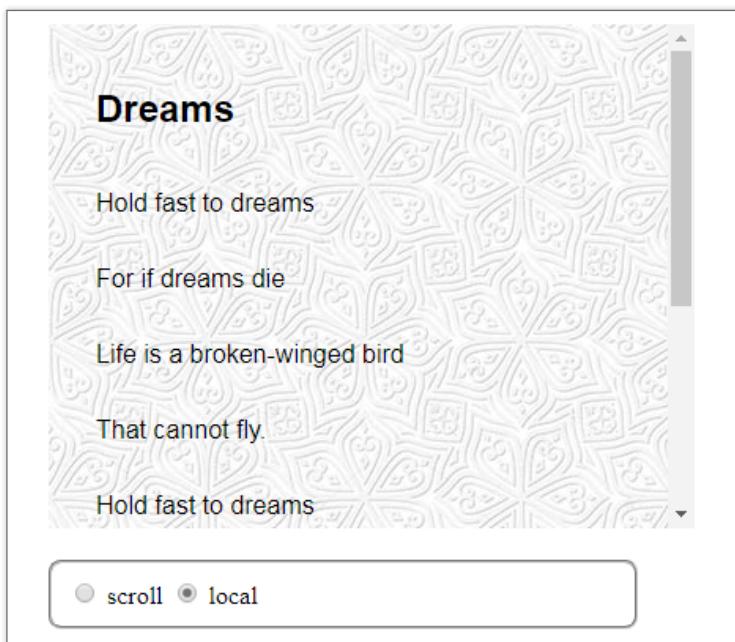


Figure 44

The background-origin Property

The `background-origin` property determines in which area the background image will be positioned. Feature of use: you cannot apply `background-origin` if the `background-attachment` value is set as `fixed`.

The `background-origin` can take the following values:

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

Visually, this property determines whether the background image will take up space, including paddings (`padding-box`), paddings + borders (`border-box`), or only the content area without paddings and borders (`content-box`).



Figure 45

You will find this example in the `bg-origin.html` file of the `examples` folder.

The background-clip Property

The `background-clip` property determines how the background color or background image will be displayed below the element's borders. You can see the effect of this property with either transparent or dashed or dotted borders.

Values that `background-clip` can take:

```
background-origin: padding-box |  
                  border-box (by default) |  
                  content-box | text
```

The first three values of the `background-clip` property look similar to the `background-origin` property. The `text` value allows you to cut the text from the background. To do this, add the `color: transparent` property to the rest of the styles:

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

You can see this example in the file named *bg-clip.html* in the *examples* folder.

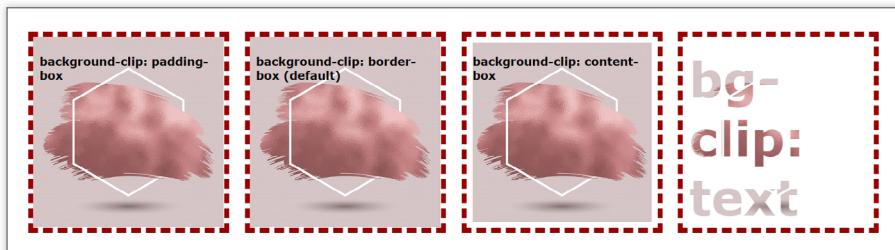


Figure 46

- **Note:** the `background-origin` and `background-clip` property first appeared in the CSS3 standard and are not supported by early versions of Android and Opera. Find more information about it on caniuse.com.

The Generalized background Property

You may need several properties of the `background` group. In order to shorten the notation of CSS properties, it makes sense to use a universal or generalized `background` property. Its syntax is as follows:

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

In this property, it is important to know the default value of its properties:

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

The following code is used for a background element in the file named `examples/background/bg.html`:

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

It lists not all properties of the `background` group, but only those that are necessary for this `div`. All other properties take default values. The appearance of this example changes

depending on the size of the open browser window, because the `background-size: cover` property is used as part of the generalized `background` property.

