<Strong></Strong> – bold

<Em></Em>- italic

<Ins></Ins> – underline

<Br> – linebreak

<P></P> - paragraph

<Hr> – horizontal line

<Sup></Sup> – superscript

<Sub></Sub>- subscript

<Del></Del> – cross the word

<Ul></Ul> – unordered list

<li></li> - lists

<ol></ol> - ordered list

<ol start =10></ol> - starts with 10

<ol type = i></ol> starts with roman numeral

<ol reversed> </ol>- reverse order

<img src=”panda.png”> - pic source(pic should be in same folder as the browser)

<form> - interactive( creates a form)

-------------------------------------------------------------------------------------------------------------------------------------

<input type = “text”> - inputs text

<input type = “text” value = “your name”> ---default value is “your name”

But its not a good way. A better way to guide your username is:

<input type = “text” placeholder =”your name”> ----it gives faded “your name”

----------------------------------------------------------------------------------------------------------------------------------------

<input type = “checkbox”> - inputs checkbox – for default checked(<input type = “checkbox” checked>

-----------------------------------------------------------------------------------------------------------------------------------------

<input type = “radio”> inputs checkbox (round) but can be linked :

<input type = “radio” name = “age”>

<input type = “radio” name = “age”>

Now only one option can be selected

-----------------------------------------------------------------------------------------------------------------------------------

---------------------------------------------------------------------|

<select> |

<option>pizza</option> |--------------drop down menu select

<option>sandwich</option> |

</select> |

----------------------------------------------------------------------|

Here pizza is default because it is at first. But we can select sandwich just by writing

<option selected>sandwich</option>

------------------------------------------------------------------------------------------------------------------------------------------

<input type = “submit”> = gives submits button

<input type = “submit” value = “click me”> = gives submit button but has name click me.

<table> |

<thead> |

<tr>

<th>Name</th> |--------------creates table with head Name, Class

<th>Class</th> |

</tr> |

</thead>

------------------------------------------------------------------------------------------------------------------------------------------

<tbody>

<tr> |

<td>Rohan</td> |

<td>bachelors</td> |-----------fills data in table(Rohan, bachelors)

</tr>

</tbody> |

</table> |

------------------------------------------------------------------------------------------------------------------------------------------

<a href = “http://www.google.com”>click to open google</a> |

<a href = http://www.wikipedia.com”<img src = “Panda.jpg”></a>|------absolute link

-----------------------------------------------------------------------------------------|

< a href = “My webpage”>click here</a> -------------------relative link(our own site)

------------------------------------------------------------------------------------------------------------------------

< p id = “top”>

Adfadsf

Adsfad

Fasdf

Ads<p>

<img src = “Panda.jpg”>

<a href = “#top”> back to top</a>----------------back to where id = “top”

Html entities can be used to display the signs in webpage.

<iframe src = http://www.ecowebhosting.co.uk></iframe>

This displays the content of ecowebhosting in your webpage. In youtube you can get the iframe link to use in the webpage. You can interact with the contents as you would do in the normal webpage.

Inline CSS : <p style = “color:green; font-size:200%;”>CSS is cool!</p>

Internal css: <style type = “text/css”>

P{

Color:gree;

Font-size:200%

}

</style>

-\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class: .red{

color:red;

}

.underlined{

Text-decoration:underline;

}

.size{

Font-size:200%;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<p class="red">

The quick brown fox jumped over the lazy dog.

</p>

We can underline the text in a class. For example;

<p class=”red”>

I love CSS. <span class=”underlined”> And my name is Rohan</span>

</P>

This will make the paragraph red, but it will underline the part from span

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Id: #green{

Color:green;

}

<p id = “green”>

I want this text to be green

</p>

<div id or class = “first-line”>

<p>I am rohan.</p>

<p>adfads</p>

</div>

<p>asdfas</p>

. or #first-line{

Color:white;

Background-color:black;

Width:50%;

}

Here div can affect area within first two lines

Colorzilla: chrome add ons to pick colors from any website.

Color code: code that represents color that you can add on your html files.

Float can be used to position a chunk of contents left or right. Anything written after the float will follow normal flow of page i.e if floats is left and right then the text will be in between them.

