Code First: Session 1 Web Basics

What makes a website?   
Website = collection of linked web pages (and associated resources). Website is the domain name, e.g tabs and links on Facebook open new web pages.

Webpage = HTML is a mark-up language, just to put the content in and style the content, styling is in CSS.

Page styles are stored in CSS

JavaScript is used for user interactivity scripts.

Terminology:  
Clients – your laptop

Servers – always connected to the Internet, requests sent to server, which then finds the IP address. Where the website is hosted.

Internet

Routers – “ifconfig” lists laptop IP address, “lo” is local host four numbers with dots between. Routers used to store IP address. Routed to the server.

Modems – Main device to connect to internet.

ISP -

URL – Uniform resource locator. Unique for every company. Three parts; protocol (before the slash, these are rules http or https (secure)), domain name (e.g [www.facebook.com](http://www.facebook.com) owned by Facebook), path (unique to you, e.g. your profile).

DNS – tracks IP address and locates it, domain name system. Static (cannot change them e.g. wikepedia) vs dynamic sites (can request whatever you want e.g. facebook)

Client-side and Server side Technologies:

Client side: JavaScript, HTML, CSS.

Server side: PHP (some domain providers charge you), ASP (active server pages) etc.

HTML:

< >anything between these is a tag. Put all information in between tags.

<h1>Hello</h1> biggest font

Always close with slash

Session 2:

In terminal (spotlight) go to shell>homebrew. Used for automating your tasks, 1 or 2 or 3 or 4 etc line scripts.

CSS = cascading style sheets. Style of html. Presentation in three ways; 1. Can put inside the html under the body. 2. Put it in the head tag at the top <head> <style> ------ </styles> </head>, this applies it to the whole thing, TIME CONSUMING. 3. Inside the head tag rather than writing the code can import a separate file usually done in link tags <link> ----- </link> BEST ONE

Properties in {} and start with a : and end with a ;

<style>

h1{color:red;}

p{color:blue;

font-family: 'Helvetica';

font-size:16px;}

h2{color:green;

background-color: orange}

ul{font-family: 'Calibri';

color:purple;

background-color: yellow;}

ol{color:pink;

font-style:italic;

background-color:teal;}

ID is unique to tags but CLASS is the same for all. Eg. This class should have a yellow background. No spaces in the names

ID is # and class is dot

ul#Unordered\_list{color:purple;}

li.Unordered\_list\_item{background-color:yellow;}

</style>

</head>

<body>

<h1>Make me red!</h1>

<p>I am a paragraph I should be blue, Helvetica, size 16px.</p>

<ul id="Unordered\_list">

<li class="Unordered\_list\_item">This is an unordered list.</li>

<li >All items in this list should have a yellow background.</li>

<li class="Unordered\_list\_item">Just the items though - not the whole list!</li>

Division applies to a complete block. Inserts one empty line before and like a list /paragraph

Span applies to a certain bit of text. Just span this btit of text

Links are like this :

<link rel="stylesheet" type="text/css" href="stylesheets/mypage.css">

can also put a url as a website

Session 2:

API: application, programming, interface. Cannot access data directly because of copyrighting, all those websites have their own APIs. E.g. in Google you need to create an account to create an application.

E.g. Facebook can use Soton.ac.uk email address to access the group.

Framework: like an umbrella with APIs and libraries which you can use in your application. Any framework has it’s own prebuild css and javascript files. Can copy frameworks from the other sites if it fits, can import the framework and change the files.

<div class='container'> this changes the container to centre

<div class='col-sm-4' adds columns

Using command line >command automation (good at keeping track of changes) red=removed green=added

Workflow = what stages are in github

1. coding\_course in local folder
2. “git repository” (folder), git init (initialise my respository as a git one)
3. “profile creation”: changes to update details after initialising so you can easily update changes
4. “git add”
5. “git commit” (m flag for message which will inform someone else of the changes)
6. “git push”

Homework make website using bootstrap in ‘my first website’ transfer bootstrap 3.3.5 folder to other folder.