**Week one slide notes: Course introduction, HTML history, Introducing basic tags, attributes and elements.**

**Course introduction**

Understanding HTML and CSS can help anyone who works with the web; designers can create more attractive usable sites, website editors can create better content, marketers can communicate with their audience more effectively.

I have developed this course as an introduction for complete beginners. I understand that code can look intimidating, but by the end of this course you will have built a simple website from scratch and will have the foundations to develop further on these skills.

My name is Sue. I first became interested in website development when I studied for a degree in IT. At that time I was also working part time in an admin job and, once I had completed my degree, needed to return full time to my admin role. I had decided that this was what I really wanted to do though and, over time, started to build up a portfolio of websites.

I would build them for friends who were artists or my Mum’s friend who had a wedding dress shop. Most of these were done free of charge as I just wanted to get the experience.

Once I had a few websites built, I applied for a job as a junior front end web developer at a company called New Mind | tellUs. They took a chance on me, and working for a web agency gave me lots of skills. Since then, I have worked on the Graduate Prospects website and am currently working as a frontend web developer for a company called Twist Digital, which I love doing.

I hope that once you have finished this course some of you might be inspired to get into the world of web development - there are lots of opportunities and jobs available if you are willing to work hard for it, and it is never boring (well, I don’t think it is).

The objectives for the course is to develop a fictitious website project that will be completed over the six weeks: a mini website for who better than Liverpool Girl Geeks:

The website will include:

* The logo and a background image.
* A navigation linking to another page.
* A list, which will link to recent blog posts on the Liverpool Girl Geeks Wordpress site.
* A nice, two column page layout.
* A table.
* A simple ‘Contact Us’ form.
* And then some nice CSS3 effects.

The plan for this evening is as follows:

* I will start by giving you a bit of background into the history of HTML, web standards and we will go through some useful terminology
* The only tools that you will need for this course are a computer with a web browser and a text editor, so we will make sure that everyone has this set up on their computer before we are all ready to continue.
* I will then introduce basic tags, attributes and elements in HTML and we will have a go of setting up a first web page.

**A history of HTML, HTML standards, browser compatibility slide notes:**

Before we look at the code used to build websites, it’s important to have a little bit of a background:

In order to understand the history of HTML, it’s vital to understand the differences between the World Wide Web and the internet. The internet is a series of huge computer networks that allows many computers to connect and communicate with each other globally. Upon the internet, there are a series of languages which allow information to travel between computers. These are known as protocols, and HTML is one of these protocols.

In 1989, British scientist Tim Berners-Lee invented the World Wide Web whilst working in a computer services section of CERN in Geneva Switzerland. CERN is the European Laboratory for Particle Physics, which often involves collaboration among institutes from all over the world. Berners-Lee had the idea of the World Wide Web as a way of sharing scientific and mathematical documentation across computers.

People access websites using software called web browsers. The first web browser was also developed by Berners-Lee in 1990. He shared his findings as he wanted the web to be free for everyone. Other browsers such as Mosaic were then developed in the 1990s.

In 1994, a browser called Netscape was released, and became so popular that it began inventing its own HTML tags without openly discussing them first with others in the web community. This, however, created inconsistency between browsers.

Because of this, Berners-Lee founded the World Wide Consortium (W3C), recruiting many of the best known names in the Web community to advocate best practises for building websites. Adherence to web standards refers to the development of web pages that validate according to the W3C recommendations. W3C’s mission is to lead the web to its full potential, meaning HTML is an ever evolving language. With each new version, standards change to allow cleaner, more efficient code, which contains even more features than before.

Popular examples of browsers we use today include Firefox, Internet Explorer, Safari, Chrome, and Opera. Software manufacturers regularly release updated versions of browsers with new features. It is important, however, to remember that many computer owners will not be running the latest versions of these browsers. So therefore, don’t rely on all visitors to your site being able to use the latest functionality offered.

When you ask your browser for a web page, the request is sent across the internet to a web server, which is a powerful computer that is constantly connected to the Internet, and is optimised to send web pages out to people who request them. Some big companies run their own web servers, but it’s common to use the services of a web hosting company who charge a fee to host your site.

All websites use HTML and CSS to define the structure and presentation of a webpage, but content management systems, blogging software and e-commerce platforms often add more technologies into the mix. Larger, more complex sites may use a database to store data or have interactivity that cannot be achieved with HTML and CSS alone. JavaScript, PHP, ASP, .Net and Ruby are all programing languages that can be used for more complex sites. All of these technologies are advanced topics that you can go on to learn more about once you have mastered the basics of HTML and CSS. But ultimately, if you’re looking to build websites, you should know about HTML for the structure and CSS for the presentation. The skills you’ll learn in this course should be enough to help you on that road.

Since the web was first created, there have been several versions of HTML and CSS, each intended to be an improvement on the previous version. The Web Hypertext Application Technology Working Group (WHATWG) is a community of people interested in evolving HTML and related technologies. It was founded in 2004 by individuals from Apple, the Mozilla Foundation and Opera Software, in response to the slow development of the World Wide Web Consortium (W3C). In 2007, the WHATWG and W3C agreed to use the proposed WHATWG HTML5 working standard. Many browsers are supporting these new features, and a lot of developers are using the latest code on their websites, which is why we are going to use the latest versions, HTML5 and CSS3, in this course.

