Section 1 : HTML

# Your First Website :

1. Download Editor for your website. Some of the Editors are Notepad++, Brackets etc.
2. Download Brackets as the Tutor uses Brackets. (brackets.io) path (C:\Users\LocalAdmin\OneDrive\Android\_Blog\Udemy\_WebDevelopmentCourse\Resources\softwarez)
3. Create folder (C:\Users\LocalAdmin\OneDrive\Android\_Blog\Udemy\_WebDevelopmentCourse\Resources\HTML) to add all the Resources shown in the videos.
4. Folder (C:\Users\LocalAdmin\OneDrive\Android\_Blog\Udemy\_WebDevelopmentCourse\Resources\Documents) to add all the documents from the course.

**Resources** : (1.helloworld.html)

# Your Passcode :

1. Your passcode : 9661843876. You will need this in FTP lecture. You need to use the Passcode to get valid voucher.

# 12) Headings:

1. Splitting the Screen in two half with Live preview : Click on live preview icon to right of Brackets.
2. Inspecting element : Open Nature Wikipedia on google 🡪Right click 🡪Inspect (u get HTML Elements)
3. Ctrl+d : to duplicate a line in Brackets.
4. Headings go from <h1> : Biggest to <h6> : smallest

**Resources :** (2.headings.html) not saved.

# 13) Paragraphs :

1. Paragraph : <p>
2. use <br> for line break

# 14) Links:

1. <a href = “” title = “”> text </a>
2. Web links : <a href=”www.google.com” title=”Google” > Link 1 </a>
3. Linking document page : <a href=”3.paragraphs.html” title=”Paragraph” > Link 2 </a>
4. Linking in same page : use “id” <a href=”#ParaId” title=”Paragraph 1” > Link 3 </a>

# 15) Images :

1. Create folder Images inside the HTML Folder and add some pictures to it.
2. Image from folder <img src=”Images/cat1.jpg” width=”200” height=”200” alt=”Cat”>
3. Image from Url : < img src=” <https://freenaturestock.s3.amazonaws.com/1553.jpg>” width=”200” height=”200” alt=”Nature”>

# 16) Inline vs Block Elements :

1. Block elements : Start and end with break line. Eg <p>, <h1>
2. Inline elements : have no line break. Are placed one after the other. Eg <a>, <img>

# 17) Iframes:

1. Embedding existing file : <iframe src=”4.paragraphs.html” width=”200px” height=”200px” frameborder = “1”></iframe>
2. Embedding Link : <iframe name=”myIframe”> </iframe>

<p> <a href=”https//www.google.com” target=”myIframe” > Link 1</a><p>

1. Embeding youtube Video (click on video 🡪 share 🡪Embed🡪copy and paste in code ):

<iframe width="560" height="315" src="https://www.youtube.com/embed/YI-YXVrduBk" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

# 18) Unordered Lists :

1. <ul><li>Item 1</li><li>Item2></li></ul>

# 19) Ordered Lists :

1. <ol type=”A”><li>Item 1</li><li>Item2></li></ol>

# 20) Description List :

1. Used for defining terms
2. <dt>TERM</dt><dd>Description</dd>

# 21) Tables :

1. <table><tr><th>HEADING</th></tr>

<tr><td>DATA</td><tr>

</table>

# 22) Entities :

1. (&nbsp;) ($#160;) : for space
2. &lt; : < symbol
3. &copy; : copyright symbol

# 23) Form 1 : Simple Login Form :

1. <form method=””> : Method : to submit the form (**get** : shows password in address bar. **post** : doesn’t show password
2. <input type=”” name=”” id=””> : Name required to refer to this field by PHP or during Submission. Id required to refer by other elements or in CSS.
3. <label for=””> : to add label for referred by id.

# 24) Form 2 : Marketplace checkout Form:

1. <form method=””>
2. For Dropdown list selection use : <select name=”” value=””>

<option value=””>Free shipping</option>

1. For RadioButton selection : <input type="radio" name="payment" value="paypal">
2. For Assigning Text Area : <textarea rows="4" cols="40" name="comments" id="comments"></textarea>

# 25) Text Decoration :

1. Foe Emphasizing : <em>
2. Bold : <strong>
3. Italic : <i>
4. Striking : <strike>
5. Power and chemistry : <sup>, <sub>

# 26) Comments:

1. <!—Comments -->
2. Shortcut for comment : ctrl+/

SECTION 2 : CSS

# 29) Inline CSS :

1. Mention style attributes inside the tag.
2. <p style="color:coral; font-size: 100px">

# 30) Internal CSS:

1. Define the <style> tag in the head section.
2. Define all the ttributes and their properties in this style.
3. <style>

