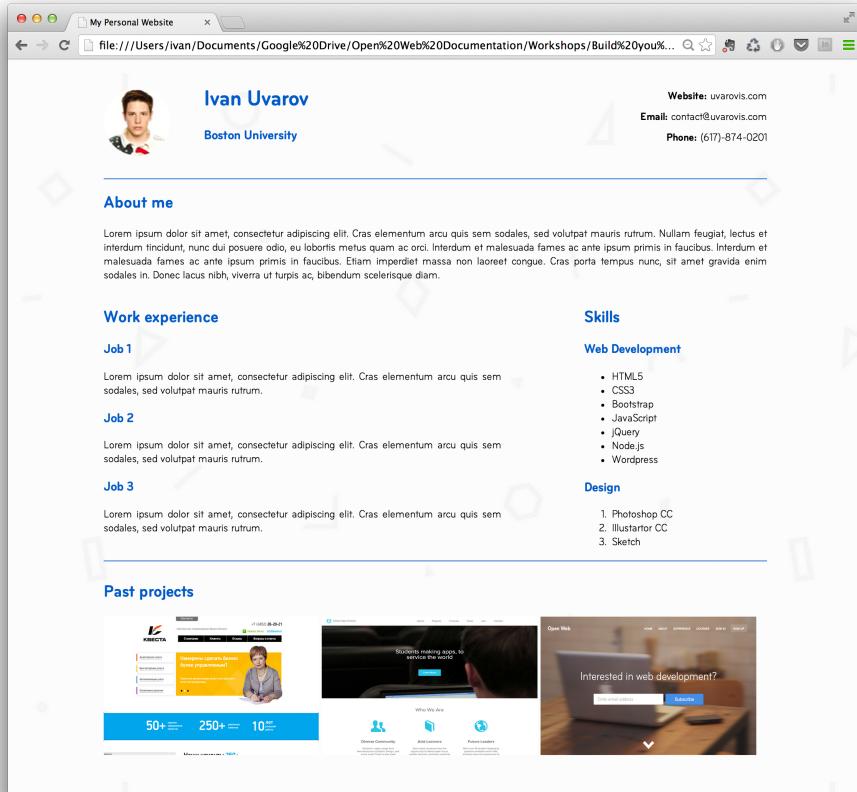


Build your first website



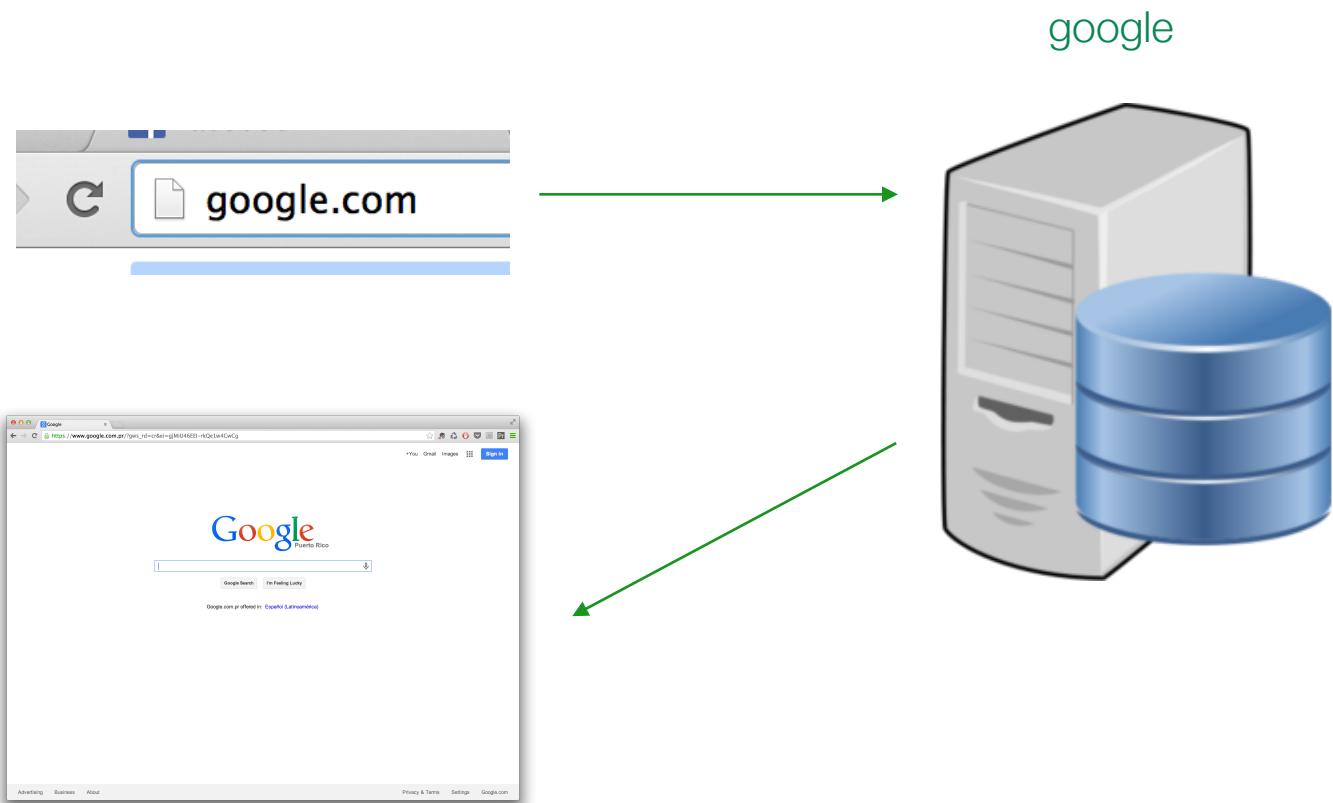
by Ivan Uvarov

Basics	3
Browser - server communication	3
Webpage structure	4
Introduction to HTML	5
Creating a simple HTML file	5
Your first HTML code	7
HTML document structure	8
Introduction to CSS	10
Creating your first website	12
Introduction to project	12
HTML tags	13
Class and ID	14
Making markup of the page with HTML	15
Styling your page with CSS	21
Useful resources	29

Basics

Browser - server communication

What happens when you enter a web address into your browser, say for example 'google.com'? All websites are stored in servers, special computers that are connected to the internet all of the time. The address of the website (google.com) explains to your computer how to find a specific server to download the page that you want to open. In this way, it's the same as telling someone your home address, like 700 Commonwealth Ave, Boston, MA, 02215. Your computer then asks this server to send a webpage to you. It sends you ready-to-open files of the requested webpage that your browser can open. Usually, the server sends three types of files: HTML, CSS, and JavaScript.



