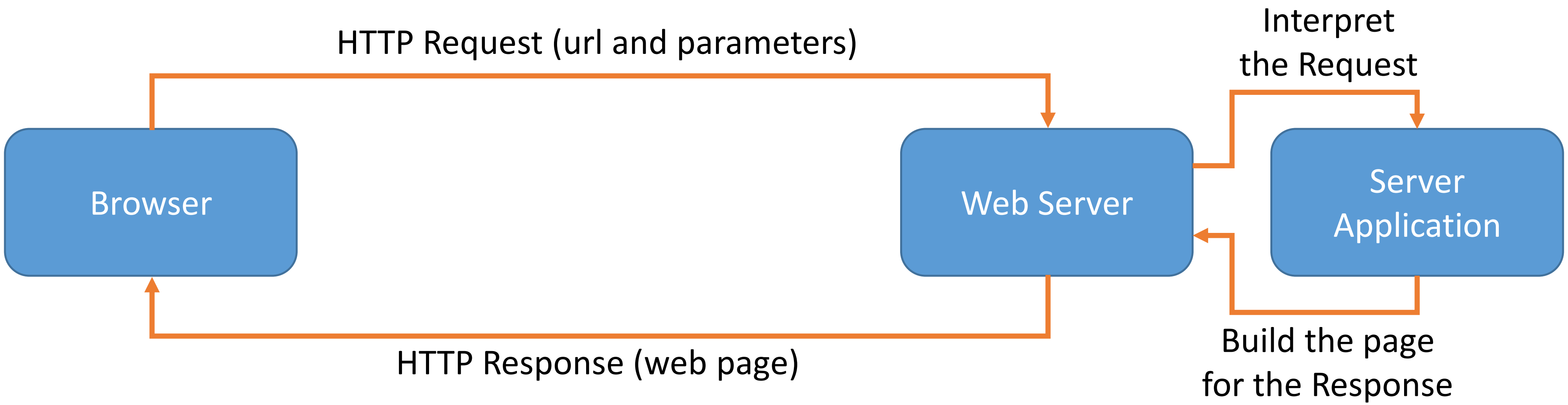
Introduction to HTML & CSS

HTML - HyperText Markup Language

Every web page is made up of HTML code, in one way or another. Whenever you nagivate onto a page, you tell the browser to connect to a server using the HTTP (HyperText Transfer Protocol) protocol and download a HTML page as a response. This cycle of **request** and **response** is the web at its most basic.



HTML - Elements, Style and Behaviour

A HTML page built according to the current standard – HTML5 – has three main parts: HTML, CSS and JavaScript.

The HTML code controls **what exists** on the page – the **elements** themselves, defined as tags with **attributes** describing their function.

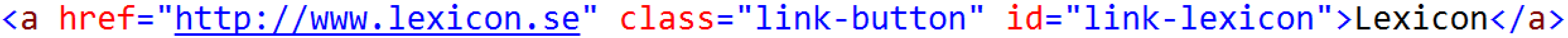
CSS determines the **styling** of the page, how elements look once they are displayed in a browser window.

JavaScript is used to control the **behaviour** of the page’s elements. Any element that changes after the page has been loaded, such as animations, is done through JavaScript.

HTML Elements - Element Tags and Attributes

Elements on the web page are defined as HTML **tags**. These are defined in two ways – either with a start and end tag (**<element>** and **</element>**), or with a self-terminating tag (**<element/>**).

Tags also have **attributes** that handle any additional info that is attached to them.



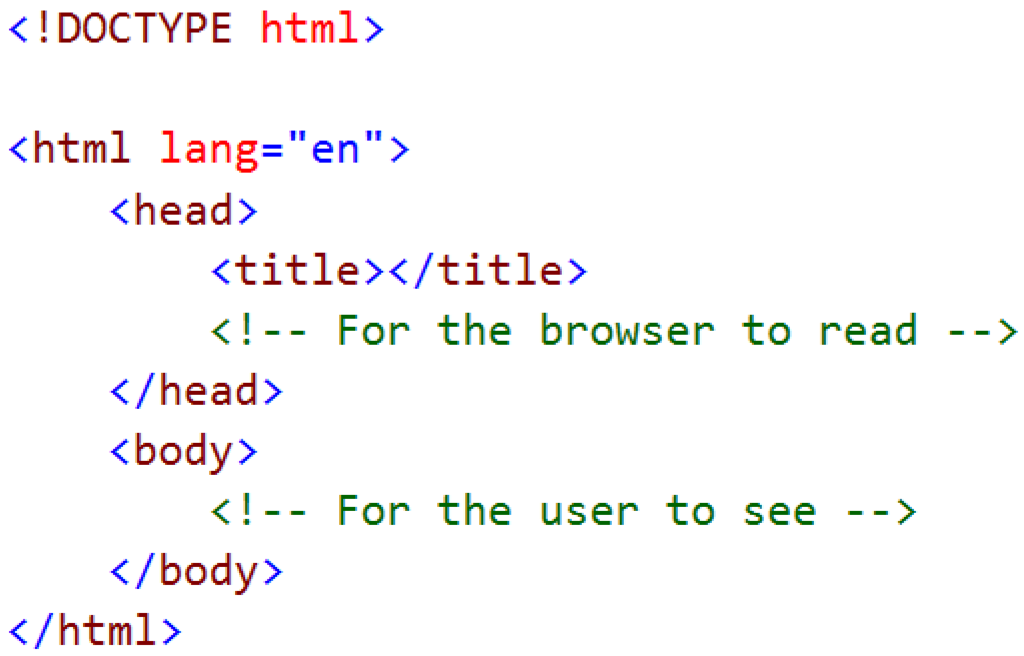
```
<a href="http://www.lexicon.se" class="link-button" id="link-lexicon">Lexicon</a>
```