Also we can write the text below the float instead of between by adding the following:

.clear-float{

Clear:both;

}

<div class= “clear-float”></div>

<p> this paragraph is not floated</p>

#first-line{

Position: relative;

Left:200px ;(goes 200px to the right)

Left:-50px;(goes 50 px to the left)

Top:200px;(goes 200 px to the down)

Top:-50 px;(goes 50 px to the up)

z-index:1;

z-index:2;

2 will be on top of 1. We need position for z-index to work.

Opacity:1; (total opaque)

Opacity:0;(total transparent)

Opacity:0.5;(half opaque)

Position: absolute;(takes the element out of the flow of the page. That is it is above everything but all the texts and elements ignore it as if it wasn’t there.)

Position: fixed; (similar to absolute but keep the element fixed even if the user scrolls the page)

Margin:100px; (creates 100px margin all around the element)

Margin:100px 50px;(100px top and bottom, 50 px left and right)

Margin: 100px 50px 20px 10px( starts from top,right, bottom, left - clockwise)

Margin-left:100px(creates left margin of 100 px and leaves everything normal). There is right, top and bottom too.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-

Padding:20px 10px 3px 5px; (It is similar to margin but has affect inside the element(box). It creates space between the edge of the box and the text inside it. )

<head>

Body{ --------------------- |

Margin:0; |

|------ This sets the margin and padding of the browser to zero.

Padding:0; |

}---------------------------------|

P{--------------------------|

Margin:0; |------------ This sets the margin and padding of the paragraph tag to zero.

Padding:0; |

}-------------------------- |

</head>

Css reset: different browsers have different default margin and padding. Using css reset can help achieve what you plan to .

Border:10px red solid;

Border-style: solid dotted dashed groove;

Border-color: red blue green yellow;

Border-width: 10px 20px 30px 40px

Border-radius:10px;(curles the border);

Border-radius: 50%; (makes a circle border)

Google Border style to get various styles

P{

Font-family:fantasy; (changes font of the text)

}

Web safe font gives list of fonts that can be used and are supported by various browsers

Font-family: 'shadow into light’,cursive; (This displays the font shadow into light, but if that font is not available in the user’s browser then ‘cursive’ is automatically used. So, it is a good idea to put all the fonts in the family.

Use google fonts to copy the external css in the google server which provide different fonts for free.But first we have to copy the link of css and paste it somewhere between head tags.

Font-style:italic;

Font-weight:bold;

Text-decoration:underline;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

a{ ---------------------------------|

Text-decoration:none; |----------This shows links without underline.

}------------------------------------|

a : hover{ --------------------------| |

text-decoration:underline; |--------This underlines links when a mouse is hovered

| and changes the color to green.

color:green;------------------------|

These are called pseudo classes

a:visited{

color:green;

}

a:link{

color:red;

}

a : active{

color:red;

}

google css styling to know more about the types of styling.

Google lorem ipsum to copy the text so that you can look what your texts look like after you justify the text. It is standard way in internet I don’t know why.

Text-align:left;

Text-align:right;

Text-align:justify; ( Don’t use this often because it makes weird spaces to match the last spacing).

<a href=”<http://www.google.com> target=”\_blank”>google</a> ----opens in new tab

To keep div center aligned: magin:0 auto;

<input type=”image” src=”search.png”> -------This creates search icon which can be clicked.

.no border{

Border;none !important;

}

This emphasizes the importance of having no border. If there are two classes in which one says border and the other says no border, then the one in which ”!important” is written gets executed.

**JAVASCRIPT:**

Inline javascript:

<button onclick=”alert(‘Hi Rohan’)”>Click me</button>----This creates a button with name “click me” which when clicked gives an alert “Hi Rohan”

Internal javascript:

<script type=”text/javascript”>

Alert(“Hi Rohan”);

</script>

We can use chrome console to check the error in our javascript by right clicking on the window and then inspect.

<p id="text">Don't click</p>

<script type="text/javascript">

document.getElementById("text").innerHTML="cLICK";

</script>

Document: we’re taking element from the document we’re working on.

GetElementById(“text”): using the id

InnerHTML=”cLIck” : changes the contents between the html tag.