P {

color: coral;

font-size: 100px;

}

</style>

# 31) External CSS:

1. Create a new Folder CSS.
2. Create anew File in the Folder CSS and name it as style.css.
3. Add the attributes for the Tags in this File. Eg. For tag <p> define attributes in style.css as

P {

color: coral;

font-size: 100px;

}

1. Add a link for style.css in the html File as follows:

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

# 32) Classes and ID :

1. An element can use a single id .
2. Defining id:

#coral {

color: coral;

font-size: 40px;

}

1. An element can use multiple classes. Multiple classes can be defined for an element. If styles in the classes overlap, only one class is considered.
2. Defining class :

.blue{

color: blue;

font-style: italic;

}

1. Defining multiple classes:

<p class="class1 class2">

# 33) Div and Span :

1. Div takes a whole line. <div>
2. Add Google Chrome extension ColorPick Eyedropper to pick color from the websites.
3. Add extension “Brackets Color Picker” in Brackets. Whwnever you want to specify color, click ctrl+Alt+k.
4. Span is an Inline Element. i.e It takes only the needed space and not the whole line. <span>

# 34) Box Model :

1. Go to divand span.html File and open with Google Chrome. 🡪 Right click 🡪Inspect Element.

Inspect the <p> element. It has three elements:padding, Border, margin(default is 16).

# 35) Box Model Padding :

1. padding: 16px; /\* takes padding 16px on all sides \*/
2. padding: 16px 25px 10px 20px; /\* takes padding values for top right bottom left \*/
3. padding: 16px 25px; /\* takes padding values for top&bottom, right&left \*/

# 36) Box Model Borders :

1. Changing div to circle using borders:

height: 100px;

width: 100px;

border-width: 10px;

border-color:cyan;

border-style: ridge;

border-radius: 100px;

# 37) Box Model Outline :

1. Gives a outline to the Element.

# 38) Box Model Margin :

1. Sets a gap between other Elements.

# 39) Background :

1. Copy the images folder to the CSS Folder.
2. background-image: url(Images/cat1.jpg);
3. background-size: 100% 100%; /\* To match the size of Image with the Box give 100%. I less than 100%, the Image is repeated. \*/
4. background-repeat: no-repeat; /\*Image is not repeated. If the size of image is small or if background-size is less than 100% the image gets repeated. \*/
5. background-origin:content-box;/\* Image starts from the <div> Element \*/
6. background-position: left center; /\* Position where the Image is to be displayed \*/

# 40) Floating :

1. Makes the element to be displayed to the defined side. Takes the exact space required.
2. The next element is displayed just after the element ends.
3. To make the element not to be displayed as the nest add syntax “clear: both”.

# 41) Positioning :

1. Static position : Default position.
2. Position : fixed 🡪 All elements are placed on pixel position. Default is 0,0 unless specified by the top and left position.
3. Position : relative 🡪Relative to the original position.
4. Position : absolute 🡪positioning the elements relative to the first non- static parent. If no non-static element then it is placed relative to the other HTML elements.
5. Z-index : 1 🡪 Element with highest value shows on top.

# 42) Display :

1. To hide an element 🡪 visibility : hidden ( This only hides element with blank space shown.)
2. To fully hide an element 🡪 display: none;
3. To display an inline element a block element 🡪 display:block;. Block element is where element is displayed in a seperate block. Inline Elements are displayed right after other Elements.

# 43) Text Decoration :

1. Text-decoration : overline;
2. Font-weight : bold;
3. Font-style: italic;
4. Font-size: 1.5 em ( 1 em = 16px, 1.5em=24px)
5. Letter-spacing : 10px ;
6. Text-shadow : 3px 3px 3px grey (horizontal shadow, vertical shadow, blur radius, color)
7. Text-indent : 40px;

# 44) Text Align:

1. Text-align: right;

# 45) Text-Font :

1. Font-family :sans-serif;
2. Font-family : “Times new Roman” , times, serif;(if first doesn’t work, next font is displayed)

# 46) Text Effects :

1. Overflow:hidden 🡪 When the content is more than the space assigned, it will go out of the box. To avoid this make it hidden
2. White-space : nowrap 🡪 text displayed only on single line. Remaining text is not displayed.
3. Text-overflow : ellipsis 🡪 when no wrap is used and there is more text, ellipsis is displayed at the end. (Not working)
4. Add the hover property to make the extra content visible. Do not use the nowrap property with hover. Eg .bluebox:hover { overflow:visible}
5. Word-break: break-all . Breaks a long word to show in next line. “keep-all” moves the whole word to the next line.
6. Word-wrap : break-word 🡪 in order to break word and also move to next line use this.