Webpage structure

As we mentioned later, the server sends you 3 types of files: HTML, CSS and JavaScript. Lets look into this files and see what they do and how they create a complete webpage.

HTML stands for **H**yper **T**ext **M**arkup **L**anguage. This language is used to create the structure of a page. It allows you to create a markup of the page, adding all the necessary information to the page including text, titles, paragraphs, images and so on. This language is based on tags and the content inside them: <tagname>content</tagname>. For example, to create a paragraph of text, you should do this: <p>This is a paragraph.</p>. This language creates a ready-to-open page but doesn't have any styles applied to it.

HTML code

```
1  <!DOCTYPE html>
2  <html>
3      <body>
4
5          <h1>My First Heading</h1>
6          <p>My first paragraph.</p>
7
8      </body>
9  </html>
```

Webpage

My First Heading

My first paragraph.

CSS stands for **C**ascading **S**tyle **S**heets, and it is used to add styles to your previously created markup of the page. For example, you use CSS to change the colors and positioning of all the elements on a website. Overall, CSS is what makes your website look pretty.

JavaScript is a more advanced language, and it used to add interactivity to your page. For example, if you want to change the background of the page when you click a button, you can use JavaScript. You won't need JavaScript for creating a simple webpage, but it's good to learn if you want to make more complicated websites.

Introduction to HTML

As it was mentioned before, HTML is based on tags and the content inside of them. The tag structure of the page helps to separate a page into different parts, so you can reference them when you style your page with CSS.

`<tagname>content</tagname>` - this is the typical structure of HTML tags. There is an opening tag, closing tag, and content between them. Almost all HTML have to be closed, so don't forget to do this!

Creating a simple HTML file

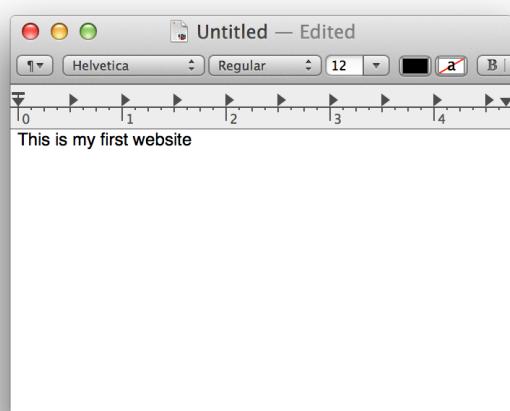
Let's make a very simple HTML file. You can write HTML code in any text editor, but you should save it with the extension `.HTML`: `file_name.HTML`. Then, you can open this file with your browser and you'll see your page!

To create HTML file:

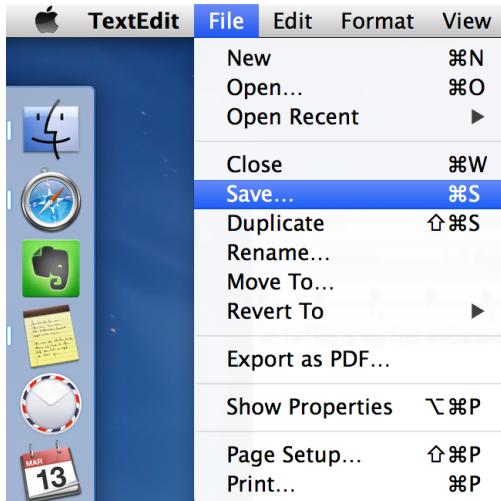
1. Open TextEdit (Notepad on Windows)



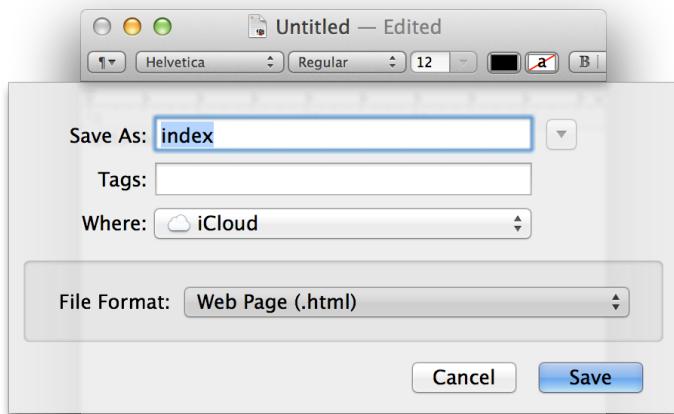
2. Type something if you want



3. Choose File - Save on top menu



4. Choose file format “Web Page (.html)” (If you’re on Windows, just give name to the file in format **file_name.html**)



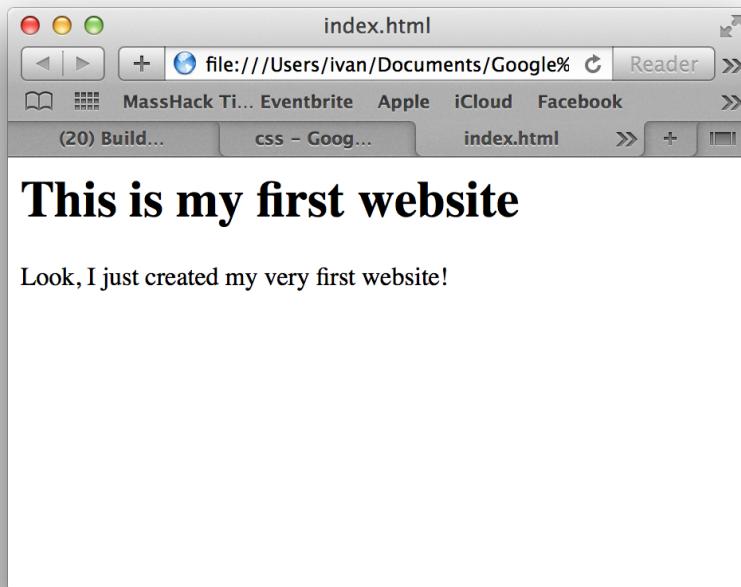
Your first HTML code

Now let's write your first lines of HTML code in file you just created. The first tags we'll look at are the heading and paragraph tags. There're 6 types of heading tags: `<h1>`, `<h2>`, ... , `<h6>`. `<h1>` defines the most important heading, `<h6>` defines the least important heading. The paragraph tag looks like this - `<p>` - and just represents one paragraph of the text page. Now, it's time to write some code!