We can add comment by // or /\* \*/

<p id="text">Don't click</p>

<button id="change">click to change</button>

<script type="text/javascript">

document.getElementById("change").onclick=function(){

document.getElementById("text").innerHTML="What the hell man!";

}

</script>

<p id="text">Don't click</p>

<button id="change">click to change</button>

<p id="text2"></p>

<button id="change2">click</button>

<script type="text/javascript">

document.getElementById("change").onclick=function(){

document.getElementById("text").innerHTML="Rohan";

}

document.getElementById("change2").onclick=function(){

document.getElementById("text2").innerHTML="I think "+document.getElementById("text2").innerHTML+"Awesome";

}

</script>

-\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Document.getElementById(“text”).style.color=”red” or fontSize=”20px” or display=”none”or display=”block”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<input type="text" id="input">

<button id="textChanger">Change the text</button>

<p id="para">This is some text.</p>

<script>

document.getElementById("textChanger").onclick=function(){

var textEnter="";

textEnter=document.getElementById("input").value;

document.getElementById("para").innerHTML=textEnter;

}

</script>

Here, textEnter is a variable which value is empty string. Then we change the value of users input.

<script type="text/javascript">

var myArray= new Array();

myArray[0]="pizza";

myArray[1]="chocolate";

console.log(myArray);

var tweets=["Good morning","I love coffee"]

console.log(tweets);

alert (tweets[1]);

tweets.push("Back to work")

tweets.splice(1,1);

tweets.splice(1,0,"cornflakes")

</script>

Array is used to store many values in one variable. Above are two ways to create an array. Console.log(myArray) shows the names and values of the variable myArray in console when we inspect. We can also use console.log(myArray.length) to just show how many values are stored.

Alert(tweets[1[) pops us a box which says I love coffee.

Tweets.push(“back to work”) adds the value at the end of the array.

tweets.splice(1,1); deletes the tweets starting from array 1(I love coffee) to just 1 item.

tweets.splice(1,2); deletes the tweets starting from array 1(I love coffee) to 2 items(back to work);

tweets.splice(1,0,"cornflakes") : 1 = add at number one. 0= delete no item. “cornflakes”=added item

tweets.splice(1,1,”cornflakes”): replaces the I love coffee.

**IF ELSE FUNCTION:**

<script type="text/javascript">

document.getElementById("press").onclick=function(){

var word="Rohan";

var userChoice= "";

userChoice=document.getElementById("userInput").value;

if (userChoice==word){

alert("You can go");

}else{

alert("What is the magic word?");

}

}

***Variable should be declared insice a function not outside.***

var randomNumber=Math.floor(Math.random()\*11);

math.floor() gives whole number.

Math.random() gives random number between 0 and 1. And when we multiply by 11, we get random number between 0 and 11.

for(i=0; i<5; i++){

alert(i);

}

Here, initital value of i is 0(i = 0). Its value will be increased by 1(i++ or i+1) until the value of i is 4(i<5).

for(i=5; i>0; i--){

alert(i);

}

var tweets=["Good morning","I love coffee","night night"];

console.log(tweets);

for(var i=0;i<3;i++){

alert(tweets[i])

}

We can use tweets.length instead of i<3 because we don’t know how many tweets there will be but tweets.length will count as many tweets as there are.

<div id="tweetDiv"></div>

<script type="text/javascript">

var tweets=["Good morning","I love coffee","night night"];

var tweetString="";

for(var i=0;i<tweets.length;i++){

tweetString=tweetString+tweets[i]+"<br>";

document.getElementById("tweetDiv").innerHTML=tweetString;

}

Tags can be used inside javascript by adding putting them inside “tags”.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**WHILE LOOP:**

Var i=0;

While (i<tweets.length){

TweetString+=tweets[i]+”<br>”;

I++;

}

**FUNCTION:**

Function alertme(){

alert(“Hi Rohan”);

}

alertme()

This alerts whenever alertme() is called in the program.

------------------------------------------------------------------------------------------------

Function average(a,b){

Return((a+b)/2);

}

Var x=2;

Var y=3;

alert(average(x,y));

var start=new Date().getTime();

//some function

var end=new Date().getTime();

Subtract end-start to get time spent.

Variables defined outside the function can be used inside another function but the vairables used inside the function cannot be used outside the function.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-

Document.getElementById(“square”).style.top=`”100px”;

Document.getElementById(“square”).style.left=`”100px”;

Or,

Top = Math.random()\*400

Document.getElementById(“square”).style.top=`top+”px”

We can apply the same method to the left positioning as well. Also width and height.

These can be used to position the object.