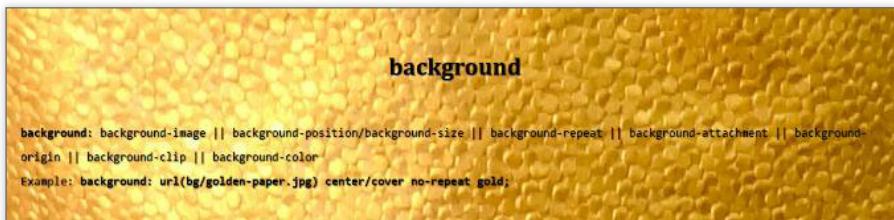


Figure 47

You can see the value of all other properties in the Property inspector (F12 key) on the `Computed` tab.

The screenshot shows a browser's developer tools with the CSS panel open. The 'Computed' tab is selected. A search bar labeled 'Filter' is present. Below it, a list of properties under the 'background' section is shown, with 'background' itself expanded to show its sub-properties:

Property	Value
<code>background-attachment</code>	<code>scroll</code>
<code>background-clip</code>	<code>border-box</code>
<code>background-color</code>	<code>rgb(255, 215, 0)</code>
<code>background-image</code>	<code>url(http://127.0.0.1:1640/bg/golden-paper.jpg)</code>
<code>background-origin</code>	<code>padding-box</code>
<code>background-position-x</code>	<code>50%</code>
<code>background-position-y</code>	<code>50%</code>
<code>background-repeat-x</code>	
<code>background-repeat-y</code>	
<code>background-size</code>	<code>cover</code>

Figure 48

- **Note:** you can first use the generalized `background` property enumerating a number of properties, and then write an additional property on a separate line, but you cannot do the opposite — first write a property, and then write `background`. In this case, all unspecified properties will be reset to their default values.

The example (the file named *examples/background/bg2.html*) first sets the generalized **background** property for the **bg** class, and then adds various **background-image** properties for the **.bg1** and **.bg2** classes:

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

Change the order of the properties for the last block with the *my-bg* class:

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

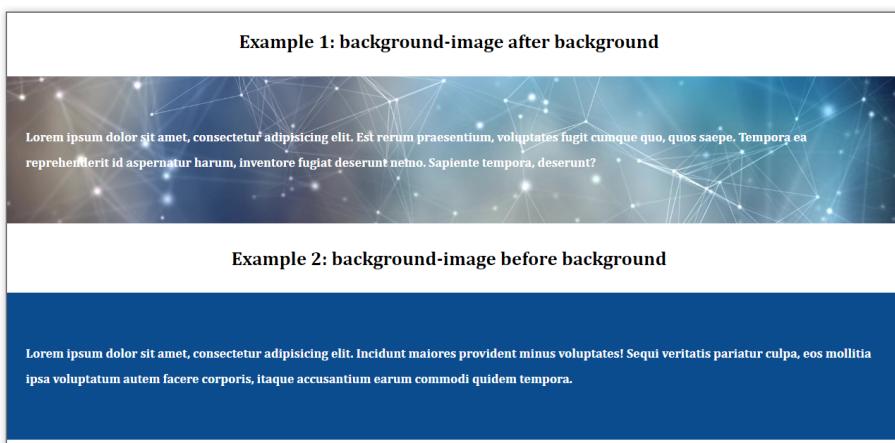


Figure 49

As a result of such actions, we will see a text against the blue background of an image, because the property is set to its default value: `background-image: none` (Fig. 49).

It is this value that the Property inspector shows us on the **Computed** tab.

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

Figure 50

Related links:

1. https://www.w3schools.com/cssref/css3_pr_background.asp;
2. <https://developer.mozilla.org/en/docs/Web/CSS/background-clip>;
3. <https://codepen.io/Jintos/pen/crlxk>.

Multiple Backgrounds

CSS3 makes it possible to add multiple backgrounds to the same HTML element. To do this, you need to put a comma and write the desired value in each property of the `background` group. Let's consider an example in which we need to place 3 images of children below the main text + the background for the entire text:

```
#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;
}

```

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 наблюдать 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 «гудить» и лепетать. И поскольку нет ничего привлекательнее улыбающегося, дпечечущего малыша, вы не сможете не ответить на его приветствие.



