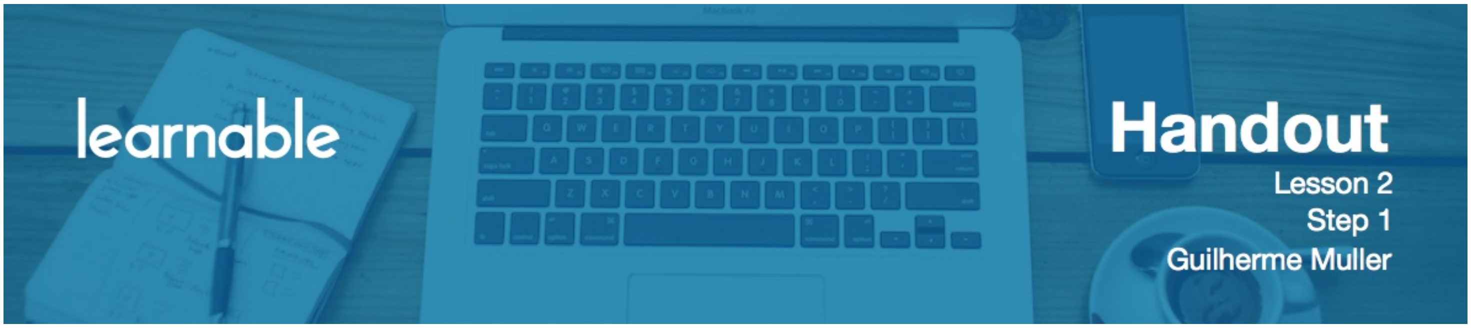

learnable

Handout

Lesson 2

Step 1

Guilherme Muller

Introduction

In this lesson we are going to learn about the main tags that can be used inside the `head` tag. To prepare for the lesson open the *"head-begin.html"* file.

The `head` tag is an important tag of a page. It can contain a variety of tags that provide meta information about the page, or load other files to compliment the HTML code like CSS of JavaScript files. We've already seen two of these tags before, the meta tag for meta data, and the title tag, to set the title of a page.

The meta tag

Let's talk a bit more about the `meta` tag. It's a very versatile tag and can be used for many purposes. In the following example it's being employed to set the character encoding of the page to `UTF-8`:

```
<meta charset="UTF-8">
```

It is also can be used to set a description of the page which search engines may use to display on search results page.

```
<meta name="description" content="Description about products and services - Company Name">
```

Try to write a description that is concise and contains the keywords you want to focus your website on; the same applies to the title tag as well. This kind of thinking is part of a range of techniques in **SEO**, or **Search Engine Optimization**.

The `meta` tag has many other uses. It can define the author of the page, or it can be used to confirm ownership of a website, like Google does. It can even help your responsive websites to load properly when browsed via mobile phones and tablets. The most important attributes however, are the `charset` and `description`.

The link tag

The `link` tag, despite the name, is not for linking, like you might think. It's a tag that can only exist in the `head` tag, and it's used to link to external files, loading them into the page. Its most common uses are for loading CSS files and favourite icons.

In the same folder as our *"head-begin.html"* file there is a *"test.css"* file with some simple CSS code that just changes the background of the page. If this is loaded correctly, the background of the page should change to grey.

```
<link rel="stylesheet" href="test.css">
```

As you see the `link` tag is also a void tag. The `rel` attribute specifies the relation of the loaded file to the document: in this case, it's a stylesheet. The `href` works just like the `href` attribute in the `a` tag, indicating where the file is located. In some cases, you can also find a `type` attribute here, specifying `text/css` as the value. This attribute, however, is only required if you are using XHTML in your page because of some requirements of the XML language. For HTML you can skip this attribute.

In the same folder there is also a *"favicon.ico"* file that can be used to display web site's icon in the browser tab.

```
<link rel="icon" href="favicon.ico">
```

Notice that the `rel` attribute has a new value.

The style tag

There is another way to include CSS in the document by dropping it to the HTML file itself. Of course, it's much better to use an external file, but in some cases it may be necessary to include styles right in the HTML file. We can do that with the `style` tag:

```
<style>
    /*
    -- CSS code here
    */

    p {
        background: white;
    }
</style>
```

The `style` tag should always be used inside the `head` tag. If you put it outside the `head` tag, it may still work, but your code would be considered invalid unless you use an attribute called `scoped`. But this attribute still has very poor support across browsers, so I recommend that you put style tags only inside the head section, and if possible avoid it all together by loading external CSS files instead.

The script tag

Finally, there's another very important tag called `script` that is used to contain JavaScript code. The `script` itself can be inside the page or it can be an external file. The `script` tag is used both to insert script in the HTML document and to point to external files whereas with CSS we have to use two different tags.

In the same folder as the *"head-begin.html"* file there is a *"test.js"* with very simple JavaScript code that displays a message to the user. Let's try to load this script in our document:

```
<script src="test.js"></script>
```

Notice here that I have to close the `script` tag unlike the `link` tag that we saw earlier. And to point to the file we used the `src` attribute like in the `img` tag.

To include scripts inside the HTML file, we also use the `script` tag: the only difference is that we write the code between the script tags:

```
<script>
    //javascript code
</script>
```

The `script` tag can be put inside the `head` tag but it can also reside in other places inside the document. If we place the `script` tag in the `head` section, the browser will have to load the scripts first and depending on the case, it can freeze the page for a bit, before loading the rest of the content. If there's an error while loading the scripts the page may not even load completely, so we have to be careful.

For that reason, it's advised to place the `script` tags just before the closing `body` tag, at the end of the page. This ensures that the whole page will load and only then the scripts will be loaded and executed.

```
<body>
    <script>
        //javascript code
    </script>
</body>
```