**<!doctype html>**

**<html>**

**<head>**

**<title>FunWithJavascript</title>**

**<style type="text/css">**

**#shapes{**

**border:1px black solid;**

**width:200px;**

**height:200px;**

**background-color:bisque;**

**display:none;**

**position:relative;**

**}**

**#time{**

**border:1px black solid;**

**width:70px;**

**height:20px;**

**position:relative;**

**top:-35px;**

**left:70px;**

**text-align:center;**

**}**

**#bestTime{**

**position:relative;**

**top:-70px;**

**left:200px;**

**}**

**#best{**

**border:1px black solid;**

**width:70px;**

**height:20px;**

**position:relative;**

**top:-105px;**

**left:270px;**

**text-align:center;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>Click as fast you can.</h1>**

**<p>Your time:</p>**

**<div id="time"></div>**

**<p id="bestTime">Best time: </p>**

**<div id="best"></div>**

**<div id="shapes"></div>**

**<script type="text/javascript">**

**var startTime = new Date().getTime();**

**var bestTime=0;**

**function colors(){**

**var letters = "ABCDEF0123456789".split("");**

**var color = "#";**

**var i = 0;**

**while (i<6){**

**color+=letters[Math.floor(Math.random()\*16)];**

**i+=1;**

**}**

**return (color);**

**}**

**function showShapes(){**

**document.getElementById("shapes").style.display="block";**

**document.getElementById("shapes").style.top=Math.random()\*400+"px";**

**document.getElementById("shapes").style.left=Math.random()\*1000+"px";**

**document.getElementById("shapes").style.backgroundColor=colors();**

**if(Math.random()>0.5){**

**document.getElementById("shapes").style.borderRadius="50%";**

**}else{**

**document.getElementById("shapes").style.borderRadius="0%";**

**}**

**startTime = new Date().getTime();**

**}**

**function makeShapeAppear(){**

**setTimeout(showShapes,Math.random()\*2000);**

**}**

**makeShapeAppear();**

**document.getElementById("shapes").onclick=function(){**

**var endTime = new Date().getTime();**

**var totalTime = (endTime-startTime)/1000;**

**var yourTime = totalTime;**

**document.getElementById("shapes").style.display="none";**

**document.getElementById("time").innerHTML=yourTime+" sec"**

**makeShapeAppear();**

**}**

**</script>**

**</body>**

**</html>**

<script type="text/javascript" src="javascript.js"></script>

This can be used to link external javascript. We should put it at the end of body section because it will load faster.

**JQUERY:**

**It is a javascript framework i.e, it sits in top of javascript and its main purpose is to make codes simpler. Jqueryui.com gives example of drag, drop, pull, etc plugin.**

Google ‘embed JQuery’ and copy the link and paste in head.

<script type="text/javascript" src="jquery-3.1.1.min.js"></script>

if( typeof jQuery=="undefined")-------here, typeof is used to determine if the variable ‘jQuery’ exists.

if(typeof jQuery=="undefined"){

alert("undefined");

}else{

alert("You are good")

}

JQuery starts with ‘$(“#circle).click(function(){})’

$("#circle").click(function(){

$("p").html("Changed")

This changes the element of p into ‘changed’ when circle is clicked.

We can also write **$(“#circle”).hover(function(){})**

$("iframe").attr("src","http://www.ecowebhosting.co.uk")----this is used to first select iframe and then change the source.

alert($("body").css("width"))---we can get the width of the page.

alert($("body").css("width",”400px”))----we can change the width of the body. Similarly we can change the other css attributes

We can select every div by just writing $(“div”)

$("div").click(function(){

$(this).css("display","none")

‘this’ is used to make image disappear which are clicked. It can be used in javascript

if($(this).attr("id")=="circle"){

alert("You clicked circle")

}else{

alert("You clicked square")

}

The image that is clicked, if it has an id of circle then display ‘you clicked circle’ else ‘you clicked square’

$("div").click(function(){

$(this).fadeOut()

Fades the picture, whichever is clicked, out. We can also use hide() to make image disappear, or fadeIn(). Google jquery fadeout to find ways to add parameters between these () brackets.