Figure 51

Note the `background-image` property: it first contains images with children, and only at the very end — the repeated background in the form of a squared notebook. If we did the opposite, the squared background would overlap the images of the children, and there would be no point in adding them. Experiment with the styles yourself.

The `many-bg.html` file of the example from the `examples` folder looks as follows (Fig. 51).

You can use the shorthand notation to assign multiple backgrounds using the `background` property. And if you need default values, you simply do not specify them.

The second text block in the file named `many-bg.html` from the `examples` folder is styled similar to the first one, but using the generalized background `property`:

```
#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.



Figure 52

Notice that `padding-bottom: 270px` was set for both divs in this file so that the `256x256px` images could be easily placed under the text.



Figure 53

PSD Layout

Layout assumes some kind of design. Of course, you can design a layout using a PNG or JPG image, but then you will face the problems of copying text and saving images. Moreover, there will be no such handy tool as guides that allows you to measure the distances between blocks.

At the moment, designers prefer to work with software such as Adobe XD, Adobe Photoshop, Figma, Sketch, Affinity Designer (Photo), etc. The choice of software usually depends on the preferences of a particular designer or software purchased and installed in the design studio. Since the use of Adobe Photoshop is still very popular among designers, and you can download on the Internet a large number of free PSD layouts for developing your layout experience, today we will consider how you can save (export) images, calculate indents, and copy text from a PSD file.

The layout was [downloaded from one of the sites](#) offering a range of free PSD layouts. In this part of the lesson, we will analyze only half of the layout (in the screenshot below). You will have to do the rest at home.

The screenshot displays a professional web design template for a creative agency named 'Startup'. The header features a dark teal bar at the top with the word 'Startup' in white. Below it is a navigation menu with links: Home, Services, About, Works, Blog, Clients, and Contact. The main visual is a photograph of a red brick building with a central entrance featuring a light-colored wooden door and a small window. A 'WELCOME TO STARTUP' banner is overlaid on the image, along with the subtitle 'Your Favourite Creative Agency Template' and a 'GET STARTED' button. The page is divided into sections: 'SERVICES' (with three circular icons labeled 'CLEAN TYPOGRAPHY', 'ROCK SOLID CODE', and 'EXPERT SUPPORT'), 'ABOUT US' (with four staff portraits and their names listed below), and a footer.