Here, we define a **anchor** (clickable link, through the **a** tag) element, that points at the *http://www.lexicon.se* URL through the **href** attribute, has a style **class** of *link-button*, and an **id** of *link-lexicon*.

HTML Elements - Html, Head and Body

The three basic elements that you will see in any HTML page are <html>, <head> and <body>. These three define the basic structure of a html file.

The **<html>** tag define the *beginning and end* of the page itself. The **<head>** include information that is meant to be *interpreted by the browser*, while anything contained inside **<body>** define what is actually *shown on the page*.



```
<!DOCTYPE html>

<html lang="en">
    <head>
        <title></title>
        <!-- For the browser to read -->
    </head>
    <body>
        <!-- For the user to see -->
    </body>
</html>
```

HTML Elements - Useful Tags

There are a lot of different types of elements, with corresponding tags and attributes. There are some that are very good to know about:

Containers Containers are ”sections” of the page, which contains other elements.
<header>, <footer>, <nav>, <article>, <aside>, <section>, <div>

Text Text is written either straight into a tag, or into a paragraph or span.
<p> for paragraph, <span> for words or sentences inside a bigger text.

Images <img src=”fileURL.jpg" /> puts an image element onto the page.

There are too many tags to list them all – I have listed a few more in the cheat sheet for this lecture. Otherwise, refer to http://www.w3schools.com/ as a reference.

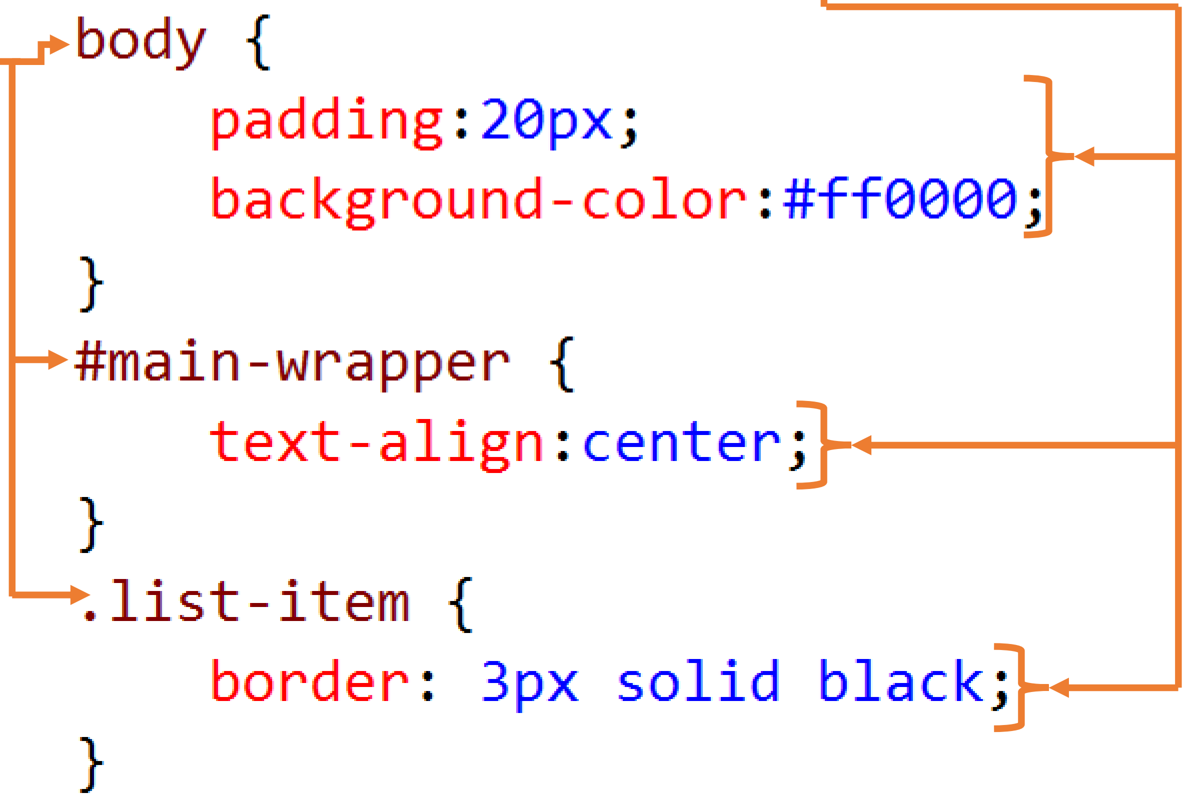
CSS - Style

CSS applies styling rules to the page. It does this through **style rules**, which is applied though **selectors**.

