**PINESPHERE INTERN**

**DAY -1** **TIME: 1:15PM**

**HTML: HyperText markup language – invented by Tim Berners Lee in 1993 who also invented www, url system and http**

**As of jan 10th 2023 – html5 is the version**

**Html – struct of webpage**

**Css – color and style to webpage**

**Javascript – to perform actions like clicking, moving the cursor etc. on the webpage**

**HYPERTEXT – text which will take us to other pages (Links are also called hyperlinks or hypertexts)**

**MARKUP – set of symbols in a text doc to control it’s struct, format**

**HTML – TAGS: - not case sensitive**

**Open tag - <>**

**Close tag - </>**

**Head tag - <head> - contains the info about the page**

**Body tag - <body> - container for visible contents like headings, paragraphs, images, hyperlinks etc.,**

**Heading tags – h1, h2, h3, h4, h5, h6 – decreasing order of headings in size**

**Paragraph tag - <p> - to define a paragraph – eventhough u leave some spaces btw the contents in a text inside p tag, it automatically removes and provides a clear paragraph**

**Preformatted text - <pre> - to display the text as of u specify it with spaces and fonts in same way. To overcome the problem in p tag**

**Break tag - <br> - it breaks the current line and continues from the next according to the number of <br> tag used. It doesn’t need a close tag**

**Horizontal rule - <hr> - to define a thematic break or like draw a horizontal line and then continue**

**HTML ATTRIBUTE:**

**Provide additional info about the elements**

**Always specified inside the open tag or start tag**

**These are name value pairs – like – name – “value”**

**Style attribute – to add styles to an element such as color, font, size etc.,**

**Can be used in any text related tags**

**Syntax = <tagname style = “property:value;”>**

**Property – css property , value – css value**

**background-color , powder blue, tomato etc.,**

**color – to define text color, blue,red etc.,**

**font-family , verdana,courier, etc.,**

**font-size , 300% , 160% etc.,**

**text-align , center,left,right**

**border – to define the color of the border , 2px solid tomato**

**COLORS IN HTML**

**RGB – RED, GREEN,BLUE (0 to 255)each color**

**RGBA – RGB WITH alpha value – alpha value to define opacity or transparency (0 to 1)**

**HEX colors – defining rgb in hexadecimal format - #RRGGBB**

**HTML TEXT FORMATTING:**

**Lang attribute – used inside html tag to define the language of the webpage**

**<html lang = “en”></html>**

**Title attribute – to define some extra information about an element**

**Displays some information if moved over the element for which the title attribute is defined**

**Called as tooltip to the element**

**Anchor tag - <a> - to move from one page to another**

**Href – attribute to define which page to move when content of <a> tag is clicked**

**The target attribute can have one of the following values:**

**\_self - Default. Opens the document in the same window/tab as it was clicked**

**\_blank - Opens the document in a new window or tab**

**\_parent - Opens the document in the parent frame**

**\_top - Opens the document in the full body of the window**

**Image tag – <img> - to add image to the page**

**Src – attribute that contains the name of the img file with extensions**

**Width – attribute to set the width of the image**

**Height – attribute to specify the height of the image**

**Alt – attribute that contains alternative text if image doesn’t load or the specified image doesn’t exist due to spelling mistake in mentioning file name or any other thing**

**Center tag - <center> - within which we can add elements which are to be present at the center of the webpage.**

**Comment tag - <!—comment here-->**

**To place notifications and reminders in html code, to hide one or more lines or to hide only a part of the text .**

**Map tag – <map> to define an image map. Image map is an image with clickable areas defined by <area> tag**

**To use map we need an image with attributes src and usemap mainly**

**Usemap – attribute of img tag mentioning which map function to be used. Usually contains the name of the map with a hastag at the start.**

