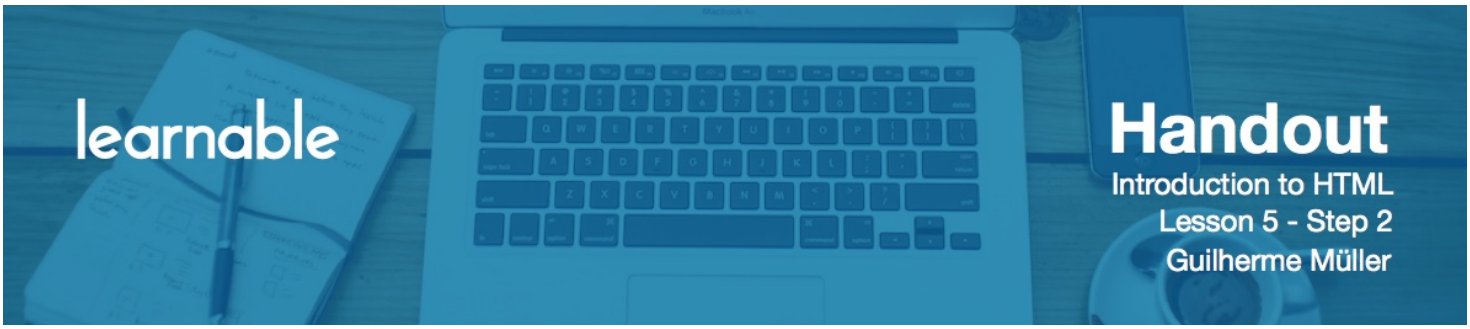

Introduction

In this lesson we are going to see the importance of code validation and having clear and semantic code. Not all problems are solved with validation, but a valid HTML file is a good place to start.

Open up the *"html-invalid.html"* file to prepare for the lesson. Also visit the http://validator.wc3.org page that presents a free online validator by W3C. It can be used to validate files by the URI, by file upload or by direct input.

Overview of some common coding mistakes

In the first example of invalid code the `style` tag is put outside of the `head` tag:

```
<body>
    […]
    <style>
        body {
        background: #EEE;
        }
    </style>
    […]
</body>
```

We may use the `style` tag inside the `body` tag but with the `scoped` attribute, however this attribute does not have a good support in browsers, moreover it serves a more specific purpose.

Next, we have tags with improper nesting:

```
<p>
    Invalid: text with <strong><span>improper nesting</strong></span>
</p>
```

Notice here that the `strong` tag starts first and then it's being closed first as well. The `span` is inside the `strong` tag but it's closing last.

It is also invalid to wrap a block element like `p`, `div` or headings with an inline tag, like `strong`, `span` or `em`:

```
<span>
    <p>
        Invalid: span containing p tag. Inline tags cannot contain block tags (except the <code>a</code> tag). Inline tags include
<code>span</code>, <code>em</code>, <code>strong</code>. Block tags include <code>div</code>, <code>section</code> (and all other
structural tags), <code>h1</code> to <code>h6</code> and lists (<code>ul</code>, <code>ol</code>).
    </p>
</span>
```

Inline elements in most cases can contain only text or other inline elements, but not block tags like in this example. Block tags usually can contain other block tags and inline tags. The only exception is the `a` tag which can wrap around elements like headings and paragraphs in HTML5.

Of course, it's also invalid to use tags that are not present in the HTML specification:

```
<invalid>Invalid: non-existent tag</invalid>
```

Another case that happens sometimes is that a tag ends up but not closing:

```
<div>
    Invalid: div lacking closing tag
```

This will cause major problems in the website.

Surprisingly, these two pieces of code are valid:

```
<p>Valid: some tags do not necessarily need a closing tag (though it's best practice to close all tags). These tags include
<code>p</code> and <code>li</code>.

<p id=example>
    Valid: attribute values (with one word) without quotes (though it's best practice to quote all attributes).
</p>
```

Although this is not recommended practice, you can, for example, leave a `p` tag open depending on the tag that comes next. Also if attributes have one word values they do not necessarily need to be quoted, however you should always quote attribute values.

Before running this code in the validator you may open the document in the browser. Browsers are extremely forgiving when it comes to errors in the code. For example, the CSS code would still be applied to the page and all content would be shown in the best way possible. This means that we as web developers have the most responsibility when we seek to write valid and semantic code. To ensure better compatibility between browsers and devices, and better user experience.

Code validation and correction

So now run this code in the validator. You should receive seven errors and two warnings. The warnings can be ignored. The first one is saying that the validator is using experimental features and the second one concerns the direct input method for validation: they are assuming `UTF-8` is the encoding and that's what we're using in the page.

Let's analyse the errors now:

- *"Element style is missing required attribute scoped"*. Like we've discussed, this attribute can be added to the `style` tag, but that is meant for another use of the tag. The better solution here would be to move this tag inside the `head` tag, or to an external file.
- *"End tag strong violates nesting rules"* - this is related to the improper nesting example.
- *"Stray end tag span"* - this is directly related to the previous error.
- *"Element p not allowed as child of element span in this context"* - this is related to the `span` tag wrapping around the `p` tag. To solve this, we must invert the nesting or remove the `span` tag.
- *"Element invalid not allowed as child of element body in this context"* - we cannot create new elements that do not exist in the HTML specification.
- *"End tag for body seen, but there were unclosed elements"* - the validator is warning us that there are unclosed tags found.
- *"Unclosed element div"* - the validator informs us that there's a `div` tag that is unclosed.

Code revalidation

You may now correct these errors and validate the code once again. The document should be passing the validation now.

We have seen some cases of errors in the HTML code, how to correct them, and how to run the code through the W3C validator. Valid HTML files are not a guarantee of a good website, but they help to ensure compatibility across browsers and devices. The validator does not evaluate semantics however, so, that's up to the web developer to interpret and properly mark up the website's content.