Type this in your HTML file:

```
<h1>This is my first website</h1>  
<p>Look, I just created my very first website!</p>
```

Now, open this in your web browser (for example, Google Chrome) and you'll see a page like this:



Notice that you don't see your tags on the actual web page. You can see only content that you put inside these tags. Also, you can clearly see the difference between the heading tag and paragraph tag. As you can notice, different tags assign different styles for the content inside them.

HTML document structure

Although your code seems to be working well, it's not exactly correct. We have to add some more HTML tags which create the basic structure of HTML files. Before we do this, I want to say that tags can be nested inside each other! For example, using the **tag, you can make some text inside a paragraph bold: `<p>This is not bold. This is bold</p>`.**

This is the basic structure of all HTML files:

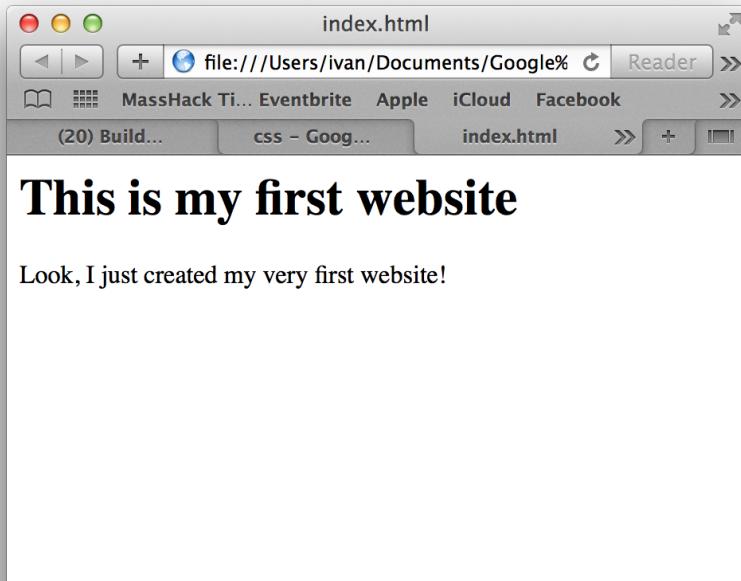
```
<!DOCTYPE HTML>
<HTML>
    <head>
        </head>
    <body>
        </body>
</HTML>
```

The first line of the code tells the browser that you are using the HTML language. If you don't write this, it will work fine in most cases, but it's better to write this for better compatibility. Next, the `<HTML>` tag encloses all your HTML code. Inside this you have the `<head>` and `<body>` tags. The head tag is created for putting some important information about your website which won't actually be displayed on the page. Finally, the `<body>` tag is the place where you write all your code that you want to display on the page.

Now lets add this template to the webpage we previously created:

```
<!DOCTYPE HTML>
<HTML>
  <head>
  </head>
  <body>
    <h1>This is my first website</h1>
    <p>Look, I just created my very first website!</p>
  </body>
</HTML>
```

If you save and open this page in your browser now, you won't see any changes on the page, but now your code is properly organized:



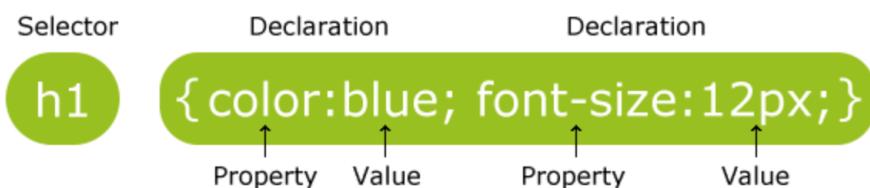
Congrats, you've just created your first website! I know it doesn't look like an actual website, but it's an amazing place to start!

Introduction to CSS

We just created a webpage, but it doesn't look pretty, right? Let's add some styles to it using CSS. Normally when you create a website, you write CSS in a separate file, but for now we'll write in the same HTML file that we previously created. For doing this, we have to create `<style>` tag inside your `<head>` tag. Remember, that you always need to close a tag!

```
<!DOCTYPE HTML>
<HTML>
    <head>
        <style>
        </style>
    </head>
    <body>
        <h1>This is my first website</h1>
        <p>Look, I just created my very first website!</p>
    </body>
</HTML>
```

Now we're ready to write some CSS code. As HTML consists of tags, CSS consists of selectors, properties, and values.



First, you have to choose an element to which you want to assign a style, then in curly braces, you can use property-value pairs to add some styles. Notice that property-value

pairs are separated by colons and you always have to put semicolon after each of the pairs. Any tag name is a valid selector that you can use to assign some properties to all tags with this name.