**usemap = “#mapname”**

**Name – attribute that takes the name of the map as value**

**Area tag –**

**Shape – attribute that define the shape of the area on an image (rect,circle,poly,default – entire region)**

**Coords – these are the co ordinate values of the area**

**Alt – to define a name of the area if not loaded**

**Href – to define to which page have to be reached if that particular area is clicked**

**BACKGROUND IMAGES:**

**Use style attribute and the css background-image property**

**Also by specifying the image in style element in the head section**

**DAY – 2 Time : 8:30am**

**HTML FAVICON (favorite icon)– small image that is displayed to the left of the page title**

<link rel="icon" type="image/x-icon" href="/images/favicon.ico">

**Rel – specifies the relationship btw crnt doc and linked resource**

**Type – defines the type of image or text of the linked resource**

**Href – specify loc url of the icon**

**HTML TABLES: <table>**

**Use the style attribute with the width or height properties to specify the size of a table, row or column.**

**Caption - <caption> - defines the table title or caption**

**Row - <tr>**

**Table header - <th> - bold and centered by default**

**We can have the header vertically as well as horizontally or can be both**

**We can also have the headers specified for more then one col or more than one row. It can be done by using colspan and rowspan attribute on the <th> tag**

**Table data (Cell values) - <td> - it can be text,img,list,link,other tables etc.,**

**<colgroup> tag – specifies the group of one or more col in a table for formatting**

**Useful to apply styles to entire columns instead of repeating the styles for each cell for each row.**

**To define different properties to a column within a <colgroup>, use the <col> tag within the <colgroup> tag**

**Span – attribute that specifies the number of col a colgroup should span**

**<thead>, <tbody>, <tfoot> - groups the head, body and footer contents of the table**

**HTML LISTS <li> - list tag**

**Lists can be nested**

**<li> - can have images,links,new list, other html elements**

**Ordered list – has some ordered bullets alphabetical or numerical <ol>**

**Type – attribute sets the item marker type (1 default numbers, A – uppercase alphabets, a – lowercase alphabets, I – uppercase roman numbers, i – lowercase roman numbers)**

**Start – attribute controls where to start the counting from**

**Unordered list – has bullets as symbols <ul>**

**Style attribute – list-style-type -define the style of the list item marker (disc(default), circle, square, none)**

**Each item in a list should be assigned with <li> tag**

**Description list - <dl> - list of term with description**

**<dt> - defines the term**

**<dd> - describes each term**

Display values

block

Inline

**HTML BLOCK – LEVEL ELEMENTS:**

**Always starts on a new line and the browsers automatically add some space(margin) before and after the element**

**Always stretches out to the left and right as far as it can**

**<div>, <p>**

**<h1>, <h2>, ..., <h6>**

**<ul>, <ol>, <li>**

**<table>, <tr>, <td>**

**<p> and <div> commonly used**

**<div> - defines a division or a section in an html page**

**Block-level elements create a "block" on the web page, meaning they typically start on a new line and take up the full width available.**

**They stack on top of each other, and each block-level element creates a new "block" or "box" in the layout.**

**Typically used to structure the layout of a web page.**

**Often used for grouping and dividing content into sections.**

**<div> tag: - to group sections of a webpage together**

**HTML INLINE ELEMENTS:**

**Inline elements do not start on a new line and only take up as much width as necessary.**

**They do not create new "blocks" in the layout; instead, they flow within the content.**

**Typically used for styling and formatting individual pieces of text or elements within a block-level structure.**

**Often used for adding emphasis, links, or inline styling.**

**HTML CLASSES:**

**Class attribute – to specify the class for an html element**

**Multiple elements can share same class**

**Can be used on any html element**

**Class name is case sensitive**

**It is mainly used to style the elements**

**An element can belong to more than one class too. It can be defined by separating the class name with spaces.**

**HTML ID:**

**Id attribute – to specify unique id for an html element**

**Cannot have more than one element with same id**

**To point to a specific style declaration in a style sheet.**

**Syntax: #idname and define css properties with {} under style element**

**Id name is case sensitive**

**Cannot strt with number and not contain white spaces**

**HTML IFRAMES: <iframe>**

**To display a webpage within a webpage**

**Src – attribute to specify which page to be displayed**

**Height, width – attribute to specify the size of the frame**

**Style – attribute to specify the styles of the frame**

**Name – attribute that has the name of the iframe**

**This name can be used as a value for target attribute inorder to open the desired webpage in the frame. This webpage will be opened once the name and target matches**

