## Mobile first approach vs desktop first approach:

A mobile first approach is meant to be fit for a mobile device by default. For example, the navigation bar could be more compact to account for less space, or div elements could be displayed vertically instead of horizontally. A website first approach would be on the opposite end of the spectrum, where the developer has much more freedom.

Mobile first is usually the wise choice, given that a larger percentage of the world view websites from a mobile device compared to a desktop. It also tends to be easier to make a website with mobile-first in mind. With the restricted space of a mobile device, there isn’t a lot of room for flashy stuff like a large header stretching across the screen. This encourages the developer to be more content-focused, and they can add the extra stylized elements later when developing for PC.

**Advantages of mobile first:**

* A higher percentage of people view websites using mobile phones. Prioritizing mobile allows this version of the website to be better.
* More content focused since it doesn’t have space for more stylized elements.

**Disadvantages of mobile first:**

* Since the website is being developed on a PC, it may be a bit more difficult to develop it compared to using your entire screen.
* Space restrictions will limit the creativity you could have implemented had you been doing the PC version first, since the desktop version will still have to follow the same design of the mobile version.

**Advantages of desktop first:**

* No limits, the user can do whatever they want and come up with design choices as they go. For example, a larger header spread across the screen. This can be done without consideration that it wouldn’t work as well on mobile.
* If your website is primarily going to be used on desktop, say for a business, it would be better to prioritize the desktop version.

**Disadvantages of desktop first:**

* For most website purposes, your users will more than likely be using a mobile device.
* Takes more time and effort to convert a desktop website to mobile, since you need to remove some elements that wouldn’t work on a small screen. This can get quite messy, especially deciding whether to remove an element or adjust it.

## CSS Frameworks:

**Advantages of using a framework:**

* Most people will find it easier to use.
* Can be a lot quicker to develop with, since you can easily add classes as you go.
* Everything kept in one place, right next to the element that is being edited.

**Disadvantages of using a framework:**

* Will sometimes clash with regular CSS causing problems.
* Has limitations, cannot do everything that CSS can.
* Version will work differently. You may want to import a new version for the latest features, but they could have changed how something from a past version works breaking that feature on your website.

## Reduce Page Loading Times:

* Move the JS tag to the bottom of the page, so it will load asynchronously.
* Optimize image sizes so they won’t take too long to load.
* Choose a web hosting provider with good performance.
* Have 1 CSS file for all CSS, and 1 JS file for all JS. This is opposed to 1 file for every website page.
* Include minimal http requests since the requests take up resources.

## Differences between cookies and local storage:

**Cookies:**

* Data accessible at client and server side. Cookie request rends data to server side.
* Storage capacity is 4095 bytes.
* Cookies have expiration where the data gets deleted.

**Local Storage:**

* Data accessible at the local browser side.
* Storage capacity is 5mb per domain.
* No expiration, must be removed manually.

## Guidelines and Policies:

**For staff in a web development company:**

* Coding standards
* Client confidentiality
* Website content
* Ethical behaviour for copyright
* Email use
* E-security

**To be included in a website:**

* Privacy policy
* Terms and conditions

## SEO and Website Maintenance:

**5 Ways to improve visibility of a website online:**

* Improve loading times
* Mobile friendly and responsive
* Keywords
* Page names
* Description tags
* Accessibility options
* URL length

**5 ways to maintain and update a website:**

* Keep information relevant
* Monitor speed of website
* Continuously contribute more content to the website

## Discussion about web hosting and marketing:

**Important web hosting factors:**

* Price – Make sure you aren’t getting overcharged for the service that they provide.
* Security – Ensure that your website will be safe from being targeted by hackers, who can steal data and repel customers by taking down/editing the website.
* Performance – Customers will get bored if they have to wait too long for a website to load, so ensure that the web host will not cause any delays.
* Customer support – The company should have quick and effective customer support whenever you have a problem, so your website can be up and running as soon as possible.
* Domain name – Use a domain name that will be easy to find in a search engine, and looks trustworthy enough for visitors to click it.

**Important marketing factors:**

* Search engine optimization so that the website will be more likely to show up to potential visitors.
* Engagement, which will encourage visitors to stick around and draw in new visitors.
* Good design. A website that is difficult to navigate will just frustrate users and cause them to click off.
* Good communication which will allow users to be aware of the website. Advertise it on social media, or in a physical store if your business owns one.

## Threats and preventative measures:

**Threats:**

* Brute force attacks to get into a user’s account and access their personal data, which usually works for weaker passwords.

**Preventative Measures:**

* Add a two-factor authentication method, so even with the account password a hacker wouldn’t be able to access the account.

## Purposes and issues with planning, design, and development stages:

**Planning:**

* Outline the purpose of the website, and who will be using it.
* Could be considered quite boring, developers may be eager to get into making the website itself rather than writing, which can cause them to omit some important planning details.

**Design:**

* Decide between a mobile first or a desktop first approach.
* Plan the page layout. What goes where, colour scheme and style to draw users in.
* Make website look appealing with appropriate white space.
* May be difficult to know the feasibility of adding a feature in the planning stage. Your entire website may be oriented around a specific navigation system, but when it comes to developing you discover it may be too difficult to implement, or has issues that you hadn’t considered at the time.
* Make your website accessible so more people can use it. For example, alt tags that can be read out by text-to-speech software so the visually impaired can consume visual content as well.

**Content Development:**

* What goes on the website in regards to text, images, videos.
* Gather/create the images and videos that you believe will be most appropriate for the website’s purpose.
* When adding text, attempt to update and improve it as you go or go through multiple iterations until you create the best possible version.
* If using materials not created by yourself, you could run into problems with copyright.

## Difference between different methods of adding CSS and JavaScript code:

**Inline:**

* The code exists in the body of an HTML page, usually in the element of the page that is being affected. An example is a style tag in a header, which can be used to change the colour, size, etc, of the text. For JavaScript this could be an onclick event that is calling a function through a button.

**Embedded:**

* Is separated from the other elements but still exists in the same document, this time in the <head> tags. For example, all the CSS code can be contained within two <style> tags, which works similarly to it having its own separate document. JavaScript is similar, except the code is placed between two <script> tags.

**Linked:**

* Both the CSS and JavaScript exist on their own separate documents, but are connected to the HTML page though a link. For example:
* <link rel="stylesheet" type="text/css" href="workshop.css"> for CSS or
* <script type="text/javascript" src="workshop.js"></script> for JS.
* Separating the documents can be cleaner, as all the CSS will be in one place, and all JS will be one place.