# 47) Image Sprites :

1. Instead of loading the same image multiple times, CSS image position is used to display images that are chunks of big image.
2. Makes user experience smooth avoid multiple HTTP requests.
3. To get Image coordinates 🡪Open the image in Brackets and hover mouse over the image.
4. The co-ord to be given for the images are the negatives . eg.

#image1{

width: 50px;

height: 50px;

background: url(Images/cat1.jpg) -120px -50px; /\* Image path and the starting co-ord\*/

}

# 48) Image Opacity :

1. To make the image blur use opacity 🡪 opacity: 0.4; (values from 0-1. 0: brighter 1 it is)

# 49) Styling Lists :

1. List-style : circle 🡪various styles to represent list.
2. list-style-image: url(Images/red.gif) 🡪 use the gif image (50\*50 max)
3. background-image: url(Images/red.gif);
4. background-repeat: no-repeat; 🡪 to see the image only once
5. padding-left: 50px; 🡪 so that image and list name don't overlap
6. background-position: 0px center;

# 50) Styling Links :

1. a:link{

color: #fb0303;

} 🡪 Once the link is clicked the color will be changed. Try using a new link or deleting from your cache or visited links to work properly.

1. a:hover{

font-size: 3em;

} 🡪 Change properties of the link when mouse goes over that link

1. div.bluebox a:active{

font-size: 4em;

} 🡪 Change the properties of link in the div with class bluebox

1. a:visited{

color: darkorange;

} 🡪 change color of link once it is visited.

# 51) Gradients :

1. Smooth transition between colors.
2. background: linear-gradient(blue,white); 🡪 Goes from top blue to bottom white.
3. The syntax varies with the browsers. Inorder to be compatible with all browsers,
4. background: linear-gradient(blue,white); /\* for Google \*/
5. background: -webkit-linear-gradient(blue,white);/\* for Safari \*/
6. background: -o-linear-gradient(blue,white); /\* for Opera \*/
7. background: -moz-linear-gradient(blue,white); /\* for Mozilla Firefox \*/
8. For opacity of the color use : background: linear-gradient(to left, rgba(0,0,255,0),rgba(0,0,255,1)); 🡪 0,0,255 is the color and 0:opacity. Opacity goes from 0-1.
9. Radial gradient 🡪 background: radial-gradient(blue,white,orange);
10. background: radial-gradient(blue 5%,white 25%,orange 70%); 🡪 % represent how big the color is to be.
11. background: radial-gradient(circle, blue 5%,white 25%,orange 70%);

# 52) 2D Transforms :

1. Changing shape size and positioning of Elelments.
2. Rotate 🡪 transform : rotate(30deg)
3. Translate div 🡪 🡪 transform : translate(30px) 🡪 (for left use -30px)

🡪 transform : translate(30px, 40 px) 🡪 move to left and Down.

1. Scaling 🡪 transform: scale (2,2) 🡪 (width,height )
2. Skewing 🡪 transform : skew(10deg, 10deg) 🡪 (X-axis, Y-Axis)

# 53) 3D Transformations :

1. Rotating around the axis 🡪 transform: rotate(30deg);

90 deg shows nothing, as surface becomes flat.

# 54) Transitions :

1. Creating effects when we move from one div to another.
2. Transition: width 1s ; 🡪Property that is going to change and duration.
3. For multiple properties to change 🡪 Transition: width 1s , height 1s, transform 1s;
4. Always add the hover property to work.

# 55) Animations:

1. Animation: myAnimation 3s; 🡪 (Animation name and Duration). Check for browser compatibility or use -webkit-animation.

Define your Animation outside the property:

@keyframes myAnimation {

From{}

To{}

}

1. Animation based on % of Duration 🡪

@keyframes myAnimation {

}

Section 3 : Professional Project – Mathematics Tutorial WebSite (Using HTML and CSS )

# Introduction :

1. Sections in site :
2. Header : Title( abrand) , search Box with search Icon, Background, Input Colors.
3. Menu : Background, Links, Color and Transitions, Styling Borders, adding margins, Padding etc.
4. Content Box :Text, header, Paragraph.
5. Sidebars : Left and Right, Colors, Transitions , Pictures.
6. Footer .