For example, lets make our heading red and paragraph blue:

```
<!DOCTYPE HTML>

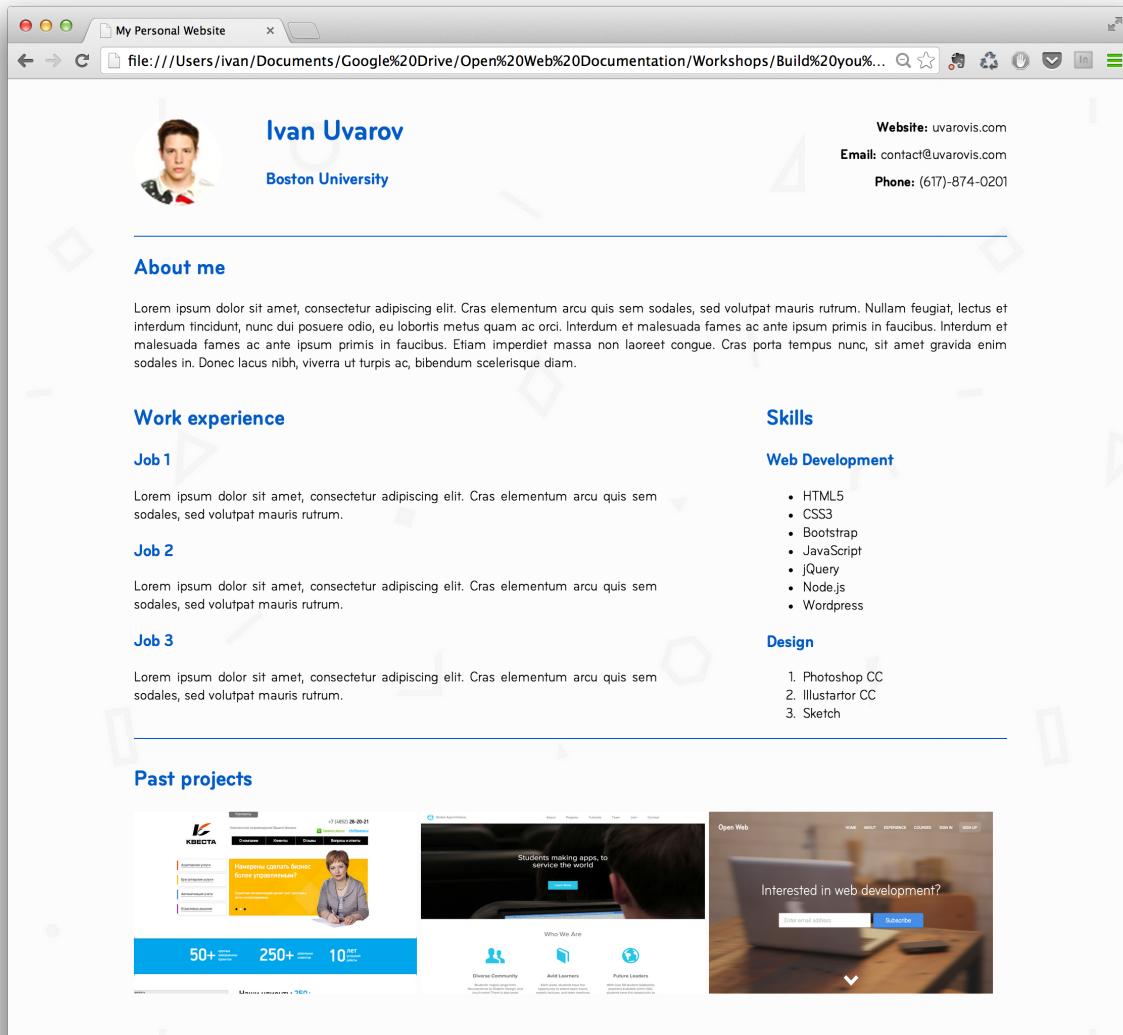
<HTML>
    <head>
        <style>
            h1 { color: red; }
            p { color: blue; }
        </style>
    </head>
    <body>
        <h1>This is my first website</h1>
        <p>Look, I just created my very first website!</p>
    </body>
</HTML>
```

Great! You're now familiar with HTML and CSS! Now, as you have a strong background, you can start learning about more advanced tags, selectors and properties. If you don't understand something now, don't worry, we'll put this to practice in the next part, so you can understand everything.

Creating your first website

Introduction to project

As we're already familiar with HTML and CSS, we'll start putting our knowledge in practice. Practice is the only way to properly learn web development. In this part we'll be building a personal portfolio website, which at the end will look like this:



It will introduce you to some basic HTML tags and to basic CSS attributes.

HTML tags

To begin, I'll give a short description of each HTML tag that we're going to use to build this webpage. You don't need to memorize them, because you'll eventually remember them after some practice.

`<title>content</title>` - in this tag you define the name of your webpage, which will be displayed in a browser toolbar. You use it once inside a `<head>` tag. Here's an example:

```
<!DOCTYPE HTML>
<HTML>
    <head>
        <title>The name of the website here</title>
    </head>
    <body></body>
</HTML>
```

`<div>content</div>` - defines a division or a section in an HTML document. This tag is used to group block-elements to format them with CSS. Basically, you use this tag to group some elements, so you can refer to the whole group with CSS. This is the tag that you'll see in a lot in all websites.

`` - this is the tag you use to add an image to your website. Notice, that it doesn't have a closing because obviously you cannot put anything inside an image. The src attribute is used to indicate the source of the file relative to the current HTML file.

`content` - anchor tag, which is used to create links. You put the address of the link inside the href property, and people will be able to click the content of this link to go to your link.

`content` - used to create a line of text

The last two tags are pretty much the same and they allow you to create unordered and ordered list:

```
<ul>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ul>
```

A diagram illustrating the mapping of an unordered list to a bulleted list. On the left, there is an `` tag followed by three `` tags: "First item", "Second item", and "Third item". A green arrow points from the third `` tag to the right, where the corresponding bulleted list items are shown: "• First item", "• Second item", and "• Third item".

- First item
- Second item
- Third item

```
<ol>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ol>
```

A diagram illustrating the mapping of an ordered list to a numbered list. On the left, there is an `` tag followed by three `` tags: "First item", "Second item", and "Third item". A green arrow points from the third `` tag to the right, where the corresponding numbered list items are shown: "1. First item", "2. Second item", and "3. Third item".

1. First item
2. Second item
3. Third item

Class and ID

Before we dive into coding, there's one more concept that we should learn. Remember that when we styled content with CSS before we used the tag selector? But what happens if you want to change one of the paragraphs of the page but don't want to change others? This is where classes and ids become handy.

You can assign a class or id to any tag you want. When you have some elements with the same properties on a page, you give to all of them the same class. When you have an element with a unique property, you assign an id to it. Then, in CSS you'll be able to change the style of elements with certain ids or classes.