**HTML LAYOUT ELEMENTS:**

**Websites often display content in multiple columns**

**HTML responsive web design:**

**To create webpages that look good on all device.**

**Automatically adjust for different screen sizes and viewports.**

**<meta> tag is must to create a responsive web design**

**HTML FORMS:**

**To collect user input and that input is often sent to server for processing**

**<form> - to create an html form**

**Action – attribute that has the link of the page which will be displayed once the submit button is clicked**

**Target – attribute that specifies where to display the response after submitting the form**

**\_blank – new window or tab**

**\_self – current window**

**\_parent – in parent frame**

**\_top – full body of the window**

**Framename – named iframe**

**Method – attribute that specifies the http method to be used when submitting the form**

**Get – append the form data to the url in name/value pairs**

* **Not to be used with sensitive data**

**Post – appends form data inside the body of http request**

* **Specially used at tyms of sensitive data**

**<input> - to get user input in different formats by type attribute**

**Type – attribute that specifies the type of user input to be got. Default is text**

**Type = text – single line text input field**

**Radio – radio button for selection one of the choices**

**Checkbox – to select zero or more of many choices**

**Submit – a submit button**

**Button – a clickable button will be created**

**Email, file, image, month, number, password, search, time, url, week, color, date**

**Name – attribute for identifying the input. It is used to associate a name with the data that the user enters into the form.**

**<label> - it is like providing an idea of what should be entered in the box given**

**For - attribute must be equal to the id attribute of the input element to bind them together**

**<select> - defines a dropdown list**

**Size – attribute to specify the number of visible values**

**Multiple attribute – to allow multiple value selection**

**<option> - defines an option that can be selected**

**Default selection is the first element in dropdown menu**

**We can also specify the default value to be selected by using selected attribute**

**<textarea> - defines multi line input field**

**Rows – attribute specifying the visible number of lines in a text area**

**Cols – visible width of a text area**

**Can also define the size of the text area by using css style property inside textarea tag**

<textarea name="message" style="width:200px; height:600px;">

The cat was playing in the garden.

</textarea>

**<button> - defines a clickable button**

**Onclick – specifies the message or link of what to be displayed once the button is clicked**

**<fieldset> - to group related data in a form**

**<legend> - defines caption for the fieldset element**

**<datalist> - specifies a list of pre-defined options for an <input> element. User will see a drop-down list of pre-defined options as they input**

**List attribute of the input element must refer to the id attribute of the datalist element**

**HTML MULTIMEDIA:**

**<video> - element to add vdo in a webpage**

**Width,height – attribute that sets the height and width of the vdo.**

**Controls – attribute that controls the vdo like pause,play and volume**

**<source> - to specify alternative vdo file**

**<audio> - to play an audio file**

**Youtube vdo adding – use iframe and src**

**DAY – 3 TIME: 8:10 AM**

**CSS: - CASCADING STYLE SHEETS**

**To define style to the html pages**

**CSS SYNTAX:  
 h1{color:blue; font-size:12px;}**

**H1 – selector**

**Color, font- size – property**

**Blue, 12px – value**

**{--} – as a whole with property and value inside is called declaration**

**CSS SELECTORS:**

**To find or select the html elements you want to style**

* Simple selector – selects elements based on name, id, class
* Combinator selector – select elements based on specific relationship btw them
* Pseudo class selectors – selects elements based on certain state
* Pseudo element selectors – select and style a part of an element
* Attribute selectors – select element based on an attribute or attribute value

**ELEMENT SELECTOR- selects html element by name**

**p{text-align: center; color:blue;}**

**it assigns blue color to all the texts declared inside p tag**

**ID SELECTOR – selects the id attribute of an HTML element to select a specific element**

**#para1 { text-align: center; color: blue; }**

**It assigns blue color to the text of the element with para1 as id**

**CLASS SELECTOR – selects html element with a specific class attribute**

**.center {text-align: center;color: blue;}**

**It assigns blue color to the element with class name as center**

**If you want to assign style to a specific element , use the element name with class selector**

**p.center{text-align:center;color:blue;}**

**UNIVERSAL SELECTOR – selects all elements on the pages**

**\*{color:green;font-size:20px;}**

**\* - symbol used to select all elements**

**GROUP SELECTOR – to select all elements with same style definition**

**Instead of defining same styles separately for each elements, we can just mention the elements separated by comma and declare the style only once.**

**h1,h2,p{color:blue;font-size:20px;}**

**WAYS TO INSERT CSS:**

**External CSS – using a separate file to define the css properties of the element and linking it with the html page**

**Internal CSS – defining the style of the elements inside the <style> tag of the html page**

**Inline CSS – defining the style of the element by using style attribute inside the specific element’s open tag**

**Multiple style sheets:**

**When we use more than one way of styling for the elements, the recent style will be considered as final**

<head>

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

<style>

h1 {

color: orange;