For example:

$("div").click(function(){

$(this).fadeOut("slow",function(){

alert("FadeOut complete")

});

})

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

To toggle between fadeIn and fadeout

$("#button").click(function(){

if($("#text").css("display")=="none"){

$("#text").fadeIn();

}else{

$("#text").fadeOut();

}

})

Another way:

Var textShowing=true;

$("#button").click(function(){

if (textShowing){

$("#text").fadeOut(function(){

textShowing=false;

})

}else{

$("#text").fadeIn(function(){

textShowing=true;

})

}

})

**ANIMATION:**

$(this).animate({width:”400px”},2000); ---time in millisecond

$("#circle").click(function(){

$(this).animate({

marginLeft:"500px",

width:"100px",

marginTop:"200px",

},2000,function(){

$(this).css("backgroundColor","red")

})

})

2 sec animation and then the circle changes to red.

var time = new Date().toLocaleTimeString()---shows current time

**Modify placeholder**

::-webkit-input-placeholder{

font-family:cursive;

font-weight:600;

padding-left:20px;

}---

Overflow:scroll; hidden;

**HTML CANVAS**

<script type="text/javascript">

var canvas = document.getElementById("myCanvas"); ------setting the canvas variable

var context = canvas.getContext("2d"); ----specifying if we will draw 2d or 3d image

context.beginPath(); -----usually used at the beginning of each shape

context.fillStyle = "#FBBC05"; -----sets the color of the fill

context.fillRect(100,50,300,100); ------- draws a rectangle (start from co -ordinate 100,50 and then draws width of 300 and height of 100)

context.beginPath();

context.moveTo(100,500); ------start from co-ordinate 100,50

context.lineTo(500,500); ------line ends at co – ordinate 500,500

context.lineWidth = 20 ------width of the line is 20 px;

context.strokeStyle = "#FBBC05"; ---sets color of line

context.stroke(); -----draws line

context.beginPath();

context.arc(100,50,20,0,2\*Math.PI); ----- 100,50🡪starting coordinate, 20 =radius, 0 radiant start, 2pi radiant end) ----draws full circle

context.stroke();

context.fillStyle="black"; ----fill color

context.font = "20px Arial" ---font size and font family

context.fillText("My name is Rohan",200,100); ------ starting position co-ordinate200,100

</script>

<!DOCTYPE html>

<html>

<head>

<title>canvas</title>

<style type = "text/css">

#myCanvas

{

border: 1px solid black;

}

</style>

</head>

<body>

<canvas id = "myCanvas" width = "600" height = "600"></canvas>

<script type = "text/javascript">

var canvas = document.getElementById("myCanvas");

var context = canvas.getContext("2d");

var gradient = context.createLinearGradient(200,200,400,400); ------start from 200,200 and end at 400, 400

var textGradient = context.createLinearGradient(200,50,360,50);

gradient.addColorStop(0,"white"); ---at 0, color = white.

gradient.addColorStop(1,"turquoise"); At 1, color = “turquoise”

textGradient.addColorStop(0,"red");

textGradient.addColorStop(1,"yellow");

context.beginPath();

context.fillStyle=gradient;

context.fillRect(200,200,200,200);

context.beginPath();

context.moveTo(200,200);

context.lineTo(400,400);

context.strokeStyle = "green";

context.stroke();

context.beginPath();

context.moveTo(400,200);

context.lineTo(200,400);

context.strokeStyle = "green";

context.stroke();

context.fillStyle = textGradient;

context.font = "20px Arial";

context.fillText("My name is Rohan maharjan",200,50);

context.stroke();

</script>

</body>

</html>

**ANIMATED CLOCK**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>clock</title>**

**<style type = "text/css">**

**</style>**

**</head>**

**<body>**

**<canvas id = "myCanvas" width = "600" height = "600"></canvas>**

**<script type = "text/javascript">**

**function clock(){**

**var canvas = document.getElementById("myCanvas");**

**var context = canvas.getContext("2d");**

**var clockRadius = context.canvas.width/2;**

**context.beginPath();**

**context.arc(clockRadius,clockRadius,clockRadius,0,2\*Math.PI);**

**context.fillStyle = "black";**

**context.fill();**

**context.closePath();**

**context.beginPath();**

**context.font = clockRadius/10+"px Arial";**

**context.fillStyle = "white";**

**context.textAlign = "center";**

**context.textBaseLine = "middle";**

**context.fillText("12",clockRadius,clockRadius-(clockRadius\*0.9));**

**var i = 1;**

**for( var angle = 2\*Math.PI/12; angle<=2\*Math.PI; angle+=2\*Math.PI/12)**

**{**

**context.fillText(i,clockRadius+clockRadius\*0.9\*Math.sin(angle),clockRadius-clockRadius\*0.9\*Math.cos(angle));**

**i+=1;**

**}**

**context.closePath();**

**var hours = new Date().getHours();**

**var minutes = new Date().getMinutes();**

**var seconds = new Date().getSeconds();**

**var fullHours = hours % 12 + minutes/60 + seconds/3600;**

**var hoursAngle = fullHours \* 2\*Math.PI/12;**

**var minutesAngle = minutes \* 2\*Math.PI/60;**

**var secondsAngle = seconds \* 2\*Math.PI/60;**

**context.beginPath();**

**context.moveTo(clockRadius,clockRadius);**

**context.lineTo(clockRadius+clockRadius\*0.6\*Math.sin(hoursAngle),clockRadius-clockRadius\*0.6\*Math.cos(hoursAngle));**