Example:

```
<div id="header">content</div>  
<div class="project">content</div>
```

Making markup of the page with HTML

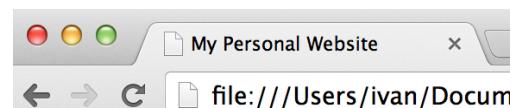
We'll start by creating file called `index.html`. This is the common name of the file for the homepage of your website. Inside this file we'll create basic structure of HTML document (just as we did before):

```
<!DOCTYPE HTML>  
  
<HTML>  
  <head>  
  </head>  
  <body>  
  </body>  
</HTML>
```

Firstly, we'll give a title to our page by adding a `<title>` tag inside of our `<head>` tag (new code is marked with **green** color):

```
<!DOCTYPE HTML>  
  
<HTML>  
  <head>  
    <title>My Personal Website</title>  
  </head>  
  <body>  
  </body>  
</HTML>
```

Now if you open this page in your browser,
you'll see that tab name has changed.



Now, we should make the top part of our portfolio website. To do this we'll create a div element and put all the content of top part inside it. To show your photo we're going to use the `` tag, for name, the `<h1>` tag, and `<h3>` tag for the university name (because it should be a bit smaller). For making contact information at the right, we'll create another `<div>` with 3 `` tags inside it. Also, we'll assign id and classes to some elements, so we can reference them in CSS later.

```
...
```

```
<body>
  <div id="top">
    
    <h1 id="name">Ivan Uvarov</h1>
    <h3 id="university">Boston University</h3>
    <div id="contact">
      <span>Website: uvarovis.com</span>
      <span>Email: contact@uvarovis.com</span>
      <span>Phone: (617)-874-0201</span>
    </div>
  </div>
</body>
...
```

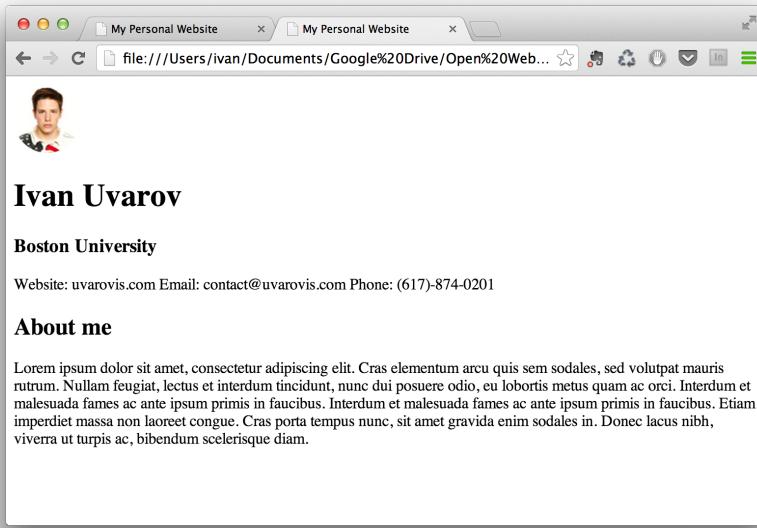
Moving forward, we should `<div>` our About block, because that is where we're going to create a title with a `<h3>` tag and a description with the paragraph tag `<p>`. As we don't have text for this paragraph now, we'll use some placeholder text that is commonly used by all developers. It can be found on the website lipsum.com.

```

...
<body>
  <div id="top">
    ...
  </div>
  <div id="about">
    <h2>About me</h2>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras elementum arcu quis sem
      sodales, sed volutpat mauris rutrum. Nullam feugiat, lectus et interdum tincidunt, nunc dui
      posuere odio, eu lobortis metus quam ac orci. Interdum et malesuada fames ac ante
      ipsum primis in faucibus. Interdum et malesuada fames ac ante ipsum primis in faucibus.
      Etiam imperdiet massa non laoreet congue. Cras porta tempus nunc, sit amet gravida
      enim sodales in. Donec lacus nibh, viverra ut turpis ac, bibendum scelerisque diam.
    </p>
  </div>
</body>
...

```

At this point your website should look like this:



In next part, we have 2 columns, so we need to create a `<div>` and 2 other `<div>` tags inside it:

```
<div id="columns">
  <div id="experience">
    content of the first column here
  </div>
  <div id="skills">
    content of the second column here
  </div>
</div>
```

The titles of both columns will be `<h2>` tags. In the first column for each job we'll use the `<h3>` tag for the title and a `<p>` tag for the description. In the second column we'll make `<h3>` tags for titles and use unordered and ordered lists (just to practice using different types of lists). Try to write code by yourself before looking at the following code. If you don't know how to do it, look at the code that I provided and retype it (not copy and paste) because it helps to understand the code.

```
...
<body>
<div id="top">...</div>
<div id="about">...</div>
<div id="columns">
  <div id="experience">
    <h2>Work experience</h2>
    <h3>Job 1</h3>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras elementum arcu quis sem sodales, sed volutpat mauris rutrum. </p>
    <h3>Job 2</h3>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras elementum arcu quis sem sodales, sed volutpat mauris rutrum. </p>
    <h3>Job 3</h3>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras elementum arcu quis sem sodales, sed volutpat mauris rutrum. </p>
  </div>

  <div id="skills">
    <h2>Skills</h2>
    <h3>Web Development</h3>
    <ul>
      <li>HTML5</li>
      <li>CSS3</li>
      <li>Bootstrap</li>
      <li>JavaScript</li>
      <li>jQuery</li>
      <li>Node.js</li>
      <li>Wordpress</li>
    </ul>
    <h3>Design</h3>
    <ol>
      <li>Photoshop CC</li>
      <li>Illustrator CC</li>
      <li>Sketch</li>
    </ol>
  </div>
</div>
</body>
...

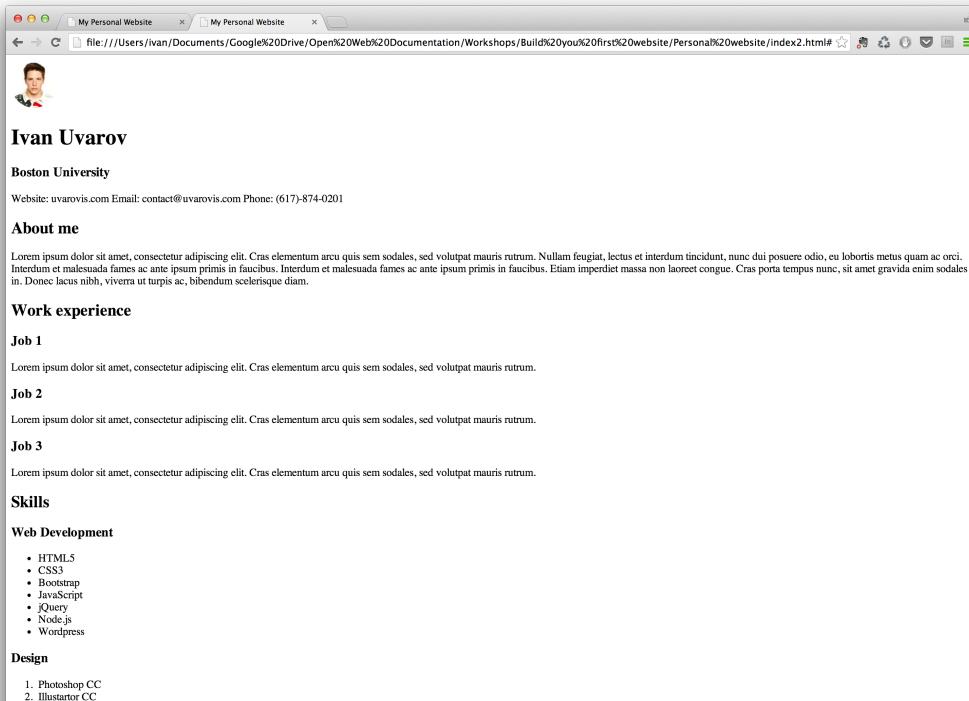
```