# Header (1) :

1. Create the required Tags : DOCTYPE, <html>, <head>, <body>
2. Create the <meta> and the <title>. Implement the <meta> to add the properties for name, width, initial-scale, user-scalable.
3. In the <body> element create <div id=”container”> for the main Container.
4. Create a External CSS File(styling.css) to place all the styles in that File.
5. Link the styling.css file in the html file by adding the <link rel = “stylesheet” href=”styling.css”> .
6. Update the styling.css for the Container id with width, height, border, margin. Shortcut to update the styles is place your cursor on ID in HTML file and press ctrl+e. the styling will be opened just below where the id is used.
7. Place the Link(Maths the Fun Way) inside the <h1> since we want the link to be also heading.
8. Style the Links inside the Heading h1 in styling.css with font-size(get the correct link font size), font-weight(get the weight of link), color(change the color of the link), text-decoration(to remove underline of link).
9. Create a <div> inside the brand <div> for the SearchBox. Create a form with method=”get” so that if we make the form functional using PHP, the address bar will show the input inside the form.
10. In the form, create input with type=”text” for search text and another input with type=”submit” o submit our search text.
11. UPDATE: Here we can see our form is outside our container. This is because we have assigned height to our #container. Update the height=auto in styling.css.
12. Make the searchBox to float to the right. 🡪 update the #searchbox in styling.css to include the property float: right;
13. Here the Searchbox is to right but again below our brand. This is because our “brand” is a block element while our “searchbox” is a float. Create the #brand in styling.css and make it to float to left.

# Header (2) :

1. In the “header” div, we have all the elements to be floating , so our “header” is collapsed. This should not happen, so we will define a height for the “header”.
2. Give a background to the “searchbox”. Use the linear-gradient for the background. Assign width and height for the searchbox.
3. Make the “submit” button to float to the right and text box to float to the left. Assign the submit class to move the button to the right and class “text” to float to the left.
4. Textbox styling : Make the size of textbox 150px. Add padding, text-size, text-color in the textbox.
5. Submit Button Styling : Add padding, make text bold, text-color as white and background .
6. Add padding to searchbox to make the elements to be in centre. Update the padding of header to make the adjustments.
7. Adding search icon: Go to logomark.com and look for search. Select Icon and change the color. Resize the Icon with the size of box. Add background as image with no-repeat.
8. Make the div “clear” in header and set its property to clear:both.

# Menu :

1. We will be creating the Menu with some transitions. In the “header” create another div with id “menu”. We will have ordered list inside. And each ordered list will be a link.
2. To avoid Bullet points, assign style:none to the <ul>
3. Make the list to be inline : For <li> inside <ul> inside #menu, make the elements to float:left; and also add padding to the elements.
4. Create a styling for the Menu id and add gradient to the background to the menu.
5. Style the text inside the List Item.ie. styling our links : Remove underline and add color to text and make text to be in uppercase.
6. Add padding to the #menu ul li so that elements are more distant from each other. Add height and border-right.
7. To assign bar(pipe) to separate the links: Assign line height to the #menu ul li a. Add Padding to #menu.
8. Last list Item doesn’t have any Border.i.e to remove pipe after the last item: Acces the last list-item of the list #menu ul li:last-child
9. Add some styling to the links so that they change when we hover on them : font-size, font-weight, color.
10. Make our list item to be Active: in the html add class=”active” and add the style for active same as that of hover. i.e. extend the hover to also work with Active links.
11. Add some transition effects that we will see when we hover over the elements : Use the transition property in the link #menu ul li a.

# Introduction Box :

1. Create a div with id in the container.
2. Create a header <h3> and a paragraph <p> to hold the cotent.
3. Add styling to the container like background, color, padding, margin.
4. Create styles for <h3> in introduction. Font-size, color and font-weight.
5. Creates styles for <p> in introduction for font-size and color.

# Sidebars (1) :

1. Create two divs with class=”sidebar” and classes as leftsidebar and rightsidebar.
2. Styling for the class sidebar: Add height, width, border and Border-radius for the sidebar.
3. Styling for the leftsidebar :
4. Styling for the rightsidebar :
5. In the Index File give heading <h3> for both the sidebars and create list of Items.
6. Add padding to the .sidebar.
7. Styling Headings <h3> inside the sidebar: add color and the font-size.
8. Styling Lists : We can see that our list has no bullet, this is because, we have styled the list with ul{ list-style: none;} earlier which applies to all the list in our document. Since our list is a link, we style our links : Remove underlines using text-decoration, font-size and color.
9. Styling links on hovered or active : Add text-decoration, color, font-size, font weight. Also in .sidebar ul li a give transition property.
10. Styling the leftsidebar list items: We need some space between the items. So we will add padding-bottom. Get picture from logomkr.com and get arrow pic and save to images and resize it to height 20px. Also add padding-left so there is some space between image and the text.

