Mid Term 1 Notes:

HTML:

1. HTML is a standard way of defining documents for web browsers.
2. First Defined by Tim Berners-Lee in 1990, has been extended and redefined multiple times.
3. The current standard is called HTML5, which was standardized in 2008.
4. 1. In this course, we only use HTML5, and all web development these days should be HTML 5.
5. All browsers (Chrome, Firefox, Safari, Edge, Brave, etc.) know how to read and understand HTML and display it on a screen.
6. HTML can be written in any text editor
7. It is based around the <, and > and & charcaters

LINKS:

1. There are 2 types of links. 1. <a href=””> </a> tag and <link> tag
2. NO SPACES AROUND = SIGN!
3. Attributes can be in any order

URL:

1)When giving link in an <a> element, use URL (typically). URL is an invention of Time Berners – Lee.

2)It was created to solely help the WWW function

3) Tim Berners-Lee created a standardized textual way of describing how to access something on the internet.

4) Although it was created for the WWW, it was created to allow for resources outside of the WWW as well.

5) Today it is a staple of almost every internet protocol and every type of document which could be put online.

URL Structure:

1. Every URL structure begins with a scheme followed by a :
2. 2 of the WWW schemes are: http and https
3. http – insecure \*obsolesced, but still around\* HTTP (Hypertext Transfer Protocol)
4. https – secure HTTP (“secure” meaning encryption, authentication, and integrity verification)
5. Both http and https require a double-slash // after the :

EXAMPLE:

<https://example.com/something?extra&stuff#more>

1. After the // comes the host name
2. This is the name of the web server that is hosting the website
3. It is case-insensitive
4. More sophisticated usage may involve a port number and/ or user authentication information

More about URL Structure:

1. Optionally, a URL can then contain a path.
2. Path begins with a / and indicates which document is being requested from the server
3. If the path is empty or ends with a / , the filename index.html is usually assumed.
4. For this reason, the main page, or landing page of a webpage with multiple pages should be called index.html
5. Path of the URL structure is CASE-SENSITIVE!

Optional Fragment of URL Structure:

1. Optional fragment may appear at the end, beginning with a #
2. In this example, #more is the fragment
3. Fragments are not sent to the web server and are intended for use by the browser to indicate that the browser should automatically skip down to the indicated section
4. We may see fragments briefly later in the course

Absolute and Relative Links

1. Providing a URL in a link is an example of an absolute link
2. An absolute link works in any context: You can copy-paste a URL elsewhere and it will still refer to the same location. That is why it is called absolute link.
3. Relative Link does not contain a URL. It only contains a path name which does not have to begin with a /.
4. Relative link refers to a document relative to the current document being viewed.
5. Relative links should always be used when referring to other documents in the same website/ project
6. This makes it very easy to move your website from one location to another.
7. So, for Assignments / projects, I should always use relative links.

ABOUT CSS (CASCADE STYLE SHEETS) -

CSS History:

1. Early days of WWW, styling information was given directly in the HTML via attributes
2. This had a few drawbacks.
3. Not every browser supported all the same attributes in the same way
4. Changing some styling (especially across large websites) was very tedious and error-prone
5. There was no possibility of multiple ways of styling
6. People had been using stylesheets before the WWW had even existed for other languages other than HTML
7. This was finally standardized in 1996 with the release of CSS1
8. However, nobody was using CSS1 because browser support was very poor and most browsers were interested in controlling the WWW and not interested in standards

Early CSS use:

1. In the early 2000s, CSS got better browser support and started picking up momentum.
2. There were still inconsistencies between browsers
3. CSS did not fully obsolescence the old HTML – styling until the mid 2000s.

Modern history:

1. These days CSS is expected in any form of web development
2. It nicely separates the content and structure (HTML) from the presentation and styling (CSS)
3. We are transitioning from CSS 3 (1999) to CSS4 (released incrementally, not at once)
4. CSS4 is backwards-compatible with CSS3

CSS –

1. CSS is a stylesheet language (Not programming language)
2. It looks quite different to HTML and its syntax needs to be learned separately

CSS example:

Td.xyz I {

Font-size : 80%;

Margin-right:1em;

}

1. It consists of a selector (td.xyz i)
2. Followed by a list of declarations; each declaration consists of a property (font-size) and a value (80%)
3. Curly-braces, colons and semi-colons are mandatory

Selectors –

1. Tag name : e.g., li will affect all li elements
2. Class name: e.g., .big will affect all elements with the big class name
3. ID name: e.g., # special will affect only one element, tagged in the HTML with the special ID name
4. Descendant : e.g., ol li will affect only li elements which exist inside of an ol