**context.strokeStyle = "white";**

**context.lineWidth = (clockRadius\*0.04);**

**context.stroke();**

**context.closePath();**

**context.beginPath();**

**context.moveTo(clockRadius,clockRadius);**

**context.lineTo(clockRadius+clockRadius\*0.75\*Math.sin(minutesAngle),clockRadius-clockRadius\*0.75\*Math.cos(minutesAngle));**

**context.lineWidth = clockRadius\*0.024;**

**context.strokeStyle = "white";**

**context.stroke();**

**context.closePath();**

**context.beginPath();**

**context.moveTo(clockRadius,clockRadius);**

**context.lineTo(clockRadius+clockRadius\*0.9\*Math.sin(secondsAngle),clockRadius-clockRadius\*0.9\*Math.cos(secondsAngle));**

**context.strokeStyle = "white";**

**context.lineWidth = clockRadius\*0.01;**

**context.stroke();**

**context.closePath();**

**context.beginPath();**

**context.arc(clockRadius,clockRadius,14,0,2\*Math.PI);**

**context.fillStyle = "white";**

**context.fill();**

**context.closePath();**

**}**

**setInterval(clock,1000);**

**</script>**

**</body>**

**</html>**

**AJAX:**

<script type = "text/javascript">

$.ajax("info.txt").done(function(data){

alert(data);

}).fail(function(){

alert("could not get data");

})

</script>

We are using .ajax because it allows us to give an option if a certain command cannot be performed. In the above example, if we can find info.txt, we alert the text in that file. ‘data’ is just a variable name in which we store the data stored in info.txt. We write the variable inside function parameter.

<script type = "text/javascript">

$.ajax("info.txt")

.done(function(data){

$("#emptyPara").html(data);

})

.fail(function(){

alert("There was a problem.")

})

</script>

Here the content of info.txt was first stored in variable ‘data’, then the html of paragraph tag was changed to the data.

**REGULAR EXPRESSION**

<script type = "text/javascript">

var searchedWord = /Rohan/;

var sentence = "Hi my name is Rohan.Rohan roHaN ";

var result = sentence.match(searchedWord);

alert(result)

</script>

Searches for ‘Rohan’ in the sentence.

If we write /Rohan/I, then the search will be case insensitive.

If we write /Rohan/g , then the result will show as many Rohan as there is in the sentence.

If we write /Rohan/m, then the result will show Rohan from multiple lines.

<p id = “demo”></p>

Var regex = /w3schools/I;

Var str = “Visit my W3Schools!”;

Var result = str.search(regex);

$(“#demo”).HTML(result);

Ans: 9. This gives the starting position of w3schools.

We can also write var regex = “w3schools” and search for it.

<p id = “demo”></p>

Var regex = “W3SCHOOLS”;

Var str = “Visit my W3Schools!”;

Var result = str.search(regex);

$(“#demo”).HTML(result);

No answer, because there is no case insensitive.

**JQuery UI:**

Download stable quick download from jqueryui.com. Then extract files.

<script src = "jquery-ui/jquery-ui.js"></script>-----for javascript

<link href = "jquery-ui/Jquery-ui.css" rel = "stylesheet">----for css

1. **Draggable:**

<div id = “containmentBox”>

<div id = “draggable”><span id = “text”>Drag me</span></div>

</div>

<script type = “text/javascript”>

$(“#draggable”).draggable();---can be dragged anywhere

$(“#draggable”).draggable({axis: “y”});------can be only dragged vertically.

$( "#draggable" ).draggable({containment: "#containmentBox"});-----the draggable div can be dragged only within the containmentBox div

$("#text").draggable({containment: "parent"})----the text can be dragged only within the div it is in i.e “#draggable”. We Can also set the containment of the text to #containmentBox. It will restrict the area of draggable to the area of containmentBox

</script>

1. **Resizable:**

$("#draggable").resizable();---now the draggable div can be resized.

