**HTML & CSS Assignment**

**Submitted By:**

**Pranjal Suyal**

**Theoretical:**

**HTML:**

**Q1.Explain structure in html ?**

**Ans.** HTML stands for HyperText Markup Language and is the basic structural element that is used to create web pages. HTML is a markup language, which means that it is used to “mark up” the content within a document, in this case a webpage, with structural and semantic information that tells a browser how to display a page. When an HTML document is loaded by a web browser, the browser uses the HTML tags that have marked up the document to render the page’s content.

There are three types of code that make up a basic website page. HTML governs the structural elements, CSS styles those elements, and JavaScript enables dynamic interaction between those elements.

HTML structure + CSS style + JS interaction = web page

A basic HTML page is a document that typically has the file extension .html, though HTML frequently appears in the content of other file types as well. All HTML documents follow the same basic structure so that the browser that renders the file knows what to do. The basic structure on which all web pages are built looks like this:

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>Homepage Headline</h1>

<p>This is a paragraph.</p>

</body>

</html>

Doctype

The first line of code, <!DOCTYPE html>, is called a doctype declaration and tells the browser which version of HTML the page is written in.

HTML Root Element

Next, the <html> element wraps around all of the other code and content in our document. This element, known as the HTML root element, always contains one <head> element and one <body> element.

Head Element

The HTML head element is a container that can include a number of HTML elements that are not visible parts of the page rendered by the browser.

The <title> element is the only element that is required to be contained within the <head> tags.

Body Element

There can only be one <body> element in an HTML document because this element is the container that holds the content of the document. All of the content that you see rendered in the browser is contained within this element.

**Q2.What is block or inline elements. Explain with some basic tag’s examples.**

**Ans.**HTML is made up of various elements that act as the building blocks of web pages. For the purpose of styling, elements are divided into two categories: block-level elements and inline elements.

In summary, a <span> element is used as an inline element and a <div> element as a block level element.

Basically, an inline element does not cause a line break (start on a new line) and does not take up the full width of a page, only the space bounded by its opening and closing tag. It is usually used within other HTML elements.

Other examples of inline elements are:

anchor <a> tag

emphasis <em> tag

image <img> tag

A block-level element always starts on a new line and takes up the full width of a page, from left to right. A block-level element can take up one line or multiple lines and has a line break before and after the element.

Other examples of the block-level tag are:

Heading tags <h1> to <h6>

List (Ordered, Unordered, Description and List Item) tags <ol> , <ul> ,<dl> , <li>

Preformatted text tag <pre>

Blockquote tag <blockquote>

**Q3.What are attributes in html element? List some attributes of INPUT tag.**

**Ans.**An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts − a name and a value

The name is the property you want to set. For example, the paragraph <p> element in the example carries an attribute whose name is align, which you can use to indicate the alignment of the paragraph on the page.

The value is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: left, center and right.

Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

**Attributes of INPUT tag:**

* Align **-** Defines the alignment of content
* Alt -This specifies the text to be used in case the browser/user agent can't render the input control.
* Autocomplete - Specifies for enabling or disabling of autocomplete in <input> element
* autofocus - specifies that <input> element should automatically get focus when the page loads
* Checked - If type = "radio" or type = "checkbox" it will already be selected when the page loads.
* Disabled - Disables the input control. The button won't accept changes from the user. It also cannot receive focus and will be skipped when tabbing.
* Form - Specifies one or more forms

**Q4.What are HTML Layouts ? How are they important?**

**Ans.**HTML layouts provide a way to arrange web pages in well-mannered, well-structured, and in responsive form or we can say that HTML layout specifies a way in which the web pages can be arranged. Web-page layout works with arrangement of visual elements of an HTML document.

Web page layout is the most important part to keep in mind while creating a website so that our website can appear professional with the great look. You can also use CSS and JAVASCRIPT based frameworks for creating layouts for responsive and dynamic website designing.

Following are different HTML5 elements which are used to define the different parts of a webpage.

<header>: It is used to define a header for a document or a section.

<nav>: It is used to define a container for navigation links

<section>: It is used to define a section in a document

<article>: It is used to define an independent self-contained article

<aside>: It is used to define content aside from the content (like a sidebar)

<footer>: It is used to define a footer for a document or a section

<details>: It is used to define additional details

<summary>: It is used to define a heading for the <details> element

**Q5. What are the different type of image extension and which one is better among them with reason ?**

**Ans.** Types of Image Files

JPEG (or JPG) - Joint Photographic Experts Group.

