

WEB TECHNOLOGIES-CS225

FIRST MID

Panidem venkata sai keerthi.

Y20cs133

ASSIGNMENT TEST

1A) DEFINE THE FOLLOWING TAGS

I) HEADING TAGS II) IMAGE TAG III) ANCHOR TAG

HTML headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading.

Ex:<h1>this is heading</h1>

The tag is used to embed an image in an HTML page.

The tag has two required attributes:

- src - Specifies the path to the image
- alt - Specifies an alternate text for the image, if the image for some reason cannot be displayed

ex:

The <a> tag defines a hyperlink, which is used to link from one page to another.

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

Ex: Visit W3Schools.com!

i.

<!DOCTYPE html>

Y20CS133

```
<html>
```

```
<body>
```

```
<h1 style="font-size:60px;">Heading 1</h1>
```

```
<h1>Heading 1</h1>
```

```
<h2>Heading 2</h2>
```

```
<h3>Heading 3</h3>
```

```
<h4>Heading 4</h4>
```

```
<h5>Heading 5</h5>
```

```
<h6>Heading 6</h6>
```

```
<p>You can change the size of a heading with the style attribute, using the font-size property.</p>
```

```
</body>
```

```
</html>
```

ii.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
/* This style sets the width of all images to 100%: */
```

```
img {
```

Y20CS133

```
width: 100%;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>Width/Height Attributes</h2>
```

```

```

```

```

```
</body>
```

```
</html>
```

iii.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>The href Attribute</h2>
```

```
<p>HTML links are defined with the a tag. The link address is specified in the href attribute:</p>
```

Y20CS133

Visit W3Schools

</body>

</html>

1B.DESIGN A WEB PAGE WITH AN IMAGE BACKGROUND AND ALSO EMBED AUDIO AND VIDEO INTO YOUR WEB PAGE

<!DOCTYPE html>

<html>

<head>

<style>

body {

background-image: url('img_girl.jpg');

background-repeat: no-repeat;

background-attachment: fixed;

background-size: 100% 100%;

}

</style>

</head>

<body>

<h2>Background Stretch</h2>

<p>Set the background-size property to "100% 100%" and the background image will be stretched to cover the entire element, in this case the body element.</p>

</body>

</html>

2A. EXPLAIN ABOUT THE FOLLOWING TERMS

i) UNORDERED LISTS II) ORDERED LISTS

lists are of different types in that major are unordered lists and ordered lists

li tag represent the list item

In unordered lists we have ul tag to define that it is a unordered list

lists items are represented by bullets by default if we want to change them we have use style="list-style-type:disc" and

disc can be replaced by square or circle or none.

In ordered lists we have ol tag to define that it is a ordered list

lists items are represented by numbers default we can change them to capital letters(A) ,small letters(a),roman numbers

(I or i) by mentioning its type in ol start tag.

in ordered lists we can number the list item in numeric by giving a starting number to them by giving value in

the ol start tag.

HTML lists allow web developers to group a set of related items in lists.

An unordered HTML list:

- Item
- Item
- Item
- Item

An ordered HTML list:

1. First item
2. Second item
3. Third item
4. Fourth item

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>An unordered HTML list</h2>
```

```
<ul>
```

```
<li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Milk</li>
```

Y20CS133

</body>

</html>

o/p:

An unordered HTML list

- Coffee
- Tea
- Milk

<!DOCTYPE html>

<html>

<body>

<h2>An ordered HTML list</h2>

Coffee

Tea

Milk

</body>

Y20CS133

</html>

o/p:

An ordered HTML list

1. Coffee
2. Tea
3. Milk

2B. WRITE HTML CODE TO CREATE A WEB PAGE WHICH DISPLAYS H_2O AND X^2+Y^2

The **<sup>** tag defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font.

The **<sub>** tag defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H_2O .

<html>

<head><title>2c</title></head>

<style>

p{

text-align:center;

font-size:30px;

color:pink;

}

</style>

```
<body>
```

```
<p style="color:blue;">Below represent sub and sup tags</p>
```

```
<p>h<sub>2</sub>O - means water</p>
```

```
<p>x<sup>2</sup>+y<sup>2</sup>=0 represent circle vth radius "0"</p>
```

```
</body>
```

```
</html>
```

3A. WHAT IS CSS? WHAT ARE THE TYPES OF CSS? EXPLAIN.

CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.

Types:

Inline,internal.external

External

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<link rel="stylesheet" href="mystyle.css">
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

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