$("#draggable").resizable({grid: 50,

alsoResize: "#containmentBox",

resize: function(event,ui){ ---after resize occurs, function is called

if($("#draggable").width()>300){

alert("big enough");

}

}});-----if you resize draggable div then containmentBox div is also resized. I don’t know yet how to resize text.

$("#containmentBox").resizable();

1. **Droppable:**

<div id = "smallSquare"></div>

<div id = "bigSquare"><span>Drop here!</span></div>

<script type = "text/javascript">

$("#smallSquare").draggable();

$("#bigSquare").droppable({

drop: function(event,ui){

$("#smallSquare").css("backgroundColor","green");

}

});

</script>

1. **Tabs:**

<div id = "tabs">

<ul>

<li><a href = "#tab1">tab1</a></li>

<li><a href = "#myCanvas">tab2</a></li>

</ul>

<div id = "tab1">

<p>Phages:</p>

<select>

<option>Alatin</option>

<option>href</option>

</select>

</div>

<canvas id = "myCanvas" width = "800" height = "800"></canvas>

<button id = "submit">Submit</button>

</div>

<script type = "text/javascript">

$("#tabs").tabs();

$("#submit").click(function(){

$("#tabs").tabs("option","active",$("#tabs").tabs("option","active")+1);----**This opens a tab which is 1 greater than the active tab.**

})

There are many powerful jquery ui. Use it in your project. Tabs, accordion etc.

$("#accordion").accordion();

<!DOCTYPE html>

<html>

<head>

<title>CODE PLAYER</title>

<style type = "text/css">

body{

margin: 0;

padding: 0;

font-family: sans-serif;

}

#container{

width: 100%;

height: 100%;

margin: 0 auto;

}

#header

{

background-color: #EEEEEE;

margin: 0 auto;

padding: 5px;

}

#wordCodePlayer

{

font-size: 200%;

font-weight: bold;

float: left;

position: relative;

top: 10px;

margin-right: 80px;

}

#tabContainer

{

height: 55px;

margin: 0 auto;

width: 335px;

position: relative;

top: 10px;

}

.tabs

{

float: left;

border-left: 1px grey solid;

border-top: 1px grey solid;

border-bottom: 1px grey solid;

font-size: 120%;

padding: 5px 10px;

}

#html

{

border-bottom-left-radius: 5px;

border-top-left-radius: 5px;

}

#output

{

border-right: 1px grey solid;

border-top-right-radius: 5px;

border-bottom-right-radius: 5px;

}

.hovered

{

background-color: grey !important;

}

.active

{

background-color: #E8F2FF;

}

.hidden

{

display: none;

}

.panel

{

resize: none;

border-left: 1px grey solid !important;

border: none;

}

#outputTextArea

{

border: none;

}

</style>

<script type = "text/javascript" src = "jquery.js"></script>

<script src = "jquery-ui/jquery-ui.min.js"></script>

<link href = "jquery-ui/jquery-ui.min.css" rel = "stylesheet">

</head>

<div id = "container">

<div id = "header">

<div id = "wordCodePlayer">

Code Player

</div>

<div id = "tabContainer">

<div id = "html" class = "tabs active">HTML</div>

<div id = "css" class = "tabs">CSS</div>

<div id = "javaScript" class = "tabs">javaScript</div>

<div id = "output" class = "tabs active">Output</div>

</div>

</div>

<div id = "textAreas">

<textArea id = "htmlTextArea" class = "panel editor"><p id = "rohan">Hey! My name is Rohan.</p></textArea>

<textArea id = "cssTextArea" class = "hidden panel editor">p{color: red;}</textArea>

<textArea id = "javaScriptTextArea" class = "hidden panel editor">document.getElementById("rohan").innerHTML = "hI";</textArea>

<iframe id = "outputTextArea" class = "panel"></iframe>

</div>

</div>

<body>

<script type = "text/javascript">

var numOfActiveTabs = 2;

function updateHeight(){

$(".panel").css("height",$(window).height()-$("#header").height());

}

function updateWidth(){

$(".panel").css("width",$(window).width()/numOfActiveTabs-10);

}

function updateOutput(){

$("#outputTextArea").contents().find("html").html("<html><head><style type ='text/css'>"+$("#cssTextArea").val()+"</style></head><body>"+$("#htmlTextArea").val()+"</body></html>");

document.getElementById("outputTextArea").contentWindow.eval($("#javaScriptTextArea").val());