Classes and ID-

1. When using classes and IDs, some coordination between HTML and CSS is required
2. The HTML should tag which class a particular elements should belong to
3. The CSS can then define how elements of that class are displayed
4. The class name used in the HTML and the CSS must match exactly (case-sensitive)

Classes VS IDs

1. Classes and IDs accomplish the same task, of marking an element for styling
2. The same class can be applied to multiple elements
3. And ID must be unique (only one element can have a given ID name)
4. When in doubt, use classes
5. Use an ID only if you know an element can only ever appear once

Multiple classes and IDs

1. An element can have both an ID and classes
2. An element can have multiple classes ( separated by spaces)
3. Example: <p class=”important big cool” id=”awesome”>Hello!</p>

Span –

1. It often happens that you want to tag only a word or part of an element with a class
2. The way to do this is to put that word in a span element
3. A span is defined as having no special styling, and exists only to give classes or IDs to words
4. Example:

CSS

. important {

Font-style: bold;

Font-size: 1.2em;

}

HTML

<p>

Please remember to <span class=” important”> not feed the bears</span>.

</p>

Colour properties

1. One common use of CSS is to set the colour of something
2. Many HTML elements have both background-color and color (text colour)

Colour Values

1. There are multiple ways of specifying colours in CSS
2. By name: ex. Blue, white, See a full list of colours
3. Red green blue – ex. Rgb (127, 191, 2) rgb (50%,75%,1%) , 7fbf01

RGB –

Color displayed in RGB forms examples:

Black – rgb(0,0,0) , rgb (0%, 0%, 0%)

But solid colors can be better be used in their names.

Like background-color: black;

Size Properties –

1. Sizes and lengths make up a big part of visual design and hence a big part of CSS
2. We will revisit sizes and lengths multiple times throughout the course
3. Our first introduction will be with font-size, changing the size of text in an element

Warning about Pixels –

1. If you search online for CSS solutions to certain problems, you will often see solutions using px
2. We will avoid using px in this course
3. There is almost never a good reason to measure a distance in pixels, and in this course we will avoid doing so
4. Screens today come in a wide variety of pixels densities

Relative –

These units work relative to an element’s parent element and are used most often. Using them allows fonts across the entire document to change size uniformly when the page is resized.

Ex. 120% means the font size should be 120% as big as it was.

Ex2. 1em is equivalent to 100%. 1.2em is 120%.

Rem \*root-element base font-size

Vw, Vh - 1% of the width / height of the screen. 1.5vmin is 1.5% of the width if held portrait, and 1.5% of the height if rotated into landscape.

Vmin, vmax – 1% of the width/height of the screen, whichever is smaller.

Pixels –

1. Most image file formats are raster or bitmap file formats ; including JPEG, PNG, GIF and WebP
2. Raster images are images which are represented as a 2D grid of pixels
3. Screens also work via pixels, so they are easy to display on screen
4. Digital cameras and scanners have sensors that work as an array of pixels, so any photographic image should be a raster image.

Resolution

1. Every raster image has a resolution, which is stated as the number of pixels in each row, following by the number of pixels in each column, often with an X between them (ex. 1440 x 768)
2. The higher the resolution, the higher the image quality.
3. Low-resolution imgaes will look pixelated (blocky, like the cat above)

Colours –

1. Component : each pixel is stored as a number of different components. RGB (Red, Green, Blue) components are one very popular way of storing component colour
2. Indexed: The image file contains a palette describing a fixed number of colours, each colour having an index number associated with it. Each pixel is represented as an index into that palette
3. If a picture is reduced to a palette (or colourmap) of 16 colours, quality is shit
4. Every pixel is represented as a number of 0 to 15 in this case
5. The palette is stored in the file separately from the pixels

Usage of indexed –

1. Indexed colour management is more restrictive
2. It is generally not suitable for photographs
3. Indexed colour management is more restrictive
4. It is generally not suitable for photographs
5. It can be great for diagrams, logos, line art, etc., where not a lot of distinct colours are needed

Components –

1. For images that use a component-based colour model, the colour depth is important
2. Colour depth describes how many different colours can be presented
3. 24-bit colour depth (8bits, 0-255 for each of red, green, blue) is popular and is referred to as true colour
4. Professionals sometimes use 36-bit or 48 bit colour depth
5. 18 bit and 16 bit (5bbit red, 6 big green, 5-bit blue) are also used