# Sidebars (2):

1. Style the rightsidebar : Add the background for list image and give paddings in #rightsidebar ul li. Also adjust the leftsidebar padding so that it fits in the sidebar.

# Footer and Congragulations :

1. Add <div> with class=clear in container after our sidebar.
2. Adding footer : Add <footer> with class=footer. Add a<p> inside with content and copyright information.
3. Create Style for .footer. with height, background, margin-top. Also added width and dispay so that it is displayed below the sidebar.
4. Style the border for searchtextbox: in the .text add the border.
5. Adding Font property to our page i.e body
6. Whenever we use the gradient function, add the cross-browser compatibility lines.
7. **Uploading our website: Pending**

Section 4 : Javascript

# 65) Introduction :

1. Create script for the html file. To create interactive effects to create user experience to be more interactive. Used to change html content and also style.
2. Create a JAVASCRIPT Folder and open using brackets.
3. Create a file alertmessage.html.
4. Javascript code can be written in either head or body section. Best way is to write in body. Also write in the end so that html loads first and then remaining javascript.
5. Inline Javascript : write the js code in the body element inside the <script>.
6. Showing a alert using Inline js : alert("This is an alert message");An alert message will pop-up as file opens in the browser.
7. External Javascript : Write in an external file. Create a folder named js and inside create file code.js. Write the <script> inside this js file.
8. Showing an alert using External js : Create a folder “js” and create a file “code”. Insert the code inside script tag alert("This is an alert message"); in this file. In the html file add the src of this file in the <script type="text/javascript" src="js/code">.

# 67) Buttons :

1. How to use js to respond to Button clicks.
2. Create a button with a alert message to be displayed on the click event of the button.
3. Create another button with id and create a script to display alert message in the script function onclick of the button.
4. Try with other mouse events like ondblclick, onmouseover etc.

# 68) Change HTML Content:

1. Copy the CSS file divandspan.html.
2. Add a button when clicked changes the HTML content.
3. To change the HTML content add a script which access the cdocument elementbyid and innerHTML to change the final facebook content.
4. Also, change the content of Apple div to add an unordered list.

# 69) Change HTML Style:

1. Copy the changeHTMLcontent file.
2. To change the style add a script which access the document elementbyid and style with the style attribute to change the facebook style.
3. Also, change the background of the facebook.
4. Also change fontFamily and fontSize of Aple div.

# 70) Variables and Datatypes:

1. Inside the script, create a var x=3; . To display the variable in the console, : window.console.log(x);
2. Open the console window in the web Browser(fn+F12) -> Console and refresh.
3. Create variables for addition, Boolean, string and Array and try various properties.

# 71) Change HTML using Variables – Activity:Random color generator :

1. Define a variable for color with rgb values
2. Onclick of a button, change the color of the facebook div.
3. Change the apple div by defining rndom colors with random variables using using Math.random and Math.round functions.

# 72) Functions : Activity:Swap the Contents:

1. Defined inside the script. Function fun\_name(){}. The function should be called inside the script.
2. window.alert(fun\_name); displays the function definition in the alert window.
3. Function with a parameter to display the name .
4. Function to add 2 numbers and display in the alert window.
5. To swap the contnt of two div, swap the respective id of divs with one another by accessing through document.getElementById(id1).innerHTML and call the function in onclick method of the button.

# 73) Objects :

1. How to create objects. Objects is a list of properties with corresponding values.
2. Objects are created with eg:

var myPhone = {

make : "Apple",

model : "Iphone 7",

warranty : 12,

color : "white"

};

1. To display objects : window.console.log(myPhone);
2. To delete an object : delete myPhone.warranty;
3. Accessing object property : myPhone.model
4. Assigning an object to other Object : var myOtherPhone = myPhone; This makes both the objects of the same reference. Hence, changing the property of on object , changes the property of all objects.

# 74) Create objects using “new” keyword;

1. Objects can be created using new Keyword as follows :

var myPhone = new Object(); and properties can be set and accessed as myPhone.warranty.