Startup

Home Services About Works Blog Clients Contact

WELCOME TO STARTUP
Your Favourite Creative Agency Template

GET STARTED

SERVICES

We offer ipsum dolor sit amet, consectetur adipiscing elit.

CLEAN TYPOGRAPHY

Ipsum dolor sit amet, consectetur adipiscing elit. Laudantium quis ornarellium instantiimplab, ut id quiescam. Nigat corporis, tunc autem.

ROCK SOLID CODE

Vulputa quis nato dispicimus velutus aliquam nisi magis auct, dolores fonsius conundratur cens et laborenum instanti quam, placent quece neq.

EXPERT SUPPORT

Doloremque metella quod dolores, eumur venitatu, curia, Distefio aliqas, liga blandit dolores tili modine curar, ratios, perpeccita aise partur.

ABOUT US

Ipsum ipsum dolor sit amet, consectetur adipiscing elit.

Ipsum ipsum dolor sit amet, consectetur adipiscing elit. sediam rorari eratpor iustitiae et labore et dolores dligimus, et id dñam volgatis. Ali vero eos eadomus et piso dolores et ea solvus. Sic. Ha leol givengen, no se talizata exectus or Leon pries dekor et ruse. Lores ipsum dolor et dolores dligimus. Lores ipsum dolor et amet. Lores ipsum dolores.

MD. KHALIL UDDIN
Lead of Team

RUBEL MIAH
Lead WordPress Developer

SHAMIM MIA
Sr Web Developer

JOHN DOE
Front end Developer

Figure 54

First, let's conduct a small analysis of the site blocks:

1. The layout width is 1600px, but in the middle, a container with a width of 1140px is separated out with the help of guides; it constrains all the content in width. Let's create the `.container` class for it.

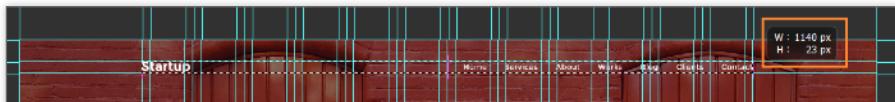


Figure 55

2. The font color in the site header is white, and in all other blocks #555.
3. The main font on the page is serif, and sans-serif for headings. The main font size (font-size) is 14px, the line-height is 24px.
4. Most of the headings have uppercase characters, the section headings have a line below the text, which, when the layer with it is selected with CTRL, has a size of 50x2px.
5. Some of the columns have a central text alignment, for which it makes sense to add a separate class: `.text-center`.