Selectors are normally appied to **class** and **id** parameters.

Use # to mark a selector as selecting by id, and . to mark it as selecting by class.

As with the HTML tags, refer to the cheat sheet and W3Schools for more tags.

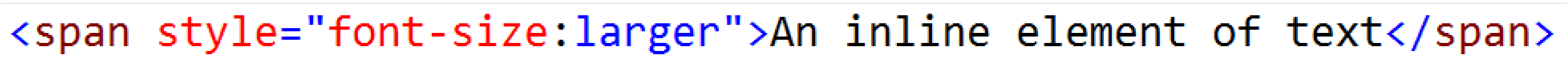


```
body {
    padding:20px;
    background-color:#ff0000;
}
#main-wrapper {
    text-align:center;
}
.list-item {
    border: 3px solid black;
}
```

CSS - Including CSS in the page.

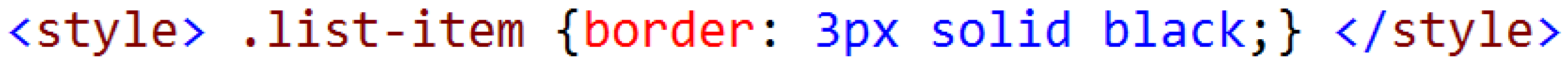
Once CSS code has been written, it must be included in the page. This can be done in three ways.

You can apply style rules to an element with the **style attribute**.



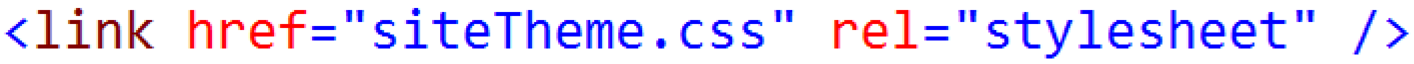
```
<span style="font-size:larger">An inline element of text</span>
```

You can also implement the styling rules in a **style tag**, with selectors.



```
<style> .list-item {border: 3px solid black;} </style>
```

The better way, which you should use as a rule, is to **link a .css file** into the html code, using the <link> tag inside the <head> element.

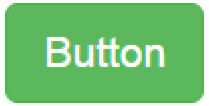


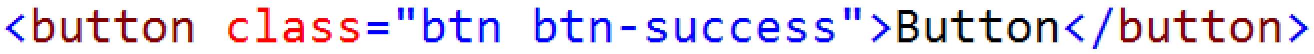
```
<link href="siteTheme.css" rel="stylesheet" />
```

CSS - Bootstrap

Because CSS is often fairly complex, and uses many different parts to achieve more or less the same result, it is common to use pre-made frameworks for it.

Bootstrap is one such library that focuses on responsive websites that work for screens of all sizes, including mobiles and tablets. The basic premise of it is a collection of CSS classes that can be applied to elements to give them a predefined styling:





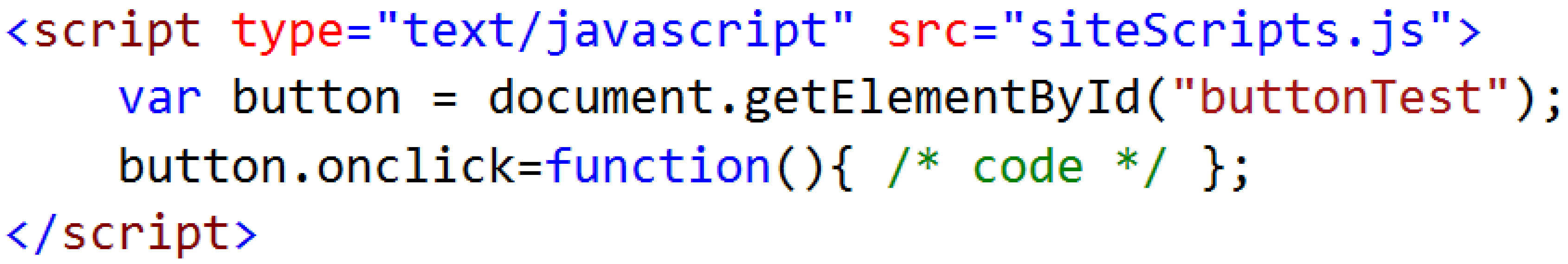
```
<button class="btn btn-success">Button</button>
```

Refer to W3Schools and http://getbootstrap.com/ for more specifics.

JavaScript - Dynamic Changes

JavaScript is a scripting language, code that runs inside another program (in this case, the browser) as opposed to the operating system.

There are a great deal of things you can do with JavaScript - you can practically make complete applications that run inside the browser.



```
<script type="text/javascript" src="siteScripts.js">
    var button = document.getElementById("buttonTest");
    button.onclick=function(){ /* code */ };
</script>
```

This, for instance, puts a method in place that is going to run when the element with the ”buttonTest” id is clicked.