# 75) Object Constructors :

1. Object Constructors are created inside the script. The syntax is function functionname(parameters){properties, methods}
2. To create a new object create the object variable with new keyword passing the parameters eg. var myPhone = new phone("Apple", "Iphone 7", 5,"White");
3. To add a new property , use objectname.property eg. myPhone.condition = "like new";
4. To add a generalized property so that it is used by all the objects created by the constructor, add a prototype eg. phone.prototype.condition = "new";

# 76) Arrays:

1. Arrays can be defined as :

var shoppingList = ["bread", "eggs", "milk"];

var shoppingList = new Array("bread", "eggs", "milk");

1. To get the elements of the array : shoppingList[0]
2. A function can also be written to get the specific element of an array.
3. Editing an item from the array : delete shoppingList[1];
4. To remove an item from an array : use splice : shoppingList.**splice**(1,1);
5. To Add item to specific position : shoppingList.splice(1,0, "Banana", "Apple");
6. To Sorting Array : shoppingList.sort();
7. Reversing Array : shoppingList.reverse();
8. Splitting a String to an array : var myArray = myString.split(' ', 2);

# 77) If and Switch Statements :

1. Create a button to check the speed. Call function onclick of a button, by passing the ID of Button.
2. Use the if else for the speed. Eg: if(speed>70){ window.alert("Your Speed is fast"); }
3. Check the season with the switch case: EG:

switch(season){

case "summer" :

window.alert("It is summer time");

break;

default :

window.alert("I do not recognise this");

# 78) For Loops :

1. For loops can be used for numbers, list etc.
2. Eg: of for loop : for(i=0;i<10; i++){ document.getElementById("myParagraph").innerHTML += "<br/>" +i; }
3. For list elements: for(i=0; i<shoplist.length; i++)

# 79) While Loops – Activity : Spend $1000 randomly in a marketplace:

1. Defining the while loop : while(balance > 0). Always define the loop variable before starting the loop.
2. Make a loop end else it will go into the unstoppable loop.

# 80) Regular Expressions:

1. To search a specific pattern in the string define the pattern eg: var pattern = /a/g;
2. To find a pattern : str.search(pattern);
3. To match a pattern : str.match(pattern)
4. To test a pattern : pattern.test(str);
5. Patterns :
6. /[a-z]/g; // Small alphanumeric characters
7. /[4-7]/g; // Numeric from 4 – 7
8. /\w/g; //Alphanumeric characters
9. /\W/g; // Non alphanumeric characters
10. /\s/g; // Space
11. /\d{10}/g; // Get first non spaced 10 digits
12. /^A/g; // Specific characterand regular "A"
13. /\n/g; // search for new line

# 81) Errors (1) :

1. Define the errors to be displayed in a paragraph <p>. Eg : try{

sum(3,4);

}

catch(err){

document.getElementById("error").innerHTML = err.message;

}

# 82) Errors(2) – Activity : Password Validation

1. Create two input fields for the password1 and password2.
2. Inside the script, create a function check() to validate the passwords.
3. Validate the password using the if-else and Regular expressions for length of password, characters, numbers and whether the passwords match.
4. Create the try-catch statements to catch the errors and display in the <p>.

# 83) Set Interval and Set Timeout – Activity : Create a simple counter

1. To run a piece of code at regular time intervals eg counter increasing by 1 every event – timing event.
2. Load a page and get alert message after some time(3 sec)
3. Eg. Create a counter starting at 0 and every sec goes by 1.
4. Create variable x and initialize to 0. Create paragraph with id = “counter”.
5. In script create var x =0 and var counter and set it to the id of the paragraph
6. Create var myCounter and set it to setInterval function . Function has 2 parameters. First is the function that we would like to run every second. setInterval( function(){x++; counter.innerHTML=x},1000). This will make the counter to go up by 1 sec.
7. Stop the counter onclick of a Button. We use the function clearInterval(myCounter). It is the same parameter we used for the counter.
8. Load the page and run code after 3 sec : Create another var delayedWelcomeMessage and the function to be used is setTimeout(function(){window.alert("Welcome to the Delayed message")}, 3000)
9. Stop getting alert message. Stop the timer: Create a button with the onClick =”clearTimeout(delayedWelcomeMessage)” . To stop seeing the alert, click on the Button within 3 sec. The timer still continues to be running.

# 84) Window and Screen :

1. Height of browser window : window.innerHeight || document.documentElement.clientHeight || document.body.clientHeight;
2. available height of screen : window.screen.availHeight;
3. width of window screen : window.screen.width;
4. To set the dimensions to div : document.getElementById("div1").style.width=window.innerWidth/4-10 + 'px';

document.getElementById("div1").style.height=window.innerHeight-10 + 'px';

1. Change the window height using For loop .

# 85) Alert Boxes :

1. Window.alert method shows an alert message.
2. Setting var to boolean variable with window method to accept cookies. var acceptCookies = window.confirm("Plwase confirm you accept cookies");
3. Alert boxes accepting input : use prompt eg : var visitor = prompt("Please enter your name : ");

Value is stored in the variable visitor.