6. Information in the layout is split into 2,3, 4 columns of the same size, and only in the **Recent Blog Posts** block, the left side has 5 out of 12 columns (41.67%), and the right side has 7 out of 12, that is 58.33%.

Export Images

At the top of the layout, we see 3 layers that control the background image for the header of the site. First you need to glue these 3 layers by highlighting them and clicking **CTRL+E**. And then export the image to JPG.



Figure 56

The dimensions of the image will be the dimensions of our header. If the image may be slightly smaller in width ([1600px](#)) than modern screens, then we will use [770px](#) for the height.

A similar operation must be performed to export images from the Services and About Us blocks. Only in this case, it is hard to do without gluing layers by exporting a group (folder) of layers to a PNG or PNG-8 image.

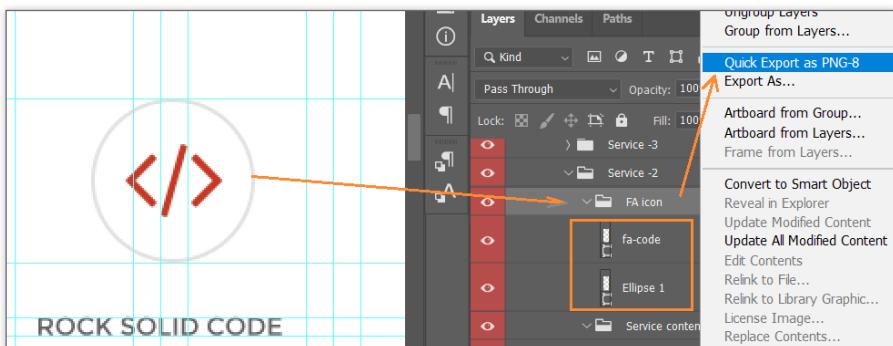


Figure 57

For the first photo of the About Us block, it is necessary to turn off the visibility of the layers with a gray rectangle and a block with a left arrow.

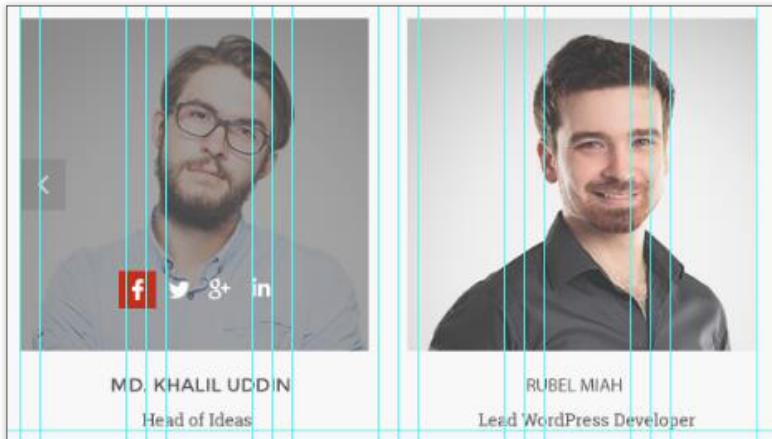


Figure 58

After that, you can export the group of layers to JPG, since transparency is not needed here.

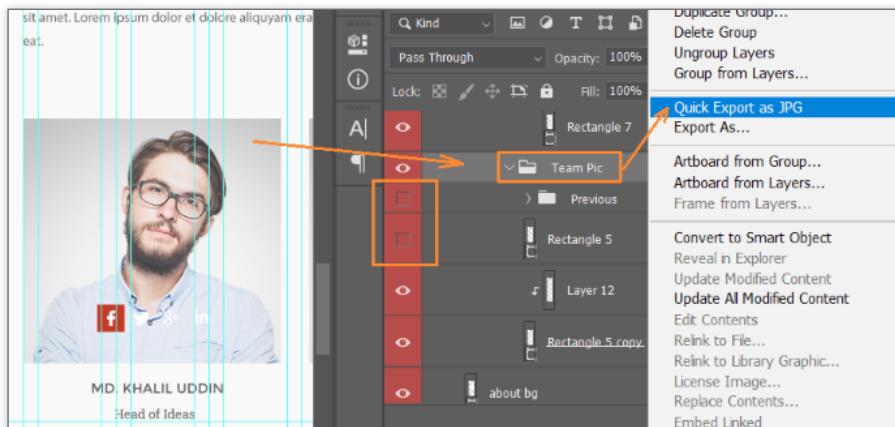


Figure 59

Copy Text

To create an HTML markup, we also need the text of each block, as well as its characteristics. Since we do not have

the fonts that are used in the layout, and we will not add them, we have 2 options for “getting” the text:

1. Copy it from layer name;

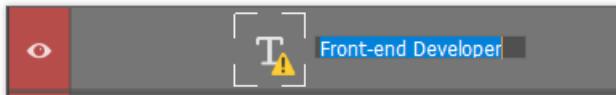


Figure 60

2. Double click on the T icon with a triangle, copy the text from the block, and view its characteristics in the Character panel: font size, line spacing, weight, and case.



Figure 61

Measure Indents

Indents can be measured with the **rectangular selection tool** while paying attention to the height (**H**) of the selection or its width (**W**). Here you will appreciate the guides, since the selection is automatically snapped to them (Fig. 62).

You can also select 2 or more objects by holding down **Shift** in the layout, and then hold down the **Ctrl** key and look at the numbers of indentation. The disadvantage of this method

is that you will not always get the numbers you need the first time (Fig. 63).

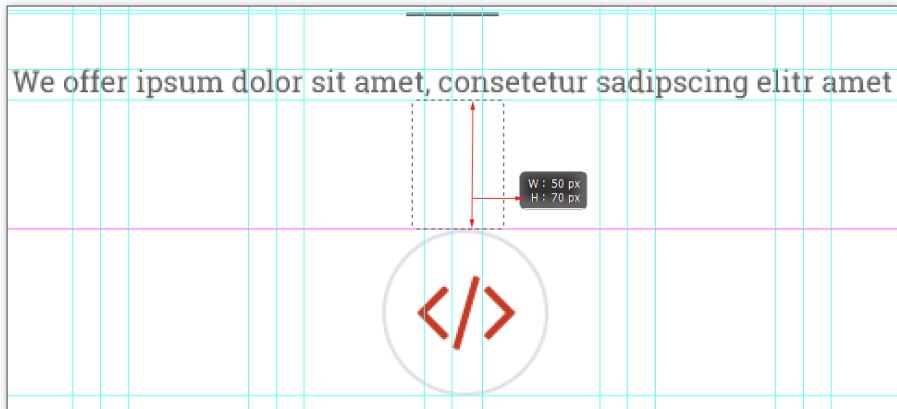


Figure 62

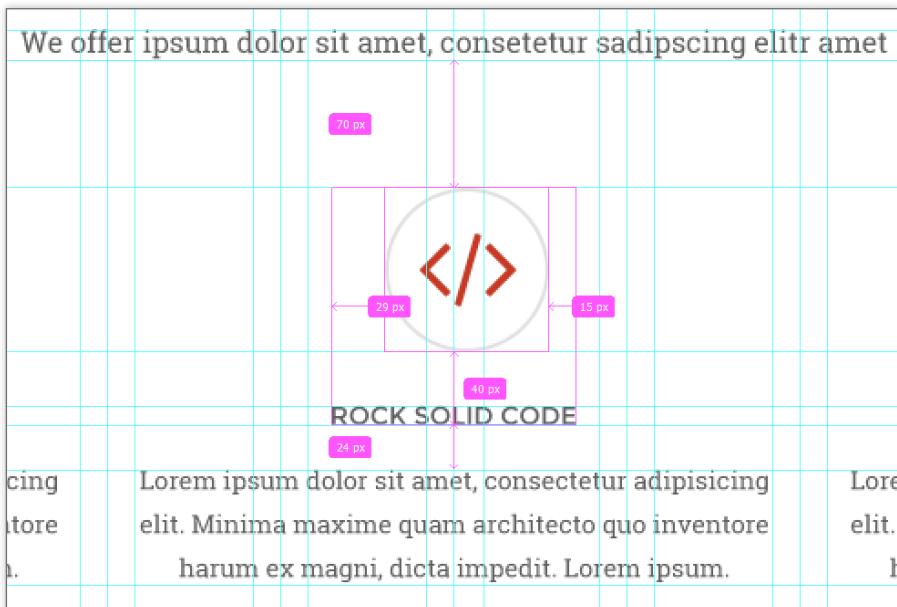


Figure 63

The indents before and after the text will have to be adjusted due to the value of the line height, which in CSS will also form some indent above and below the text block.

Website Header Markup