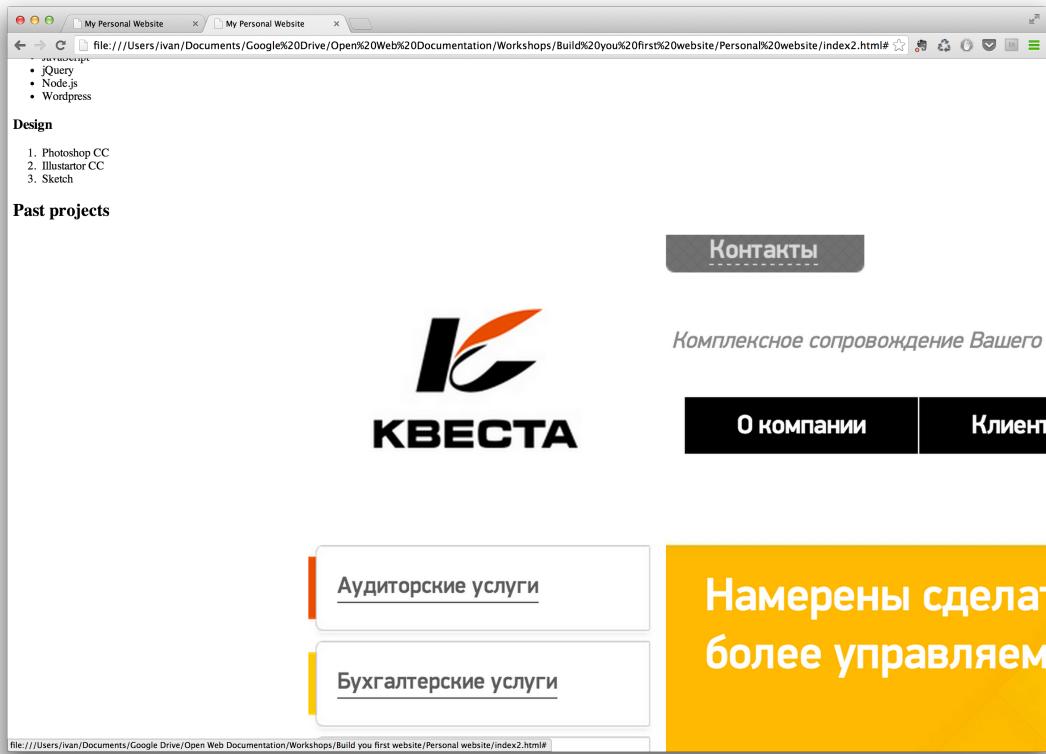
We're almost done with HTML! We just need to add the last part!

For this part we'll make a `<div>` with title and 2 images inside it. For title we'll use `<h2>` tag. To make it a bit more interesting, our images will also be links to corresponding websites. To make this you just need to wrap each image in `<a>` tag. To make placeholders for links you can use `href="#"`, which will make empty links.

```
...
<body>
<div id="top">...</div>
<div id="about">...</div>
<div id="columns">...</div>
<div id ="projects">
    <h2>Past projects</h2>
    <a href="#"></a>
    <a href="#"></a>
    <a href="#"></a>
</div>
</body>
...
```

Ok, we're done with HTML! If you did everything correctly (I'm pretty sure you did), you would see a page like this:





Congrats! You created amazing webpage. The only problem that it doesn't look pretty, but don't worry, we'll add some styles to it with CSS!

Styling your page with CSS

Writing HTML is the most boring part because after this your page doesn't look really good and it's not even close to the website we want to make. However, things become more interesting when we start writing CSS code and you start getting appropriate styles.

Lets make some quick changes in your HTML code, so we can style it properly. First thing you need to do is to wrap all content inside <body> into <div> with id="content". Then we'll be able to center the content on the website. Also, lets add <style> to the <head> of your HTML code, so we have a place to write some CSS code. Your code should look like this:

```
...
<head>
<title>My Personal Website</title>
<style>

</style>
</head>
<body>
<div id="content">
    <div id="top">...</div>
    <div id="about">...</div>
    <div id="columns">...</div>
    <div id="projects">...</div>
</div>
</body>
</HTML>
```

Now we'll write only CSS code in `<style>` tag. No more HTML. We'll start by styling our `<body>`. We'll set `text-align` attribute equal to `justify`, so all lines in our paragraphs will have equal width (just like lines in this file). Then we'll set our `font-family` attribute equal to `"Helvetica Neue"` to change the font on our website. Also, we'll change color of the text on the website to `#4B4B4B` by assigning this value to attribute `color`. And finally, we'll add `background` to our website by using a picture that we want to repeat. At this point you can feel that there're too many different attributes, but don't worry, you'll eventually become familiar with them.

```
<style>
body {
    text-align: justify;
    font-family: "Helvetica Neue";
    color: #4B4B4B;
    background: url("img/geometry.png") repeat;
}
<style>
```