```
</body>
```

```
</html>
```

```
body {
```

```
    background-color: lightblue;
```

Y20CS133

}

```
h1 {  
  color: navy;  
  margin-left: 20px;  
}
```

internal

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
  background-color: linen;  
}
```

```
h1 {  
  color: maroon;  
  margin-left: 40px;  
}  
</style>  
</head>  
<body>
```

```
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>
```

```
</body>  
</html>
```

Inline:

```
<!DOCTYPE html>  
<html>
```

```
<body>
```

```
<h1 style="color:blue;text-align:center;">This is a heading</h1>
```

```
<p style="color:red;">This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

3B. BRIEFLY DESCRIBE ID SELECTOR AND CLASS SELECTOR

CSS selectors are used to "find" (or select) the HTML elements you want to style.

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#para1 {
```

```
  text-align: center;
```

```
  color: red;
```

Y20CS133

}

</style>

</head>

<body>

<p id="para1">Hello World!</p>

<p>This paragraph is not affected by the style.</p>

</body>

</html>

o/p: **Hello World!**

This paragraph is not affected by the style.

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

Ex:

<!DOCTYPE html>

<html>

<head>

<style>

.center {

Y20CS133

```
text-align: center;

color: red;

}

</style>

</head>

<body>

<h1 class="center">Red and center-aligned heading</h1>

<p class="center">Red and center-aligned paragraph.</p>

</body>

</html>

o/p:
```

Red and center-aligned heading

Red and center-aligned paragraph.

4A. DEVELOP A WEB PAGE CONTAINING TWO IMAGES, WHERE ONE IMAGE OVERLAPS ANOTHER IMAGE BY USING THE Z-INDEX CSS PROPERTY.

Y20CS133

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
img {
```

```
    position: absolute;
```

```
    left: 0px;
```

```
    top: 0px;
```

```
    z-index: -1;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```

```

```
<p>Because the image has a z-index of -1, it will be placed behind the text.</p>
```

```

```

```
</body>
```

```
</html>
```

4B. WITH THE NEAT BLOCK DIAGRAM, EXPLAIN ABOUT CSS BOX MODEL

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:

Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
    background-color: lightgrey;
```

```
    width: 300px;
```

```
    border: 15px solid green;
```

```
    padding: 50px;
```

```
    margin: 20px;
```


Y20CS133

```
}  
  
</style>  
  
</head>  
  
<body>  
  
<h2>Demonstrating the Box Model</h2>  
  
<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>  
  
<div>This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</div>  
  
</body>  
  
</html>
```

5A. EXPLAIN HOW FUNCTIONS CAN BE WRITTEN IN JAVA SCRIPT

JavaScript Functions

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it (calls it).

JavaScript Function Syntax

A JavaScript function is defined with the **function** keyword, followed by a **name**, followed by parentheses **()**.

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

The parentheses may include parameter names separated by commas:

(*parameter1*, *parameter2*, ...)

The code to be executed, by the function, is placed inside curly brackets: {}

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript Functions</h2>
```

```
<p>This example calls a function which performs a calculation, and returns the  
result:</p>
```

```
<p id="demo"></p>
```

```
<script>
```

```
function myFunction(p1, p2) {
```

```
    return p1 * p2;
```

```
}
```

```
document.getElementById("demo").innerHTML = myFunction(4, 3);
```

```
</script>
```

```
</body>
```

```
</html>
```

Output:

JavaScript Functions

This example calls a function which performs a calculation, and returns the result:

12

Function **parameters** are listed inside the parentheses () in the function definition.

Function **arguments** are the **values** received by the function when it is invoked.

Inside the function, the arguments (the parameters) behave as local variables.

The code inside the function will execute when "something" **invokes** (calls) the function

When JavaScript reaches a **return** statement, the function will stop executing.

If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.

Functions often compute a **return value**. The return value is "returned" back to the "caller"

Why Functions?

You can reuse code: Define the code once, and use it many times.

Accessing a function without () will return the function object instead of the function result.

5B. WRITE A JAVASCRIPT FUNCTION TO GENERATE FIVE IMAGES RANDOMLY WHEN A USER CLICKS A BUTTON

```
<html>
```

```
<head>
```

```
<title> Random Image Generator </title>
```

</head>

<script>

```
function getRandomImage() {
```

```
//declare an array to store the images
```

```
var randomImage = new Array();
```

```
//insert the URL of images in array
```

```
randomImage[0] = "https://images.pexels.com/photos/858115/pexels-photo-858115.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500";
```