```
<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>
```

General styles for page and header:

```
* { 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; }
```

The Services Section

The Services section starts with a heading and a line and has paddings at the bottom and top, as well as 3 columns with an image, a heading, and center-aligned text. Let's create a markup.

```
<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
```

```

        adipisciing 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>
```

Styles:

```

.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; }
```

The `.row` class cancels wrapping that collapses the height of the parent element for the columns with the `.column-` class and `float: left` property.

The indents for all elements are taken from the layout and adjusted to fit the line heights of the elements.

About Us Section

Text blocks in this section are in 2 columns, and information about the team in 4, so the markup uses different classes: `.column-2` and `.column-4` inside the `.row` class:

```
<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>
```

We have some styles for headers and section, the rest assume indents between elements (the `.mt-team` class), as well as the usage of a `:hover` pseudo-class for images in order to create a shading effect using the `filter` property.

```
.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; }
```

Checking the Layout for Compliance with the Layout

A good PSD layout assumes that the designer makes the same indents between all similar elements (sections, headings, columns), and thus helps the layout designer get by with a minimum of classes and styles. In the proposed layout, this condition is met. Please note that the guide system in a PSD file is very clearly located, has the same vertical indents + many horizontal guides that make it easier to measure distances and sizes. However, this will not always be the case in your practice. Designers rarely use Adobe Photoshop alone for creating layouts. Quite often, it happens so that the indents

of many elements are different, the settings of the font blocks are also different. Only the color scheme is quite well.

Therefore, it would be nice to have a tool that allows you to understand how you “got” into the layout, that is, visually determine whether the design elements are in their place in the layout. You can use an extension for Chrome and Firefox named [Perfect Pixel](#) or a similar tool.

Install the plugin from the browser extensions catalog.

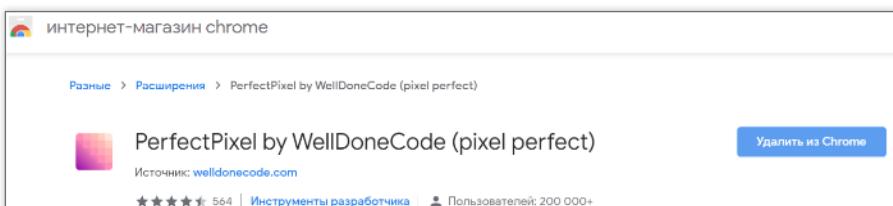


Figure 64

Next, you need to upload an image of your entire layout or its part, specify the transparency and location of the layout image by X and Y (0, 0) as well as scaling (0.8 in our case).

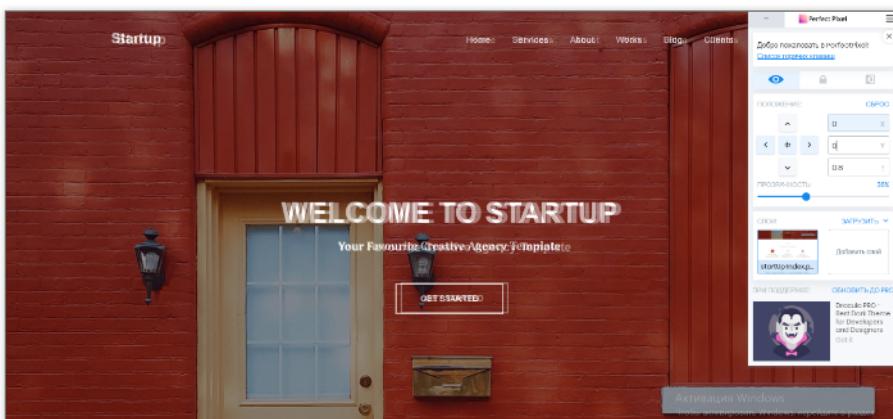


Figure 65

Then use the code inspector to check how well your elements match with the design, while adding or overriding styles.

The advantages of this plugin are that you control the transparency and position of the screenshot, and you can turn it off by pressing the eye button. The disadvantages are that at different resolutions of your screen and the horizontal resolution/size used in the layout, you will have to rescale the layout image to fit the browser on your screen.

The second option is you can create a separate element in your markup, for instance:

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

Add the following styles in the *style.css* file:

```
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;
}
```

As a result of these actions, a block with a background image appears above the elements of your layout, which is placed on top and centered relative to the browser. This is good because you do not need to adjust the dimensions, but you can shift the image using the **background-position** property (Fig. 66).

The disadvantage of this approach is that you do not have a slider with changing transparency at hand, you cannot turn off this element by clicking on the eye icon — for this you need to edit the HTML or CSS code.



Figure 66

Related links:

1. [Perfect Pixel for Chrome](#);
2. [Perfect Pixel for Firefox](#).

Homework

Finish the layout using the markup and styles from the *startup-page* folder and the PSD file as a basis. You do not need to create a layout the Contact and Feedback sections, since we have not discussed form elements yet, and feedback suggests the use of a JavaScript or jQuery plugin.

You should export the images yourself.

Notice the first image from the **Latest Works** block. You should disable the layer group named **Work** to export the image. The effect of the red translucent background when a photo is hovered is optional, because it required additional knowledge of element positioning or transformation.

Export all other images as a group of layers.

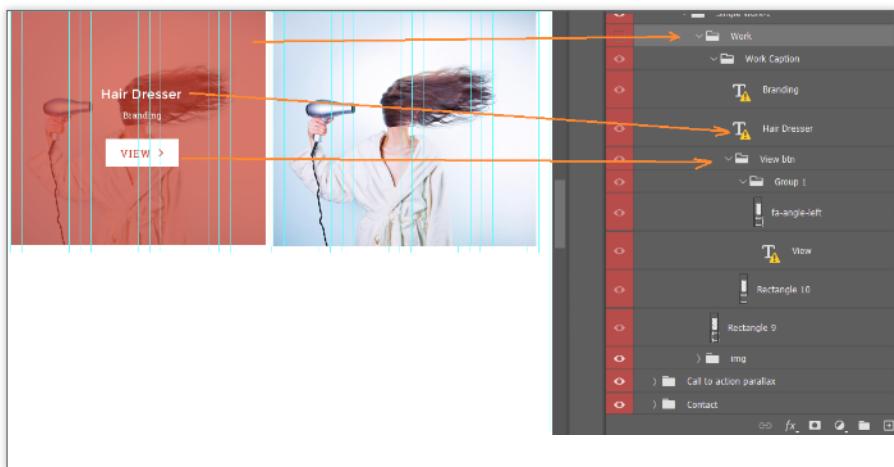


Figure 67

