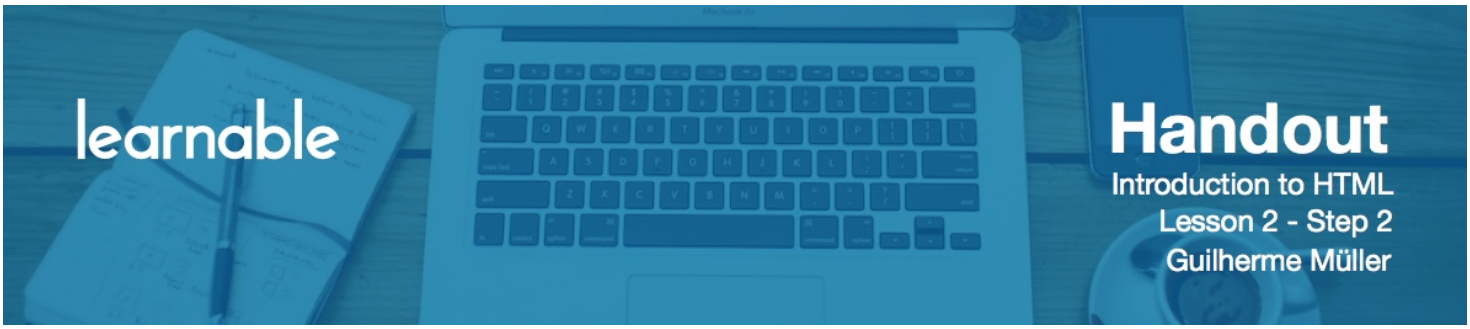

Introduction

HTML pages are common text files with the modified extension. You could simply create a `.txt` file, change the extension to `.html` and this will now change the file format to read as an HTML file.

Create an HTML file:

1. Open a Windows Explorer and create a new text file by right clicking in a folder and choosing New Text Document.
2. Name it `example.txt`. The `.txt` extension makes it a simple text file that could be edited in Notepad or in any other text editor.
3. Change the file's extension to `.html`. Your file should now be named `example.html`. This has changed the file format into an HTML file.
4. Open the file by double clicking on it. You should notice that a browser would attempt to open and render the file and display the HTML output. To edit the file you would need to open the file with a simple text editor.

If you had typed anything into the created document, you will see that the browser rendered the typed text. Anything that's not a tag will be interpreted as plain text. This is only one of the ways in which browsers are very tolerant with incomplete code and with code errors. Even this HTML file with no tags is being rendered without error messages visible to the user, even though we should have a minimal set of tags for the document to be a valid HTML document.

I want to emphasize that everything we are going to do regarding proper code structure, best practices and semantic HTML, is something we do because we want to produce quality code and minimize compatibility problems. The browser itself will not show us any errors in the code explicitly. We could write wrong and incomplete code, and even in such case the browser would try to render the page without problems. Of course, we as good developers, we will always try to code valid, semantic, and correct HTML code.

Doctype and HTML tags

There are certain basic HTML tags that should be included into every HTML document.

Doctype

This tag used to be much more complex before HTML5 but now we can only write:

```
<!DOCTYPE html>
```

This tag served other purposes in the past, but now it only shows the browser that we are working with an HTML file. This tag does not close like most of the other tags.

HTML

This is what we call the root element, the tag that contains all the others. It's good practice to indicate in what language the contents of the page will be written. That can be done by using a `lang` attribute of the tag. As we have seen previously, an attribute is like a characteristic of the thing.

```
<!DOCTYPE html>
<html lang="en">
```

You can even be more specific by writing the region like `en-US` or any other abbreviation that follows the national language support (NLS) standard. Now we can close the HTML tag like this:

```
<!DOCTYPE html>
<html lang="en">
</html>
```

HEAD

Inside the `html` tag, the first tag should be `head`. The `head` tag contains information about the document and links to other files. It is not the visual header area of a website, but the header of the document itself:

```
<!DOCTYPE html>
<html lang="en">
<head>
</head>
</html>
```

If you want, you can indent this tag using the TAB key to show that this tag is inside `html`:

```
<!DOCTYPE html>
<html lang="en">
    <head>
    </head>
</html>
```

This is one of the techniques we can use as a coder to visually organize the code. I personally like to indent all tags that are inside of another except for head and body, which I'll show you in a second.

Inside the `head` tag it's important that we begin by specifying the document's encoding. Encoding tells the browser what standards we are using when it comes to interpreting characters you've used in the text. English text uses the Roman alphabet plus numbers and other symbols, for example. We should note in the page which encoding the page is using, so that the browser knows what characters to use to represent that page's content. We'll talk more about that in our later lesson. We can specify page's encoding with a `meta` tag.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
    </head>
</html>
```

This is a tag that has become much simpler in HTML5. `meta` is what we call a **void element**: it does not have neither a closing tag nor content so we only work with its attributes. In this case, I used the `charset` attribute, but `meta` is a versatile tag that can serve many other purposes. This tag means that I'm using UTF-8 in this document, which is basically Unicode. Unicode will cover all the main characters that are employed in English and many other languages that use the Roman alphabet, including languages that use accents and other symbols. So this encoding should present no problems most of the time. But we have to ensure that the document itself is also saved with this encoding, otherwise we'll have problems. Most text editors have options for setting the document's encoding. This encoding must match what we wrote in the `meta` tag.

The title tag

Following the `meta` tag, we can now add the `title` tag:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
</html>
```

This is the title of the page which appears on search engines and in the browser tab. Let's name our page "Example HTML page":

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Example HTML page</title>
    </head>
</html>
```

If you now save the file and then reload it in the browser, you'll note that the title is appearing in the browser tab.

This is the minimum you should have inside the `head` tag.

The body tag

Now, let's create another important tag, `body`, that will contain all of the document's contents:

```
<!DOCTYPE html>
<html lang="en">
```

```
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
    </body>
</html>
```

Body is where most of the work on the HTML code will go. For now, let's only write the classic "Hello, world!" message:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        Hello, world!
    </body>
</html>
```

Now save and reload the page. We have a working HTML page. This code can serve as a blueprint for other pages, as this is the minimal recommended code for an HTML document.

Adding comments

Finally, when we want to add comments in the document for ourselves or for other developers, we can use the special comments tag like this:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        Hello, world!
        <!-- example comment -->
    </body>
</html>
```

This will only appear in the code and will not be shown by the browser.