}

</style>

</head>

**Here the style for h1 is defined both in internal and external style sheet, but the h1 will consider the orange color as it is the recent one**

**PRIORITY OF STYLE SHEET :**

1. **Inline style**
2. **External and internal – it will be based on the above example**

**CSS BACKGROUND PROPERTIES**

* **background-color**
* **opacity – specifies the transparency of an element(0.0 to 1.0)**

**use rgba inorder to not inherit the opacity to all the child elements making transparency to both text and background**

**background : rgba(0,128,0,0.3)**

**this only reducckground-es the background color**

* **background-image:url(“image.png”); -- to set the background as image**

**we can set background image for specific elements too**

* **by default the background-image will repeat the image both vertically and horizontally. But sometimes we need only horizontal or only vertical repeatition**

**background-image: url(“image.png”)**

**background-repeat: repeat-x; ----- repeated horizontally**

**repeat-y; --------- repeated verticaly**

**no-repeat; ----- the image won’t be repeated**

* **background-position --- to specify the position of background img**

**the values include top,bottom will by default in the centre, we can use left top,left bottom, right top, right bottom as our wish**

* **background-attachment --- it specifies whether the background image should scroll or be fixed --- the values include fixed, scroll**
* **background ---- to specify all the background properties in one single property – shorthand property**

body {

background-color: #ffffff;

background-image: url("img\_tree.png");

background-repeat: no-repeat;

background-position: right top;

}

**Instead of the above**

body {

background: #ffffff url("img\_tree.png") no-repeat right top;

}

**Order of property when using shorthand property**

background-color

background-image

background-repeat

background-attachment

background-position

**CSS BORDERS – To specify style, width and color of an element’s border**

* **border-style --- to specify the kind of border (dotted,dashed,solid, double, groove, ridge, inset, outset, none, hidden)**

**it can have 1 to 4 values together separated by comma**

* **border-width --- specifies the width for all four sides**

**we can set the size in px, pt, cm, em etc or by using thin, medium or thick**

**when we assign 2 values the first will be for top and bottom and second for sides**

**when we assign 4 values, it will consider in the order top, right, bottom and left**

* **border-color --- to set the color of the four borders (name,rgb,hex,hsl,transparent) the same way as width specifying**
* **border – shorthand property to specify all properties of border in single property**

**border: width style color;**

**we can also specify for individual sides by border-bottom, border-top, border-left, border-right**

* **border-radius – to add rounded borders to an element**

**CSS MARGINS – to create a space around the element outside of border**

* **margin-top, margin-bottom, margin-right, margin-left**
* **margin – shorthand property – order (top,right,bottom, left)**

**margin: auto --- this value horizontally centre the element in its container**

**margin:inherit – this value makes the element to inherit the property value from its parent**

**CSS PADDING – to create spaces around an elements content, inside the border**

* **padding-top, padding-right, padding-bottom, padding-left**
* **the values can be in – px, pt, cm or with % or inherit**
* **padding – shorthand property (top, right, bottom, left)**