You will find a screenshot of the task below and in the folder named *startup-page/HW+Layout+PSD*.

Lesson 4. Graphics in Web Design

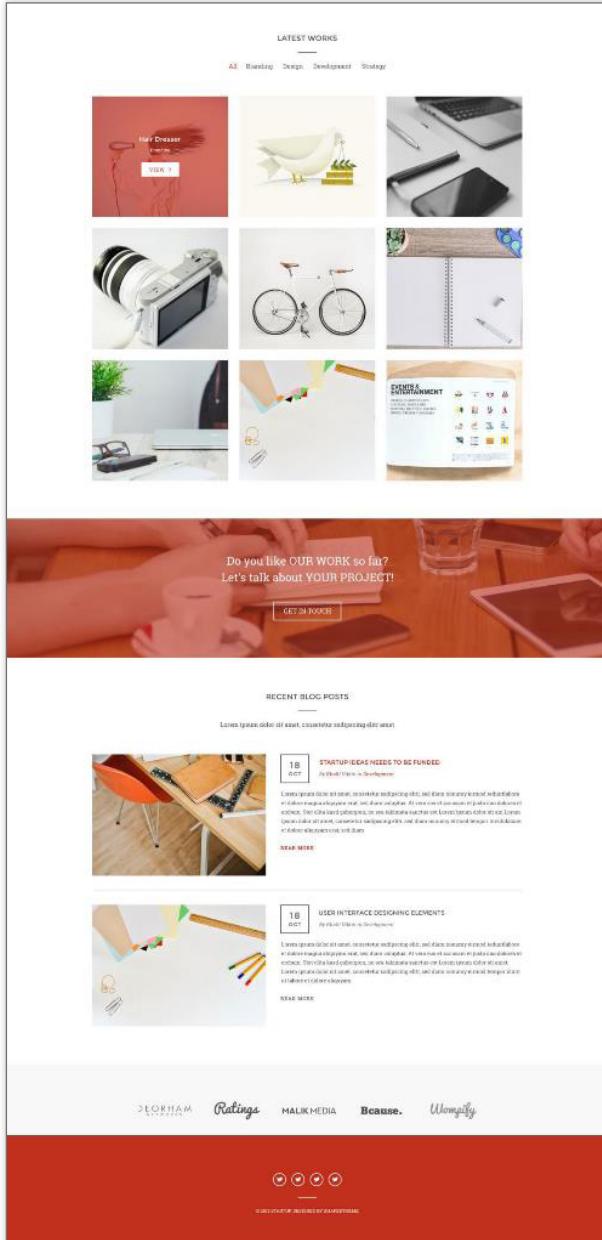


Figure 68

Homework



Lesson 4. Graphics in Web Design

© Slutskaya Elena.
© STEP IT Academy, www.itstep.org.

All rights to protected pictures, audio, and video belong to their authors or legal owners.

Fragments of works are used exclusively in illustration purposes to the extent justified by the purpose as part of an educational process and for educational purposes in accordance with Article 1273 Sec. 4 of the Civil Code of the Russian Federation and Articles 21 and 23 of the Law of Ukraine "On Copyright and Related Rights". The extent and method of cited works are in conformity with the standards, do not conflict with a normal exploitation of the work, and do not prejudice the legitimate interests of the authors and rightholders. Cited fragments of works can be replaced with alternative, non-protected analogs, and as such correspond the criteria of fair use.

All rights reserved. Any reproduction, in whole or in part, is prohibited. Agreement of the use of works and their fragments is carried out with the authors and other right owners. Materials from this document can be used only with resource link. Liability for unauthorized copying and commercial use of materials is defined according to the current legislation of Ukraine.