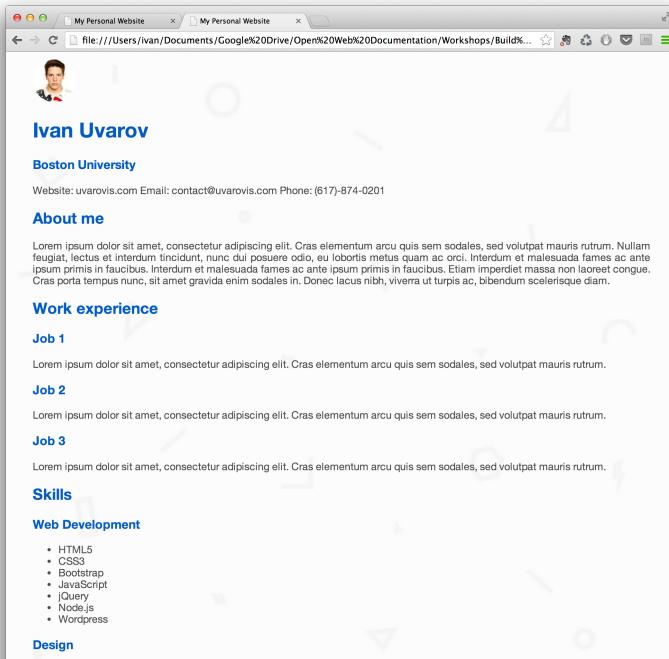
Then we'll center the content of our website by styling `<div>` with id equal to `content`. To do this we'll use the id selector: `#id_of_the_element`. By setting `max-width` we'll explain

browser that when width of the window is more than 960px, content width should not be increasing. Then, we'll center our content <div> using `margin` attribute. First value specifies that we want to have 0 margin at the top and bottom and automatically tells the computer that we want to have equal margins on the left and right sides.

```
body {...}  
#content {  
    max-width: 960px;  
    margin: 0 auto;  
}
```

Now let's change the color of all the heading tags on our page. We'll use the same color attribute that we used before. To select multiple tags (`h1`, `h2`, `h3`) we can just write all the selectors separated by commas:

```
body {...}  
#content {...}  
h1, h2, h3 {  
    color: rgb(14, 94, 197);  
}
```



At this point our page should me much more nice already:

Now lets work on top part of our webpage. We can see that photo is pretty small, so we should make it bigger. We can change size of it using `height` attribute. Also, we'll tell browser that all elements should float this photo. This will place your name and university name at the right hand side of the photo. We'll also add some margin to it, so we have some space between our photo and name.

```
...
h1, h2, h3 {...}
#photo {
    height: 100px;
    float: left;
    margin-right: 45px;
}
```

Next step is to move the `<div>` with contact information to the right top part. For this we'll need to tell the browser that all elements inside the `#top` div should placed relatively to it. We can do it with `position` attribute. Also, we'll add a position attribute to the contact part and tell it to have absolute positioning inside the `#top` div. After this we'll be able to tell it to move to the right top corner using `top` and `right` attributes.

```
...
#photo {...}
#top {
    position: relative;
}
#contact {
    position: absolute;
    right: 0;
    top: 0;
```



Ivan Uvarov

Boston University

Website: uvarovis.com Email: contact@uvarovis.com Phone: (617)-874-0201

```
}
```

As you can see, our website, email and phone are on the same line. To put them on separate lines, we should change their `display` attribute to `block`. All block elements simply move all elements after them to the next line. To choose `` inside `#contact` div we can use nested selectors: `#contact span`. We'll also change `text-align` and `line-height`, so it looks better.

```
#contact { ... }
#contact span {
  display: block;
  text-align: right;
  line-height: 30px;
}
```



Ivan Uvarov
Boston University

Website: uvarovis.com
Email: contact@uvarovis.com
Phone: (617)-874-0201

Now our top part looks amazing!

Moving next, we'll style About block. For this part we just simply need to add a margin at the top and some border at the top as well. For border we have to specify a type (solid, dashed etc.), size, and color.

```
#contact span { ... }
#about {
  margin-top: 50px;
  border-top: 1px solid rgb(14, 94, 197);
}
```

For the columns part, we need to make each column to be inline-block, so the first doesn't push the second to the next line. We also need to specify the width of the first column and the second one will take the rest of the space. Lastly, we'll add some margins to it. As you can see already, some attributes are repeated a lot, so you'll become familiar with them soon.

```

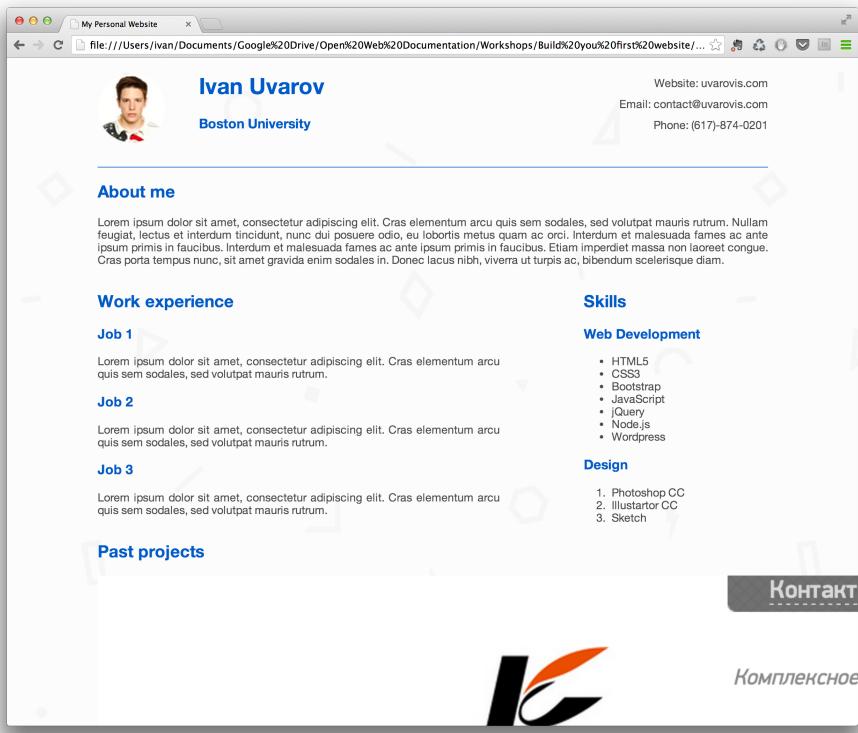
#about { ... }

#columns {
    position: relative;
    margin-bottom: 20px;
}

#experience {
    width: 60%;
    display: inline-block;
    margin-right: 120px;
}

#skills {
    display: inline-block;
    position: absolute;
}