HTML are simple text files so, in order to start writing in HTML, you need nothing more than a simple text editor. Whilst there are lots of editors available offering extra features, and software applications you can pay for such as Dreamweaver offering some useful and timesaving elements, I believe the best way to start writing is with the simplest text editor available. That way, you get to understand the work entirely as you do it - plus, they are so simple that little can go wrong! Both Windows and Mac come bundled with simple text editors; Windows using Notepad and Macs use TextEdit.

For this course, we will be using Google Chrome. As we discovered earlier, there can be inconsistencies with browsers. Different companies introduce different features, or people can have outdated browsers, which can create inconsistencies in the results. So, for the purposes of this course, we will all use the same Google Chrome browser, as it is available on both the PC and the Mac, and is one of the most standards-compliant browsers out there.

Cross-browser issues are beyond the scope of this course but, in short, it usually means testing on each platform to see if your website is displaying the same on each browser. These are best learnt once you have learnt how HTML and CSS work in the browser. Otherwise you won’t know if something isn’t working because of the code or due to the browser.

**Introducing basic tags, attributes and elements, slide notes:**

HTML stands for Hypertext MarkUp language and is the standard markup language used to create web pages. We’ll be using a combination of HTML and CSS to create our website. CSS is a language that lets you control how your web pages look, but we’ll go over that in more detail later in the course.

HTML is a markup language that describes the structure and semantic content of a web document. Content within a web page is tagged with HTML elements. These elements form the building blocks of a website.

**Tags**

When creating a web page, you add tags (known as markup) to the contents of the page. These tags provide extra meaning and allow browsers to show users the appropriate structure for the page.

Tags act like containers. They tell you something about the information that lies between the opening and closing tags.

<p></p>

A HTML element starts and ends with tags - the opening tag and the closing tag. A tag consists of an opening angled bracket <, some text, and a closing bracket >. The closing tag has a forward slash after the < symbol. Make sure to close the tag, as it might produce unexpected errors if you forget the end tag. You may use uppercase for tag names, but the W3C recommends using lower case.

Inside a tag, there is a tag name which indicate the tag’s purpose. In this example, the p stands for paragraph. There may also be one or more attributes.

**Elements**

HTML consists of a set of elements, which describe the structure of the pages. Each element has an opening tag and a closing tag. Tags tend to do little more than mark the beginning and end of an element. The HTML element is everything from the start tag to the end tag.

<p>Liverpool Girl Geeks</p>

**Empty elements**

Also known as self-closing elements. Some elements do not contain any text content or any other elements. These are empty elements and need no closing tag. For example:

<img src=”myimage.jpg” alt=”my image”>

**Attributes**

The start tag may contain additional information. This is called an attribute. Attributes usually consist of:

* An attribute name
* An attribute value

For example, src is the attribute name and the value is “myimage.jpg”

**Your first web page**

doctype

<html> tag

<head> tag

<title> tag

<body> tag

These requirements make up the basic skeleton of the web page.

**The Doctype**

<!DOCTYPE html>

This is known as the doctype, which is short for Document Type Definition. It must be the first item on a web page, and lets the browser know which version of HTML you are using, which in our case is the latest version of HTML5. The doctype tells the browser to interpret HTML and CSS code according to W3C standards.

**The HTML element**

This is the outermost “container” of our web page; everything else (apart from the doctype) is kept within that container.

There are two major elements inside the HTML element: the head and the body.

Lets look at the head first.

**The head element**

The head element contains information *about* the page*,* none of this information will be displayed on the page itself.

**The title element**

You will usually find a <title> element inside the <head> element. The title element tells the browser what to display in its title bar. If users decide to add the page to their bookmarks (or favourites), the title will be used to name the bookmark.

**Meta elements**

Inside the head element, we can see a meta element. A meta element can be used in a web page for different reasons. Some are used to provide additional information that’s not displayed onscreen to the browser or search engines; for instance, the name of the page’s author might be included in meta elements. The meta tag tells the browser to use the UTF-8 character set, which includes the characters needed for web pages in just about any written language.

**Other head elements**

Other items, such as CSS and JavaScript code can appear in the head element. We will be discussing CSS next week.

**The body element**

The pages body element contains almost everything that you see on the screen: headings, paragraphs, images, any navigation that’s required, and footers that sit at the bottom of the page.

Examples of basic elements that could be contained within the body elements are headings and paragraphs.

Let’s go through headings first:

HTML has six levels of headings.

<h1> is used for main headings

<h2> is used for subheadings.

If there are further sections under the subheadings then the <h3> element is used and so on. Browsers display the contents of headings at different sizes. The contents of a <h1l> element is the largest, and the contents of an <h6> element is the smallest. The exact size at which each browser shows the headings can vary slightly. Users can also adjust the size of text in their browser. You will learn how to control the size of text, its colour and the fonts used when we come to look at CSS.

**Paragraphs**

To create a paragraph, surround the words that make up the paragraph with an opening <p> tag and closing </p> tag. By default, a browser will show each paragraph on a new line, with some space between it and any subsequent paragraphs.