# 86) Cookies :

1. Creating the cookie : document.cookie="username = name";
2. Accessing the cookies : window.alert(document.cookie); All the cookies are shown separated by a ;. O split the cookies with a , use var cookie = document.cookie.split(';');
3. Cookies can also be accessed by for loop from the cookie array eg. document.getElementById("myPara").innerHTML += "<br/>" + cookie[i];
4. All the cookies are now saved to our webpage. To check all the cookies, delete all the code in Script and write an alert to display cookies. Your cookies will still be displayed.

Section 5 : Professional Project – Maths Game (HTML, CSS & Javascript)

# 87) Project Introduction : Maths Game(HTML, CSS & Javascript)

1. This is a game website. It checks how well you know the multiplication table upto 10.
2. Once we click on the Start Button, countdown for the game for 60 sec will start and we will get our first question of multiplication.
3. We have multiple choices for the answer.
   1. If the correct answer is clicked, we get a green message and our score goes up by 1 and we get a new question.
   2. If we click the wrong answer, we get a red message for clicking the wrong answers.
4. We can reset the game at any time by clicking on the Reset Button.
5. The timer will be continuing until it reaches 0 sec. Once the time is over, we get a message saying Game over along with the score. Also score is reset to 0.

# 88) Page Structure and Styling (1) :

1. Create a new Folder with the Project Name (C:\Users\LocalAdmin\OneDrive\Udemy\_WebDevelopmentCourse\Projects)
2. Open this folder in Brackets and create a new file in it : index.html.
3. Create the HTML prototype for the Project. i.e Add <title>, <meta charset>, <meta name = "viewport" content="width=device-width, initial-scale = 1, user-scalable=yes">
4. Create an external CSS file and link it to our HTML file.
5. In the styling.css create properties of the page i.e for the html element with backround as radial-gradient and height to be 100%.
6. Now we want all the elements to be wrapped in a Blue Box. Let this be <div id=”container”. Add properties to the div as height, width,background-color, margin, padding, border-radius, box-shadow.
7. Create another <div=”score”> in the container and Add Score to this with value in the <span id=”scorevalue”. We will calculate the score later.
8. Style the score with background-color, color, padding, position, left and box-shadow.
9. Create the <div> with id correct and wrong and position them inside container. Add styles to them i.e position, left, background-color, color, padding, display:none(We will not show this all the time. Using JS we will show this box when answer is correct).

# 89) Page Structure & Styling (2) :

1. Create a div in the container for the question with <div id="question"> and create the style for this div i.e. width, height, margin, background-color, box-shadow, font-size, text-align, font-family, color.
2. Create div in the container for instruction, and create style for this div i.e width, height, background-color, margin, text-align, line-height, box-shadow.
3. Create another <div id="choices"> and inside this div create other 4 div with class=”box” for the answer choices. Style the #choice for width, height, margin. Style the .box with width, height, background-color, float, margin-right, border-radius, cursor, box shadow, text-align, line-height. The last box needs no margin on the right. So, style #box4 with margin-right: 0;
4. Add the choice answer text inside the boxes div. Assign the hover property for the box class to change background-color, text color and box-shadow.

# 90) Page Structure and Styling (3) :

1. When the choice boxes are active, we want the box to move down a little, shadow to disappear. Add the active property .box:active with box-shadow and top. In order for the box to move down, add the position to be relative in the .box property.
2. Add transition for the .box property. i.e transition: all 0.2s;
3. Create a new <div id="startreset"> for the button to start/Reset. Style this div with the same property as box (width, padding, background-color, margin, border-radius, cursor, box shadow, text-align, position, transition). Also add the hover and active properties as the box.
4. Create a new <div id="timeremaining"> with the <span id="timeremainingvalue"> with value updated later using JS. Add properties to this div i.e. width, padding,position, top, left, background-color, border-radius, box-shadow, display.

# 91) Page Structure and Styling (4) :

1. Create a new <div id="gameover"> which will be displayed on completion of the timer. We will add content to it later using javascript. Add the properties to this div i.e height, width, background with linear-gradient, color, font-size, text-align, text-transform, position, top, left, z-index, display.

# 92) Game Logic using an Illustrative Flow Chart :

Time Left ?