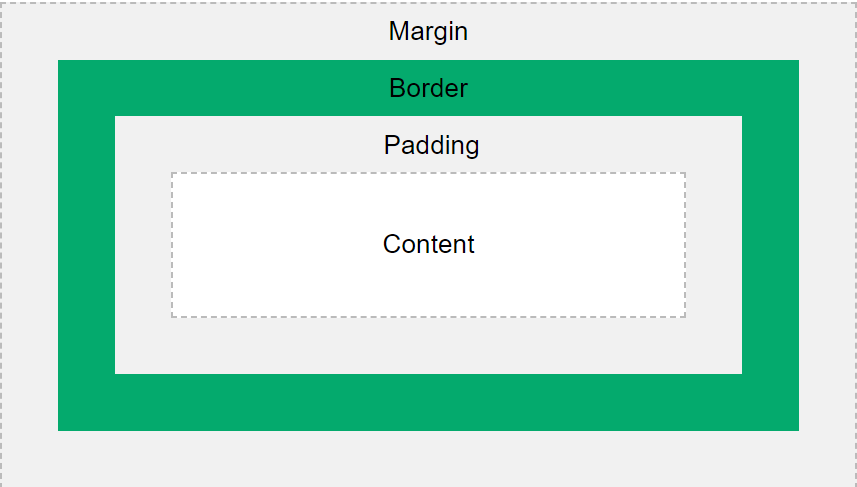
**CSS HEIGHT, WIDTH AND MAX-WIDTH:**

* **height, width – to set the height and width of an element**

**can have values in px, cm, %, initial (default value setting), inherit, auto (browser calc the height and width)**

* **max-width – to set maximum width of an element**

**CSS BOX MODEL:**

****

**CSS OUTLINE – line drawn outside the element border**

* **outline-style --- same as border-style but can have only one of the values**
* **outline-width ---- thin, thick, medium or in px, cm, pt**
* **outline-color ---- same as border color**
* **outline – shorthand property – width, style, color**
* **outline-offset --- adds space btw an outline and edge/border**

**CSS TEXT FORMATTING:**

* **color – to set the color of the text – name,hex,rgb**
* **text-align --- to set the horizontal alignment of the text (center, left, right, justify)**
* **text-align-last – specifies how to align the last line of text (center, left, right, justify)**
* **direction (rtl, ltr) and unicode-bidi (bidi-override,normal, embed) – to change the text direction**
* **vertical-align --- to set the vertical alignment of an element(baseline, text-top, text-bottom, sub, super)**

**TEXT DECORATION:**

* **text-decoration-line --- overline, line-through, underline and can be used together too..**
* **text-decoration-color**
* **text-decoration-style**
* **text-decoration-thickness**
* **text-decoration – shorthand property (line,color,style,thickness)**
* **text-decoration:none --- removes underlines from links and text**

**TEXT TRANSFORMATION:**

* **text-transform --- uppercase, lowercase, capitalize**

**TEXT SPACING:**

* **text-indent --- indentation of the first line (px,pt,%,cm,em etc.,)**
* **letter-spacing --- space btw characters can have + or – values**
* **line-height --- space btw lines**
* **word-spacing --- space btw words**
* **white-space --- to specify white space inside an element (nowrap – whole sentence in one line)**

**TEXT SHADOW:**

* **text-shadow --- order(horizontal shadow(px), vertical shadow(px), blur effect(px), color) this can be written more than once with comma separation after the 4 values**

**CSS FONTS:**

* **font-family --- specify font of a text in “”**

**it can hold several font names as a fallback system if the specified font is not available the browser will find a suitable one.**

**It is done by strtng with the wanted font name and ending with the generic family name with comma as separator**

* **font-style --- normal, italic, oblique**
* **font-weight --- normal, bold**
* **font-variant --- normal, small-caps(lowercase letters converted to uppercase letters but in the lowercase size)**
* **font-size --- px,em [1em = 16px], %, vw- viewport width (it follows the size of the browser window)**
* **google fonts --- To use google fonts, add a special stylesheet link and refer to the font in css**
* **font --- shorthand property order(style, variant, weight, size, family)**

**CSS ICONS**

**CSS LINKS**

* **Links can be styled in different ways and can be styled with respect to what state they are in.**
* **a:link – unvisited link**
* **a:visited – user visited**
* **a:hover – the cursor over it**
* **a:active – the moment it is clicked**

**CSS LISTS**

* **list-style-type --- circle, square, upper-roman, lower-alpha**
* **list-style-image --- specifies an image as the list item marker**

**list-style-image: url(“img.png”);**

* **list-style-position ---(outside, inside) specifies the position of the list item markers**
* **list-style-type:none --- remove markers/bullets**
* **list also has default margin and padding – margin:0 and padding:0 to remove it**
* **list-style --- shorthand property – order(type, position, image)**

**CSS TABLES**

* **border**
* **width, height**
* **border-collapse --- sets the table border to be collapsed into o single border if border-collapse: collapse;**
* **text-align(right, left, center)**
* **vertical-align(top, bottom)**
* **padding – can be used for each cell**
* **:hover --- selector to highlight table rows on mouse hover**

**RESPONSIVE TABLE --- display horizontal scroll bar if screen is too small to display full content**

**Add to a container element – overflow-x:auto – to make it responsive**

**CSS LAYOUT – display property – to specify how an element is shown on a webpage**

**display:none – hides the content along with its space**

**visibility:hidden – only hides the content and not the space**

**position – (static, relative, fixed, absolute, sticky) specifies the type of positioning method used for an element**

**z-index – specifies the stack order of an element,, it specifies which element should be placed in front of, or behind the others**