```

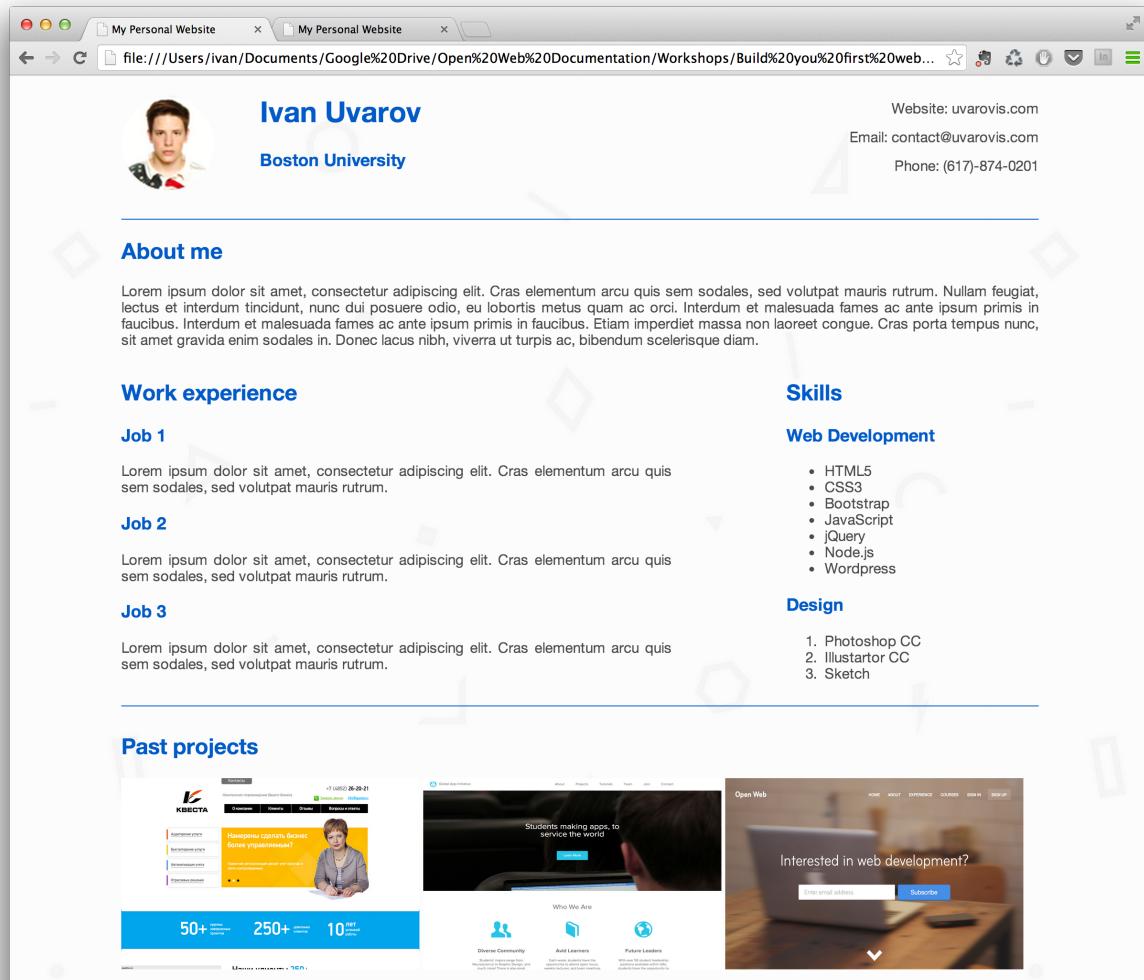


}

Good job! There's only one part left and we're done!

For the projects part we're going to create some border and padding at the top. Also, we'll change the image sizes.

```
#skills { ... }  
#projects {  
    padding-top: 10px;  
    border-top: 1px solid rgb(14, 94, 197);;  
}  
  
#projects a img{  
    width: 32.5%;  
    height: 200px;  
}
```



The screenshot shows a personal website layout. At the top, there's a header with a profile picture of a young man, the name "Ivan Uvarov", and the text "Boston University". To the right, contact information is listed: "Website: uvarovis.com", "Email: contact@uvarovis.com", and "Phone: (617)-874-0201". Below the header, there are sections for "About me", "Work experience" (with three subsections: Job 1, Job 2, and Job 3), "Skills" (with "Web Development" and "Design" subsections), and "Past projects" (displaying three thumbnail images of different websites). The overall design is clean and modern.

That's

all! You just created your first webpage with HTML and CSS!

Now you can see that it isn't really difficult to create a website. You can probably now feel that there're too many different tags and attributes, so you cannot remember them all or don't exactly understand what they're doing. It's totally ok, you'll understand everything after having more practice. I personally advise you to not even memorize all these tags and attributes, just try to make some webpages and google for whatever you don't know. When I created my first website, I didn't understand what half of my code meant, but after getting more experience, I just somehow memorized and understood everything.

Join Open Web and you'll get experience by working on actual websites even if you don't have any prior experience. The best way to do it is to learn as you go!

Useful resources

<http://www.w3schools.com> - all HTML and CSS documentation. Go to this website if you don't understand how something works.

<http://stackoverflow.com> - best websites to ask question about web development. There're a lot of people there who're ready to help you if you have any struggles.

<https://teamtreehouse.com> - one of the best websites with good video courses on web development. Perfect solution for people with no experience, all courses are easy to follow.

<https://www.codeschool.com> - another good website with web development courses, but this one is for people with a bit more experience.