```
randomImage[1] = "http://www.petsworld.in/blog/wp-content/uploads/2014/09/running-cute-puppies.jpg";
```

```
randomImage[2] = "https://images.pexels.com/photos/142497/pexels-photo-142497.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500";
```

```
randomImage[3] = "https://images.unsplash.com/photo-1543877087-ebf71fde2be1?ixlib=rb-1.2.1&ixid=eyJhcnBfaWQiOjE5MDd9&auto=format&fit=crop&w=500&q=60";
```

```
randomImage[4] = "https://wi.wallpapertip.com/wsimgs/156-1565522_puppies-desktop-wallpaper-desktop-background-puppies.jpg";
```

```
randomImage[5] = "https://images.unsplash.com/photo-1501265976582-c1e1b0bbaf63?ixlib=rb-1.2.1&ixid=eyJhcnBfaWQiOjE5MDd9&auto=format&fit=crop&w=500&q=60";
```

```
//generate a number and provide to the image to generate randomly
```

```
var number = Math.floor(Math.random()*randomImage.length);
```

```
//return the images generated by a random number

return document.getElementById("result").innerHTML = ' Random Image Generator </h2></center>

<h4> Click the button to generate and display random images on the webpage </h4>

<!-- call user-defined getRandomImage function after 2 seconds -->

<button onclick = "setInterval(getRandomImage, 2000)"> Generate Image </button>

<br> <br>

<span id="result" align="center"> </span>

</body>

</html>
```

6A. BRIEFLY DESCRIBE THE DOCUMENT OBJECT MODEL (DOM)

What is the HTML DOM?

The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:

- The HTML elements as **objects**
- The **properties** of all HTML elements
- The **methods** to access all HTML elements
- The **events** for all HTML elements

In other words: **The HTML DOM is a standard for how to get, change, add, or delete HTML elements.**

HTML DOM methods are **actions** you can perform (on HTML Elements).

HTML DOM properties are **values** (of HTML Elements) that you can set or change.

A **property** is a value that you can get or set (like changing the content of an HTML element).

A **method** is an action you can do (like add or deleting an HTML element).

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>My First Page</h2>
```

```
<p id="demo"></p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "Hello World!";
```

```
</script>
```

```
</body>
```

```
</html>
```

Output:

My First Page

Hello World!

The getElementById Method

The most common way to access an HTML element is to use the **id** of the element.

In the example above the **getElementById** method used **id="demo"** to find the element.

The innerHTML Property

The easiest way to get the content of an element is by using the **innerHTML** property.

The **innerHTML** property is useful for getting or replacing the content of HTML elements.

6B. EXPLAIN ABOUT THE FOLLOWING EVENTS WITH EXAMPLES

i) onmousedown and onmouseup ii) onmouseover and onmouseout

The onmousedown, onmouseup and onclick Events

The **onmousedown**, **onmouseup**, and **onclick** events are all parts of a mouse-click. First when a mouse-button is clicked, the onmousedown event is triggered, then, when the mouse-button is released, the onmouseup event is triggered, finally, when the mouse-click is completed, the onclick event is triggered.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<div onmousedown="mDown(this)" onmouseup="mUp(this)"
```

```
style="background-color:#D94A38;width:90px;height:20px;padding:40px;">
```

```
Click Me</div>
```

```
<script>
```

```
function mDown(obj) {
```

Y20CS133

```
obj.style.backgroundColor = "#1ec5e5";  
obj.innerHTML = "Release Me";  
}  
function mUp(obj) {  
    obj.style.backgroundColor="#D94A38";  
    obj.innerHTML="Thank You";  
}  
</script>  
</body></html>
```

The onmouseover and onmouseout Events

The **onmouseover** and **onmouseout** events can be used to trigger a function when the user mouses over, or out of, an HTML element:

```
<!DOCTYPE html>  
  
<html>  
  
<body>  
  
<div onmouseover="mOver(this)" onmouseout="mOut(this)"  
style="background-color:#D94A38;width:120px;height:20px;padding:40px;">  
Mouse Over Me</div>  
  
<script>  
  
function mOver(obj) {
```



```
obj.innerHTML = "Thank You"

}

function mOut(obj) {

    obj.innerHTML = "Mouse Over Me"

