

Internal Assessment Test 2 Answer Key- Dec. 2019

Sub:	Web Technologies				Sub Code:	18MCA13	Branch:	MCA
Date:	17/12/2019	Duration:	90 min's	Max Marks:	50	Sem	I	

Q1) Explain the various levels of style sheet with example for each [10]

1) Inline Style Specification: appears as values of the style attribute of a tag, the general form is as follows:

Style = "Property1 : Value1; Property2 : Value2; Property3 : Value3; Property_n:Value_n;"
It is recommended that last property/value pair be followed by a semicolon.

Eg: <h1 style ="font-family: 'Lucida Handwriting'; font-size: 50pt; color: Red;">Web Technology</h1>

2) Document Style Specification: appears as the content of a style element within the header of a document, general form of the content of a style element is as follows:

<style type = "text/css"> Rule list </style>

Each style rule in a rule list has two parts: a selector, which indicates the tag or tags affected by the rule, and a list of property-value pairs. The list has the same form as the quoted list for inline style sheets, except that it is delimited by braces rather than double quotes. So, the form of a style rule is as follows:

Selector { Property1 : Value1; Property2 : Value2; Property3 : Value3; Property_n:Value_n; }

Eg: <style type = "text/css"> h1 { font-family: 'Lucida Handwriting'; font-size: 50pt; color: Red; } </style>

3) External Style Sheet: have a form similar to that of document style sheets. The external file consists of a list of style rules.

Eg <head> <link rel="stylesheet" type="text/css" href="cssfile.css"> </head>

Cssfile.css P{ Background-color:blue; Color:red; }

Q2a) What are the major uses of JavaScript on Client Side? [04]

The JavaScript was initially introduced to provide programming capability at both the server and client ends of web connection

JavaScript therefore is implemented at 2 ends: 1. Client end 2. Server end

The client side JavaScript is embedded in XHTML documents and is interpreted by the browser

It also provides some means of computation, which serves as an alternative for some tasks done at the server side

This transfer of task ensures that the server is not overloaded and performs only required task

Interactions with users through form elements, such as buttons and menus, can be conveniently described in JavaScript.

Because button clicks and mouse movements are easily detected with JavaScript, they can be used to trigger computations and provide feedback to the user.

For example, when a user moves the mouse cursor from a text box, JavaScript can detect that movement and check the appropriateness of the text box's value (which presumably was just filled by the user). •Even without forms, user interactions are both possible and simple to program in JavaScript. These interactions, which take place in dialog windows, include getting input from the user and allowing the user to make choices through buttons. It is also easy to generate new content in the browser display dynamically.

But client side JavaScript cannot replace server side JavaScript; because server side software supports file operations, database access, security, networking etc.

JavaScript is also used as an alternative to java applets. Programming in JavaScript is much simpler than compared to java

JavaScript support DOM *Document Object Model+ which enables JavaScript to access and modify CSS properties and content of any element of a displayed XHTML document

Q2b) Explain Screen output and keyboard input statement available in JavaScript with examples [06]

- JavaScript models the XHTML document with the Document object.
- The window in which the browser displays an XHTML document is modelled with the Window object.
- The Window object includes two properties, document and window.
- The document property refers to the Document object.

- ❑ The window property is self-referential; it refers to the Window object.
- ❑ write is used to create XHTML code, the only useful punctuation in its parameter is in the form of XHTML tags. Therefore, the parameter of write often includes
.
- ❑ The writeln method implicitly adds "\n" to its parameter, but since browsers ignore line breaks when displaying XHTML, it has no effect on the output.
- ❑ The parameter of write can include any XHTML tags and content.
- ❑ The write method actually can take any number of parameters.
- ❑ Multiple parameters are concatenated and placed in the output. ❑ Example: document.write("The result is: ", result, "
");
- ❑ There are 3 types of pop-up boxes:
 - Alert
 - Confirm
 - Prompt
- ❑ The alert method opens a dialog window and displays its parameter in that window. It also displays an OK button.
- ❑ The string parameter of alert is not XHTML code; it is plain text. Therefore, the string parameter of alert may include \n but never should include
. alert("The sum is:" + sum + "\n");
- ❑ The confirm method opens a dialog window in which the method displays its string parameter, along with two buttons: OK and Cancel.
- ❑ confirm returns a Boolean value that indicates the user's button input: true for OK and false for Cancel. This method is often used to offer the user the choice of continuing some process. var question = confirm("Do you want to continue this download?");
- ❑ After the user presses one of the buttons in the confirm dialog window, the script can test the variable, question, and react accordingly.
- ❑ The prompt method creates a dialog window that contains a text box used to collect a string of input from the user, which prompt returns as its value.