PNG - Portable Network Graphics.

GIF - Graphics Interchange Format.

TIFF - Tagged Image File.

PSD - Photoshop Document.

PDF - Portable Document Format.

EPS - Encapsulated Postscript.

AI - Adobe Illustrator Document.

Better among them are:

**PNG**. PNG is another very common and versatile image format found on the web today.

Strength: Versatile, supports transparency

**GIF**. GIF is a growing force on the web now

Strength: Small file size, allows for animation

**TIF**. TIF is known as the highest quality image format, is best used for commercial print work and shouldn't be used on the web.

Strength: Highest quality

**Q6.What is SEO ? What are the techniques to implement it in a webpage with example ?**

**Ans.** SEO is an acronym that stands for search engine optimization, which is the process of optimizing your website to get organic, or unpaid, traffic from the search engine results page.

In other words, SEO involves making certain changes to your website design and content that make your site more attractive to a search engine. You do this in hopes that the search engine will display your website as a top result on the search engine results page.

Though search engine optimization can get quite complex when it comes to all the different factors that impact your ranking, the basic process is not as difficult to understand.

**Techniques are:**

**Title Tag** – The title tag on each page tells the search engines what your page is about. This should be 70 characters or less, including both the keyword your content focuses on and your business name.

**Meta Description** – The meta description on your website tells search engines a little bit more about what each page is about. This is also used by your human visitors to better understand what the page is about and if it’s relevant. This should include your keyword and also provide enough details to tell the reader what the content is about.

**Sub-headings** – Not only do subheadings make your content easier for visitors to read, but it can also help improve your SEO. You can use H1, H2, and H3 tags to help search engines better understand what your content is about.

**Internal Links** – Building internal links, or hyperlinks to other content on your site, can help search engines learn more about your site. For example, if you are writing a post about the value of a specific product or service, you can link to the product or service page in your blog post.

**Image Name and ALT Tags** – If you are using images on your website or within your blog content, you will also want to include your keyword or phrase in the image name and alt tag. This will help search engines better index your images, which may appear when users perform an image search for a certain keyword or phrase.

**CSS:**

**Q1.What are the different ways to implement css in a webpage?**

**Ans.** There are Four methods of adding CSS to HTML.

1. **Inline:** by inserting a style attribute inside an <head> element.

An inline CSS is used to apply a unique style to a single HTML element.

Ex:

<h1 style="color:blue;">This is a Blue Heading</h1>

1. **Embedded/Internal:** by putting all css inside head using <style> tags

An internal CSS is used to define a style for a single HTML page.

Ex:

<head>

<style>

body {background-color: powderblue;}

h1 {color: blue;}

</style>

</head>

1. **Linked/External:** CSS is placed in an external .css file, and linked to the HTML document using a <link> tag. An external style sheet is used to define the style for many HTML pages. With an external style sheet, you can change the look of an entire web site, by changing one file!

Ex:

<head>

<link rel="stylesheet" href="styles.css">

</head>

1. **Imported:** Importing a CSS file using @import

Ex:

<head>

<title>My Example</title>

<style>

@import "/css/file/example.css";

</style>

</head>

We are not limited to just one method of applying styles to a document. We can combine all three methods if required.

**Q2.Explain different type of units in css with their differences.**

**Ans:** CSS supports a number of measurements including absolute units such as inches, centimeters, points, and so on, as well as relative measures such as percentages and em units. You need these values while specifying various measurements in your Style rules e.g. border = "1px solid red".

**Different units are :**

* **% -** Defines a measurement as a percentage relative to another value, typically an enclosing element.
* **cm -** Defines a measurement in centimeters.
* **em -** A relative measurement for the height of a font in em spaces. Because an em unit is equivalent to the size of a given font, if you assign a font to 12pt, each "em" unit would be 12pt; thus, 2em would be 24pt.
* **ex -** This value defines a measurement relative to a font's x-height. The x-height is determined by the height of the font's lowercase letter x.
* **in -** Defines a measurement in inches.
* **mm -** Defines a measurement in millimeters.
* **pc -** Defines a measurement in picas. A pica is equivalent to 12 points; thus, there are 6 picas per inch.
* **pt -** Defines a measurement in points. A point is defined as 1/72nd of an inch.
* **px -** Defines a measurement in screen pixels.