}

</script>

</body>

</html>
```

SESSIONAL TEST

1. ANSWER ALL THE FOLLOWING QUESTIONS

A) WHAT ARE THE HTML TAGS WHICH DO NOT HAVE END TAGS?

These elements are called empty elements. Empty elements do not have an end tag!

, <hr>, <link>, <hr>

B) WRITE THE DIFFERENCE BETWEEN ROW SPAN AND COLUMN SPAN?

The rowspan and colspan are <td> tag attributes. These are used to specify the number of rows or columns a cell should span. The rowspan attribute is for rows as well as the colspan attribute is for columns. These attributes have numeric values, for example, colspan=3 will span three columns.

C) LIST THE TYPES OF SELECTORS IN CSS?

A CSS selector selects the HTML element(s) you want to style.

The CSS element Selector

The element selector selects HTML elements based on the element name.

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

D) DEFINE ABSOLUTE POSITIONING AND RELATIVE POSITIONING?

Relative - the element is positioned relative to its normal position. Absolute - the element is positioned absolutely to its first positioned parent.

E) HOW TO FIND MAXIMUM OF THREE NUMBERS? GIVE EXAMPLE.

```
<html>
```

```
<head>
```

```
<title>
```

```
1e
```

```
</title>
```

```
</head>
```

```
<body>
```

```
<script>

var num1=parseInt(prompt("enter 1st number"));

var num2=parseInt(prompt("enter 2nd number"));

var num3=parseInt(prompt("enter 3rd number"));

if(num1 >= num2 && num1 >= num3) {

document.writeln(num1);

}

else if (num2 >= num1 && num2 >= num3) {

document.writeln(num2);

}

else {

document.writeln(num3);

}

</script>

</body>

</html>
```

F) WRITE ANY FOUR METHODS OF STRING OBJECT IN JAVA SCRIPT?

Method	Description
slice()	Extracts a part of a string and returns a newstring
split()	Splits a string into an array of substrings
startsWith()	Checks whether a string begins with specified characters

substr() Extracts a number of characters from a string, from a start index
(position)

2. A) CREATE A BIO-DATA FORM USING THE ALL FORM TAGS

```
<html>
```

```
<head>
```

```
<title>
```

```
labcycle1
```

```
</title>
```

```
</head>
```

```
<body>
```

```
<h1>Bio-Data</h1>
```

```
    <form>
```

```
        <table align="left">
```

```
            <tr>
```

```
                <td><p>Name:</p></td>
```

```
                <td><input type="text"></td>
```

```
            </tr>
```

```
            <tr>
```

```
                <option>
```

```
                <td><p>Gender:</p></td>
```

```
                <td><p>male
```

```
<input type="radio" name="gender" id="male"></td>
<td><p>female
<input type="radio" name="gender" id="female"></td>
</option>
</tr>
<tr>
<td><p>DOB:</p></td>
<td><input type="date"></td>
</tr>
<tr>
<td><p>Phone number:</p></td>
<td><input type="text"></td>
</tr>
<tr>
<td><p>Email ID:</p></td>
<td><input type="text"></td>
</tr>
<tr>
<td><p>Father's Name:</p></td>
<td><input type="text"></td>
</tr>
```

```
<tr>

<td><p>Mother's Name:</p></td>

<td><input type="text"></td>

</tr>

<tr>

<option>

<td><p>Languages Known:</p></td>

<td><p>English

<input type="checkbox" name="Languages Known" id="English"></td>

<td><p>Hindi

<input type="checkbox" name="Languages Known" id="Hindi"></p></td>

<td><p>French

<input type="checkbox" name="Languages Known" id="French"></td>

</option>

</tr>

<tr>

<td><p>Profile pic:</p></td>

<td><input type="file" ></td>

</tr>

<tr>

</tr>
```

```
        <tr></tr>

        <tr>

        <td><input type="button" value="Submit"></td>

        <td><input type="button" value="Cancel"></td>

        </tr>

        </form>

</table>

</body>

</html>
```

2B.DESIGN A WEB PAGE WHICH DISPLAYS THE CLASS TIMETABLE WITH TABLE TAGS AND THEIR ATTRIBUTE.