}

$(".tabs").hover(

function(){

$(this).addClass("hovered");

},

function(){

$(this).removeClass("hovered");

})

$(".tabs").click(function(){

if(!$(this).hasClass("active"))

{

$(this).addClass("active");

$(this).removeClass("hovered");

}

else

{

$(this).removeClass("active");

$(this).removeClass("hovered");

}

var areaToShow = $(this).attr("id")+ "TextArea";

$("#"+areaToShow).toggleClass("hidden");

numOfActiveTabs = 4 - $('.hidden').length;-----getting the number of hidden class

updateWidth();

})

updateHeight();

updateWidth();

updateOutput();

$(window).resize(function(){

updateHeight();

updateWidth();

})

$(".editor").on('change keyup paste',function(){

updateOutput();

})

</script>

</body>

</html>

**BOOTSTRAP:**

**YOU CAN COPY AND PASTE THE CODES FROM BOOTSTRAP TO DO A LOT OF THINGS. SO, DEFINITELY CHECK THEM OUT.**

Go to <http://v4-alpha.getbootstrap.com/getting-started/> and then copy starter template to use bootstrap.

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/css/bootstrap.min.css" integrity="sha384-rwoIResjU2yc3z8GV/NPeZWAv56rSmLldC3R/AZzGRnGxQQKnKkoFVhFQhNUwEyJ" crossorigin="anonymous">

<!-- jQuery first, then Tether, then Bootstrap JS. -->

<script src="https://code.jquery.com/jquery-3.1.1.slim.min.js" integrity="sha384-A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k9Omg/a/EheAdgtzNs3hpfag6Ed950n" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js" integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KFbe8fFjk5JAIxUYHKkDx6Qin1DkWx51bBrb" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js" integrity="sha384-vBWWzlZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyeVkE+jo0ieGizqPLForn" crossorigin="anonymous"></script>

<div class = "container">

For 50-50 column

<div class = "row">

<div class = "col-md-6">Name</div>

<div class = "col-md-6">Class</div>

</div>

</div>

**Goes to the second row if the size of screen is less than 720 px. Because, we are using medium.**

**Go to component section and see different cool already built css.**

**NAVBAR:**

**Builds a nav bar.**

<nav class="navbar fixed-top navbar-toggleable-md navbar-light bg-faded"—**creates navbar, fixed in top position, will toggle in medium i.e 720px, has light theme, background color is faded. We can make other colors as well. For example:**

**Success : green**

**Primary: dark blue**

**Secondary: grey**

**Info: light blue**

**Warning: orange**

**Danger: red**

<button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

**Creates a button with pre defined navbar toggler class on right. The navbar is collapsed when toggled. The data that will collapse has id of navbarSupportedContent. Others don’t really matter.**

<a class="navbar-brand" href="#">Navbar</a>

**Everything will collapse under following div.**

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto">

<li class="nav-item active">

<a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item">

<a class="nav-link disabled" href="#">Disabled</a>

</li>

</ul>

<form class="form-inline my-2 my-lg-0">

<input class="form-control mr-sm-2" type="text" placeholder="Search">

<button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>

</form>

</div>

</nav>

**Form:**

<form>

<div class = "form-group row">

<label for = "ExampleInputEmail" class = "col-form-label col-sm-2">Email address</label>

<div class = "col-sm-10">

<input type = "email" class = "form-control" id = "ExampleInputEmail" placeholder = "Enter email">

</div>

</div>

<div class = "form-group row">

<label for = "exampleInputPassword" class = "col-sm-2 col-form-label">Password</label>

<div class = "col-sm-10">

<input type = "password" class = "form-control" id = "exampleInputPassword" placeholder = "Enter password">

</div>

</div>

<div class = "form-group row">

<button type = "submit" class = "btn btn-primary">Submit</button>

</div>

</form>

**Table:**

<table class="table">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Username</th>

</tr>

</thead>

<tbody>

<tr>

<th scope="row">1</th>

<td>Mark</td>

<td>Otto</td>

<td>@mdo</td>

</tr>

<tr>

<th scope="row">2</th>

<td>Jacob</td>

<td>Thornton</td>

<td>@fat</td>

</tr>

<tr>

<th scope="row">3</th>

<td>Larry</td>

<td>the Bird</td>

<td>@twitter</td>

</tr>

</tbody>

</table>

**JUMBOTRON:**