**Q3. What is box model in css? Explain with example.**

**Ans:**CSS box model is a container which contains multiple properties including borders, margin, padding and the content itself. It is used to create the design and layout of web pages. It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model.

Box-Model has multiple properties in CSS. Some of them are given below:

* **Borders :** It is the area between the box’s padding and margin. Its dimensions are given by the width and height of border.
* **Margins :** This area consists of space between border and margin. The dimensions of Margin area are the margin-box width and the margin-box height. It is useful to separate the element from its neighbors.
* **Padding :** It includes the element’s padding. This area is actually the space around the content area and within the border box. Its dimensions are given by the width of the padding-box and the height of the padding-box.
* **Content :** This area consists of content like text, image, or other media content. It is bounded by the content edge and its dimensions are given by content box width and height.

Ex:

div {

border: 50px solid green;

margin: 50px;

padding:40px;

width: 400px;

height: 100px;

content: "•"; /\* Insert content that looks like bullets \*/

}

The following property will apply box model properties specified to the div elements.

**Q4.Explain different type of positioning in css.**

**Ans:** The position property specifies the type of positioning method used for an element.

There are five different position values:

* **static** **-** HTML elements are positioned static by default. Static positioned elements are not affected by the top, bottom, left, and right properties. An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

Ex:

position: static;

* **relative -** An element with relative position is positioned relative to its normal position. Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

Ex:

position: relative;

* **fixed -** An element with fixed position is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element. A fixed element does not leave a gap in the page where it would normally have been located.

Ex:

position: fixed;

* **absolute -** An element with absolute position is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed). However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling. Note: A "positioned" element is one whose position is anything but static.

Ex:

position: absolute;

* **sticky -** An element with sticky position is positioned based on the user's scroll position. A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

Ex:

position: sticky;

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

**Q5. Explain different types of display in css.**

**Ans:** The Display property in CSS defines how the components(div, hyperlink, heading, etc) are going to be placed on the web page. As the name suggests, this property is used to define the display of the different parts of a web page.

Syntax:

display: value;

**Different types of display are as follows:**

* **inline -** It is used to display an element as an inline element.
* **block -** It is used to display an element as a block element
* **contents -** It is used to disappear the container.
* **flex -** It is used to display an element as a block-level flex container.
* **grid -** It is used to display an element as a block-level grid container.
* **inline-block -** It is used to display an element as a inline-level block container.
* **inline-flex -** It is used to display an element as a inline-level flex container.
* **inline-grid -** It is used to display an element as a inline-level grid container.
* **inline-table -** It is used to display an inline-level table
* **list-item -** It is used to display all the element in <li> element.
* **run-in -** It is used to display an element inline or block level, depending on context.
* **table -** It is used to set the behavior as <table> for all element.
* **none -** It is used to remove the element.
* **initial -**  It is used to set the default value.
* **inherit -** It is used to inherit the property from its parents elements.

**Q6.What are css selectors and css selector specificity? Explain with example.**

**Ans:** CSS selectors are used to select the content you want to style. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute, etc.

There are several different types of selectors in CSS.

1. **CSS Element Selector -** The element selector selects the HTML element by name.

Ex:

p{

text-align: center;

color: blue;

}

1. **CSS Id Selector -** The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element. It is written with the hash character (#), followed by the id of the element.

Ex:

#para1 { /\*for element with id : para1\*/

text-align: center;

color: blue;

}

1. **CSS Class Selector -** The class selector selects HTML elements with a specific class attribute. It is used with a period character . (full stop symbol) followed by the class name.

Ex:

.center { /\*for elements with class name center\*/

text-align: center;

color: blue;

}

1. **CSS Universal Selector -** The universal selector is used as a wildcard character. It selects all the elements on the pages.

Ex:

\* {

color: green;

font-size: 20px;

}

1. **CSS Group Selector -** The grouping selector is used to select all the elements with the same style definitions. Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

Ex:

h1,h2,p {

text-align: center;

color: blue;

}

**CSS Selector Specificity :**

Specificity is a weight that is applied to a given CSS declaration, determined by the number of each selector type in the matching selector. When multiple declarations have equal specificity, the last declaration found in the CSS is applied to the element. Specificity only applies when the same element is targeted by multiple declarations. As per CSS rules, directly targeted elements will always take precedence over rules which an element inherits from its ancestor.