```
<!DOCTYPE html>

<html>

<head>

<title>time table</title>

<style>

h1{

text-align:center;

}

table,th,td{

width:1100px;
```

Y20CS133

border: 1px solid black;

border-collapse:collapse;

}

th,td{

text-align:center;

padding:17px;

}

</style>

</head>

<body>

<h1>R.V.R & J.C COLLEGE OF ENGINEERING(AUTONOMOUS),GUNTUR-522019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TIME TABLE W.E.F.18-04-2022

II/IV B.TECH(IV-SEMESTER)A.Y 2021-2022</h1>

<table >

<tr>

<th>DAY</th>

<th>8:00-9:00</th>

<th>9:00-10:00</th>

<th>10:00-11:00</th>

Y20CS133

<th>11:00-12:00</th>

<th>12:00-1:00</th>

<th>1:00-2:00</th>

<th>2:00-3:00</th>

</tr>

<tr>

<th>MON</th>

<td colspan="3">WT LAB(C-III)</td>

<td rowspan="6">LUNCH</td>

<td>DBMS</td>

<td>OS</td>

<td>SPORTS</td>

</tr>

<tr>

<th>TUE</th>

<td>SE</td>

<td>E&HV</td>

<td>CS</td>

<td colspan="3">DBMS LAB(C-III)</td>

</tr>

<tr>

Y20CS133

<th>WED</th>

<td>OS</td>

<td>DBMS</td>

<td>WT</td>

<td>SE</td>

<td>CS</td>

<td>LIBRARY</td>

</tr>

<tr>

<th>THUR</th>

<td>SE</td>

<td>OS</td>

<td>DBMS</td>

<td>SOC</td>

<td>CS</td>

<td>WT</td>

</tr>

<tr>

<th>FRI</th>

<td colspan="3">CS LAB(C-III)</td>

<td>CS</td>

Y20CS133

```
<td>DBMS</td>
<td>WT</td>
</tr>
<tr>
<th>SAT</th>
<td colspan="2">SOC LAB(C-III)</td>
<td>WT</td>
<td>E&HV</td>
<td colspan="2">COMPETITIVE CODING(E-LE)</td>
</tr>
</table>
</body>
</html>
```

2C.DESIGN IN DETAIL CONTROL STATEMENTS IN JAVA SCRIPT WITH EXAMPLE

Conditional Statements

- Use **if** to specify a block of code to be executed, if a specified condition is true
- Use **else** to specify a block of code to be executed, if the same condition is false
- Use **else if** to specify a new condition to test, if the first condition is false
- Use **switch** to specify many alternative blocks of code to be executed
- The **while** loop loops through a block of code as long as a specified condition is true.

- The **do while** loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

If statement:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript if</h2>
```

```
<p>Display "Good day!" if the hour is less than 18:00:</p>
```

```
<p id="demo">Good Evening!</p>
```

```
<script>
```

```
if (new Date().getHours() < 18) {
```

```
    document.getElementById("demo").innerHTML = "Good day!";
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Y20CS133

If else statement:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript if .. else</h2>
```

```
<p>A time-based greeting:</p>
```

```
<p id="demo"></p>
```

```
<script>
```

```
const hour = new Date().getHours();
```

```
let greeting;
```

```
if (hour < 18) {
```

```
    greeting = "Good day";
```

```
} else {
```

```
    greeting = "Good evening";
```

```
}
```

```
document.getElementById("demo").innerHTML = greeting;
```

Y20CS133

</script>

</body>

</html>

If elseif else statement:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript if .. else</h2>

<p>A time-based greeting:</p>

<p id="demo"></p>

<script>

const time = new Date().getHours();

let greeting;

if (time < 10) {

 greeting = "Good morning";

} else if (time < 20) {

 greeting = "Good day";

Y20CS133

```
} else {  
    greeting = "Good evening";  
}  
document.getElementById("demo").innerHTML = greeting;  
</script>
```

```
</body>
```

```
</html>
```