Show Game Over Message. Chane Button to “Start Game”

Generate new questions and answers

Change Button text to reset Game

Reduce time by 1 sec

Show countdown Box

Click on Start/Reset Button

Reload Page

Are we playing?

YES

NO

YES

NO

# 93) Javascript Code (1) :

1. Create a new javascript file to write the js code in this file. Include this file in the <script src="javascript.js">
2. Define the variables playing as false to check whether the game is in progress and score to track the score of the player.
3. Define the function for the startreset onclick :
   1. If Playing : Reload the page (location.reload)
   2. If not playing :
      1. Set the playing mode to true (playing = true; )
      2. Set the score to 0. (score = 0;)
      3. Update the score value in the <span id="scorevalue"> element (document.getElementById("scorevalue").innerHTML = score;)
      4. Display the timer <div id="timeremaining"> . In styling, we set this as display: none. (document.getElementById("timeremaining").style.display = "block"; )
      5. Change Button to reset game(document.getElementById("startreset").innerHTML = "Reset Game";)

# 94) Javascript Code (2) :

1. Create the variables action for the timer function and timeremaining to track the timer.
2. Create function startCountdown(); to start the counter for the timer.
3. Define the startCountdown() as :
   1. Start decreasing the timeremaining by 1.( timeremaining -= 1; )
   2. Set the value of timer to timeremaining.( document.getElementById("timeremainingvalue").innerHTML = timeremaining;)
   3. Do the following if timeremaining is 0:
      1. When the counter reaches to 0 stop the countdown (stopCountdown(); ). Define the function stopCountdown() as { clearInterval(action); }
      2. Display the game over banner function show(“gameover”) (document.getElementById("gameover").style.display = "block"; )
      3. Display the score. (document.getElementById("gameover").innerHTML = "<p>Game Over!</p><p>Your score is " +score + ".</p>"; )
      4. Hide the timer as the game is over. Function hide(“timeremaining”) (document.getElementById("timeremaining").style.display = "none"; )
      5. Hide the correct and wrong boxes function hide(“correct”), hide(“wrong”).
      6. Set the playing mode to false.
      7. Set the Start/Reset Button to “Start Game”. (document.getElementById("startreset").innerHTML = "Start Game"; )
4. Set the timeremaining value to the start time : timeremaining=10; document.getElementById("timeremainingvalue").innerHTML = timeremaining;
5. Hide the gameover Box hide("gameover");
6. Call the function generateQA() for the new set of questions.

# 95) Javascript Code (3) :

1. Create a function generateQA() with the following :
   1. Create variables for the multiplication of the numbers. Assign random value to them from 1 - 10. (1+Math.round(9\*Math.random()) )
   2. Create a variable for the correct answer. (correctAnswer = x\*y; )
   3. Display the question in the Question Box. (document.getElementById("question").innerHTML=x + "\*" + y ; )
   4. Generate a random Box for the correct answer to be displayed. ( var correctPosition = Math.round(3\*Math.random()); document.getElementById("box" + correctPosition).innerHTML=correctAnswer; )
   5. Fill other boxes with the wrong answers generated randomly.

# 96) Javascript Code (4) :

1. In the function GenerateQA, do the following:
   1. Create an array to store the values in the boxes. Store the correctAnswer first value. (var answers = [correctAnswer]; )
   2. Create a do-while loop to generate random wrong answers, and check if the wrongAnswer is not duplicate of the right answer. (while (answers.indexOf(wrongAnswer)>-1) )
   3. Push the wrong answer in the answer array. (answers.push(wrongAnswer); )
2. Next check if the answer is correct whenever the box is clicked. Do the following in the stert/reset onclick:
3. Get a for loop to check each of the box and do the following in the for loop :
   1. Check if we are playing (if(playing == true))
      1. If the value in the box clicked is the correctAnswer,
         1. increase the score by 1 (score++; )
         2. Display the scorevalue in the score box. (document.getElementById("scorevalue").innerHTML = score;)
         3. Hide the wrong box and show the correct box (hide("wrong"); show("correct");
         4. Hide the correct box after exactly 1 sec. (setTimeout(function(){hide("correct"); }, 1000);)
         5. Generate another Question. (generateQA(); )
      2. else
         1. Hide the correct box and show the wrong box
         2. Hide the wrong box after exactly 1 sec.

Section 6 : JQuery

# 98) Introduction and Loading JQuery :

1. Link jquery.com : jQuery is a fast, small and feature-rich java library. In order to use jQuery, we need to download jQuery file.