The following list of selector types increases by specificity:

* **Type selectors** (e.g., h1) and **pseudo-elements** (e.g., ::before).
* **Class selectors** (e.g., .example), **attributes selectors** (e.g., [type="radio"]) and pseudo-classes (e.g., :hover).
* **ID selectors** (e.g., #example).
* **Universal selector** (\*), **combinators** (+, >, ~, ' ', ||) and **negation pseudo-class** (:not()) have no effect on specificity. (The selectors declared inside :not() do, however.)
* I**nline styles** added to an element (e.g., style="font-weight: bold;") always overwrite any styles in external stylesheets, and thus can be thought of as having the highest specificity.

**Note :** When an important rule is used on a style declaration, this declaration overrides any other declarations. Although technically !important has nothing to do with specificity, it interacts directly with it.

**Q7.What is pseudo class, pseudo elements and pseudo selector? List them all with examples.**

**Ans:** **pseudo selector :** Pseudo class selectors are CSS selectors with a colon preceding them. We are probably very familiar with a few of them. Like hover:

a:hover {

/\* Yep, hover is a pseudo class \*/

}

They are immensely useful in a variety of situations. Some of them are CSS3, some CSS2... it depends on each particular one.

Some of them are:

**:link -** Perhaps the most confusion-causing link-related pseudo selector. Aren't all <a> links? Well not if they don't have an href attribute. This selects only those that do, thus is essentially the same as a[href]. This selector will become a lot more useful should any-element linking become reality.

**:visited -** Selects links that have already been visited by the current browser.

**:hover -** When the mouse cursor rolls over a link, that link is in it's hover state and this will select it.

**:active -** Selects the link while it is being activated (being clicked on or otherwise activated). For example, for the "pressed" state of a button-style link

**pseudo class :** A CSS pseudo-class is a keyword added to a selector that specifies a special state of the selected element(s). For example, :hover can be used to change a button's color when the user's pointer hovers over it.

/\* Any button over which the user's pointer is hovering \*/

button:hover {

color: blue;

}

Pseudo-classes let you apply a style to an element not only in relation to the content of the document tree, but also in relation to external factors like the history of the navigator (:visited, for example), the status of its content (like :checked on certain form elements), or the position of the mouse (like :hover, which lets you know if the mouse is over an element or not).

**Note :** In contrast to pseudo-elements, pseudo-classes can be used to style an element based on its state.

**pseudo elements :** A CSS pseudo-element is a keyword added to a selector that lets you style a specific part of the selected element(s). For example, ::first-line can be used to change the font of the first line of a paragraph.

/\* The first line of every <p> element. \*/

p::first-line {

color: blue;

text-transform: uppercase;

}

**Note:** In contrast to pseudo-classes, pseudo-elements can be used to style a specific part of an element.

**Q8. What are media queries in css ? Why it is used?**

**Ans :** Media queries are useful when you want to modify your site or app depending on the device's general type (such as print vs. screen) or specific characteristics and parameters (such as screen resolution or browser viewport width). The @media rule, introduced in CSS2, made it possible to define different style rules for different media types. Media queries can be used to check many things, such as:

* width and height of the viewport
* width and height of the device
* orientation (is the tablet/phone in landscape or portrait mode?)
* resolution

Using media queries are a popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones (such as the iPhone and Android phones).

A media query consists of a media type and can contain one or more expressions, which resolve to either true or false.

@media not|only mediatype and (expressions) {

CSS-Code;

}

The result of the query is true if the specified media type matches the type of device the document is being displayed on and all expressions in the media query are true. When a media query is true, the corresponding stylesheet or style rules are applied, following the normal cascading rules. Unless you use the not or only operators, the media type is optional and the all type will be implied. You can also have different stylesheets for different media, ex:

<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="print.css">

**Q9. What if two same ids have a same css style?**

**Ans :** We can have two same ids with same css style but JavaScript and CSS selectors will only select the first element they find with the id. It's not like our website will crash. It's just very bad practice because an id should be unique one each page.

IDs should be unique, so you should only use a particular ID once on a page. Classes may be used repeatedly. In some situations it won’t matter if you use an ID more than once. In those situations nothing will break and no one will notice. But there are times when using an ID more than once can cause a problem. This is particularly true for forms and scripts. To avoid such problems it is best to get into the habit of keeping IDs unique.

**In Short :**

Should IDs be unique? YES.

Must IDs be unique? NO, at least IE and FireFox allow multiple elements to have the same ID.