Transparency and translucency

1. Some image formats (notably PNG and WebP) allow opacity, also known as an alpha channel
2. It is another color component which indicates whether each particular pixel should be transparent or partially transparent (translucent)
3. It is very useful when a diagram or part of an image should allow the background to show through

Animations

1. Some formats support animations or videos (with or without sound)
2. For many years, the only animation image format with good browser support was GIF
3. This is no longer true
   1. All browsers made in the past 10-ish years support WebM and Theora (and usually others)
   2. GIF no longer has any purpose
   3. It is strictly inferior to other technologies, so please do not use it

Compression –

1. Image data can be very large
2. E.g., a 2 megapixel image at 24-bit colour depth would theoretically be 6MiB, which is very very large for a webpage
3. To improve webpage loading time and save bandwidth, we want images to be very small
4. This is a accomplished viua compression

Lossy vs Lossless –

1. Lossless : Compression finds repetitions or patterns in the image data to do compression, and does not change the original ijmage data at all\* it is exactly perfect
2. Lossy : Compression uses psychovisual models to determine which details humans will notice and which they won’t. It changes the image data, reducing image quality, but reducing it in places where humans will not notice it.

On a webpage ( Lossy vs Lossless)

1. Lossless : should be favoured in instances where there is a lot of text or lines and fine details are important. This includes graphs, charts, diagrams and sometimes even drawings or cartoons.
2. Lossy : Should be favoured in instances where there is a lot of complex imagery, such as photographs or complex paintings or drawings

Lossy compression –

1. When lossy compression is used:
   1. Blockiness and poor definition in the background
   2. Poor colour definition / “bleeding” , particularly around the face

Vector images –

1. A vector image is an alternative to a raster image
2. Vector images do not have pixels or resolutions
3. They should not be used for photographs!
4. Vector images are made up of primitigves, such as lines, polygons, text, etc.
5. They are resolution-independent, meaning they can be scaled (or zoomed in) to any degree and not show any pixel artifacts (unlike raster images)
6. They are ideal for diagrams
7. SVG is a standard vector format supported by all tools and browsers
8. Unfortunately, making SVG images requires quite a log of design skill and we will not actually make any in this course

Major formats availability table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Lossless | Lossy | Alpha | Indexed | Component |
| JPEG |  | Yes |  |  | 24-bit RGB |
| PNG | Y |  | Y | Y | Y |
| WebP | Y | Y | Y | Y | Y |
| GIF | Y |  | Y | 256 |  |

1. Use WebP if your image editor supports it (GIMP) is usually okay)
2. Otherwise, use PNG if you need transparency or lossless date
3. Otherwise, use JPEG
4. DOnot ever use GIF

Box model –

1. Every visual demonstration of the box model in CSS
2. Every visual element in HTML actually has 4 boxes associated with it
3. From the outside in:
   1. Margin : is the blue-green box; cannot contain any content or controlled beyond its size
   2. Border : is the black box can be styled and coloured, but cannot contain content
   3. Padding : is the purple box; it is considered to be part of the element
   4. Content : Is the black text

Margin:

1. The margin is optional space that exists between this element and other elements.
2. We can control it in CSS only via margin, magin-top, margin-bottom, margin-left and margin-right
3. Set the size of a margin either according to:
   1. relative to text size (ex. Em), so that the margin will grow as text size grows
   2. Relative to screen size (ex. vh, vw) so that it is independent of text size
4. Beyond the size/spacing, the visuals of the margin cannot be controlled
5. The margin always shows the (background) colour of the parent element

Border:

1. The border is an optional rectangle that “borders” or frames the content
2. To be visible, the border must have a width, a colour, and a style
3. Border styles are most commonly solid but may also be:
   1. 2D effect – dotted, dashed, double, solid
   2. 3D effect – groove, ridge, inset, outset
4. Border width :
   1. Border widths are the one place where I will accept px as a unit of width
   2. 1px and 2px are common ways of specifying thin borders

Padding:

1. Padding is very similar to margin, but with the following differences:
   1. It is inside the border (if the border exists)
   2. It is coloured with the background colour/ image of this element, not the parent element
   3. This will become important when we learn about widths and heights of elements

Side-by-Side content (Float)

1. In CSS, there are 3 possible values for float
   1. Clear (default) : This element is not floated. All subsequent elements should go underneath it.
   2. Left: This element is floated left. All subsequent elements should be beside it to the right and wrap around it.
   3. Right : Just like left but right.
2. Floats are used often with imgs and divs
3. It allows an easy way to block of certain parts of content and have them display in multiple columns