Switch:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript switch</h2>
```

```
<p id="demo"></p>
```

```
<script>
```

```
let day;
```

```
switch (new Date().getDay()) {
```

```
    case 0:
```

```
        day = "Sunday";
```

Y20CS133

break;

case 1:

day = "Monday";

break;

case 2:

day = "Tuesday";

break;

case 3:

day = "Wednesday";

break;

case 4:

day = "Thursday";

break;

case 5:

day = "Friday";

break;

case 6:

day = "Saturday";

}

document.getElementById("demo").innerHTML = "Today is " + day;

</script>

Y20CS133

</body>

</html>

While loop:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript While Loop</h2>

<p id="demo"></p>

<script>

let text = "";

let i = 0;

while (i < 10) {

text += "
The number is " + i;

i++;

}

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

</script>

</body>

Y20CS133

</html>

Do while loop:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Do While Loop</h2>

<p id="demo"></p>

<script>

let text = ""

let i = 0;

do {

text += "
The number is " + i;

i++;

}

while (i < 10);

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

</script>

</body>

</html>

2D.WRITE A JAVA SCRIPT CODE TO PRINT 1 TO 10 NUMBERS EXCEPT 5 AND 9.

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Loops</h2>

<p>A loop with a break statement.</p>

<p id="demo"></p>

<script>

let text = "";

for (let i = 1; i < 11; i++) {

if (i === 5 || i===9) { continue; }

text += i + "
";

}

```
document.getElementById("demo").innerHTML = text;
```

```
</script>
```

```
</body>
```

```
</html>
```

3A. DEFINE RECURSIVE FUNCTION AND ALSO WRITE A JAVA SCRIPT TO FIND THE FACTORIAL OF A GIVEN NUMBER.

A recursive function is a function that calls itself during its execution. The process may repeat several times, outputting the result and the end of each iteration.

```
<html>
```

```
<head>
```

```
<title>
```

```
3a
```

```
</title>
```

```
</head>
```

```
<body>
```

```
<script>
```

```
var num=parseInt(prompt("enter a number"));
```

```
var i;
```

```
var f=1;
```

```
for(i=num;i>0;i--)
```

```
{
```

```
f=f*i;
```

Y20CS133

```
}  
document.writeln(f);  
</script>  
</body>  
</html>
```

3B.HOW TO SORT THE GIVEN ARRAY ELEMENTS ? EXPLAIN WITH EXAMPLE.

```
<html>  
<head>  
<title>  
3b  
</title>  
</head>  
<body>  
<script>  
var num=parseInt(prompt("enter number of elements "));  
var a=[];  
var i,j,temp;  
for(i=0;i<num;i++)  
{  
a[i]=parseInt(prompt("enter element"));  
}  
for(i=0;i<num;i++)  
{  
for(j=i+1;j<num;j++)  
{  
if(a[i]>a[j])  
{  
temp=a[i];  
a[i]=a[j];
```

```
a[j]=temp;
}
}
}
for(i=0;i<num;i++)
{
document.writeln(a[i]);
}
</script>
</body>
</html>
```

3C WRITE JAVA SCRIPT TO CHANGE BACKGROUND COLOR USING RADIO BUTTONS.

```
<!doctype html>
<html>
<head>
<meta charset="utf.8" />
<title>radio button</title>
<script>
var cols = {
  "r":"red",
  "b":"blue",
  "p":"pink"
} // no comma after the last
window.onload=function() { // when the page loads
  var rads = document.getElementsByName("music"); // all rads named music
  for (var i=0;i<rads.length;i++) {
    rads[i].onclick=function() {
      document.body.style.backgroundColor=cols[this.value];
    }
  }
}
</script>
</head>
```

```
<body>
<div id= "genre">
  Select color to change
  <br>
  <br>
  <form name="music" method="post" action="">
    <input type="radio" name="music" id="bRad" value="b"><label
for="bRad">Blue</label>
    <br>
    <input type="radio" name="music" id="rRad" value="r"><label
for="rRad">Red</label>
    <br>
    <input type="radio" name="music" id="pRad" value="p"><label
for="pRad">pink</label>
    <br>
    <br>
  </form>
</div>
</body>
</html>
```

3D EXPLAIN ABOUT INNERHTML PROPERTY AND GETELEMENTBYID METHOD WITH EXAMPLE

The programming interface is the properties and methods of each object.

A **property** is a value that you can get or set (like changing the content of an HTML element).

A **method** is an action you can do (like add or deleting an HTML element).

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>My First Page</h2>
```

```
<p id="demo"></p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "Hello World!";
```

```
</script>
```

```
</body>
```

```
</html>
```

In the example above, getElementById is a **method**, while innerHTML is a **property**.

The getElementById Method

The most common way to access an HTML element is to use the id of the element.

In the example above the getElementById method used id="demo" to find the element.

The innerHTML Property

The easiest way to get the content of an element is by using the innerHTML property.

The innerHTML property is useful for getting or replacing the content of HTML elements.

The innerHTML property can be used to get or change any HTML element, including <html> and <body>.