Q3) Discuss all the CSS font properties with example.[10]

- 1) Font Families: The font-family property is used to specify a list of font names. The browser uses the first font in the list that it supports. For example, the property: font-family: Arial, Helvetica, Futura tells the browser to use Arial if it supports that font. If not, it will use Helvetica if it supports it. If the browser supports neither Arial nor Helvetica, it will use Futura if it can. If the browser does not support any of the specified fonts, it will use an alternative of its choosing. If a font name has more than one word, the whole name should be delimited by single quotes
- 2) Font Sizes: The font-size property does what its name implies. For example, the following property specification sets the font size for text to 10 points: font-size: 10pt Many relative font-size values are defined, including xx-small, x-small, small, medium, large, x-large, and xx-large. In addition, smaller or larger can be specified. Furthermore, the value can be a percentage relative to the current font size.
- 3) Font Variants: The default value of the font-variant property is normal, which specifies the usual character font. This property can be set to small-caps to specify small capital characters. These characters are all uppercase, but the letters that are normally uppercase are somewhat larger than those that are normally lowercase.
- 4) Font Styles: The font-style property is most commonly used to specify italic, as in font-style: italic 5) Font Weights: The font-weight property is used to specify the degree of boldness, as in font-weight: bold Besides bold, the values normal, bolder, and lighter can be specified. Specific numbers also can be given in multiples of 100 from 100 to 900, where 400 is the same as normal and 700 is the same as bold.
- 6) Font Shorthands: If more than one font property must be specified, the values can be stated in a list as the value of the font property. The order in which the property values are given in a font value list is important. The order must be as follows: The font names must be last, the font size must be second to last, and the font style, font variant, and font weight, when they are included, can be in any order but must precede the font size and font names. font: bold 14pt 'Times New Roman' 7) Text Decoration: The text-decoration property is used to specify some special features of text. The available values are line-through, overline, underline, and none, which is the default.

Q4) Discuss all the CSS list properties with example. [10]

Two presentation details of lists can be specified in XHTML documents: the shape of the bullets that precede the items in an unordered list and the sequencing values that precede the items in an ordered list. The list-style-type property is used to specify both of these. The list-style-type property of an unordered list can be set to disc, circle, square, or none. Bullets in unordered lists are not limited to discs, squares, and circles

```
<!-- bullets2 -->
<style type = "text/css">
  li.disc {list-style-type: disc}
  li.square {list-style-type: square}
```

```

<li class="circle" (list-style-type: circle)
</style>
...
<h3> Some Common Single-Engine Aircraft </h3>
<ul>
  <li class="disc"> Cessna Skyhawk </li>
  <li class="square"> Beechcraft Bonanza </li>
  <li class="circle"> Piper Cherokee </li>
</ul>

```

Any image can be used in a list item bullet. Such a bullet is specified with the list-style-image property, whose value is specified with the url form.

```

<style type="text/css">
  li.image (list-style-image: url(smaller_airplane.gif))
</style>
...
<li class="image"> Beechcraft Bonanza </li>

```

Table 11 Possible sequencing values for ordered lists

Property Values	Sequence Type	First Four Values
decimal	Arabic numerals	1, 2, 3, 4
upper-alpha	Uppercase letters	A, B, C, D
lower-alpha	Lowercase letters	a, b, c, d
upper-roman	Uppercase Roman numerals	I, II, III, IV
lower-roman	Lowercase Roman numerals	i, ii, iii, iv

Q5) Explain the various selector forms with examples. [10]

1) Simple Selector Forms:

In case of simple selector, a tag is used. If the properties of the tag are changed, then it reflects at all the places when used in the program. The selector can be any tag. If the new properties for a tag are not mentioned within the rule list, then the browser uses default behaviour of a tag. Eg: h1 { font-size : 24pt; } h2, h3 { font-size : 20pt; } body b em { font-size : 14pt; } Only applies to the content of 'em' elements that are descendent of bold element in the body of the document. This is a contextual selector

2) Class Selectors:

Class selectors are used to allow different occurrences of the same tag to use different style specifications. Eg <head> <style type = "text/css"> p.one { font-family: 'Lucida Handwriting'; font-size: 25pt; color: Red; } p.two { font-family: 'Monotype Corsiva'; font-size: 50pt; color: green; } </style> </head> <body> <p class = "one">Web Technology</p> <p class = "two">Web Technology</p> </body>

3) Generic Selectors:

Sometimes it is convenient to have a class of Style specification that applies to the content of more than one kind of tag. This is done by using a generic class, which is defined without a tag name in its name. In place of the tag name, you use the name of the generic class, which must begin with a period. Eg <head> <style type = "text/css"> .sale { font-family: 'Monotype Corsiva'; color: green; } </style> </head> <body> <p class = "sale">Weekend Sale</p> <h1 class = "sale">Weekend Sale</h1> <h6 class = "sale"> Weekend Sale</h6> </body>

4) id Selectors:

An id selector allows the application of a style to one specific element. Eg: <head> <style type = "text/css"> #one { font-family: 'Lucida Handwriting'; font-size: 25pt; color: Red; } #two { font-family: 'Monotype Corsiva'; font-size: 50pt; color: green; } </style> </head> <body> <p id = "one">Web Technology</p> <p id = "two">Web Technology</p> </body>

5) Universal Selectors:

The universal selector, denoted by an asterisk (*), applies its style to all elements in a document. <head> <style type = "text/css"> * { font-family: 'Lucida Handwriting'; font-size: 25pt; color: Red; } </style> </head> <body> <p>Web Technology</p> <p>Web Technology</p> </body>

6) Pseudo Classes:

Pseudo class selectors are used if the properties are to be changed dynamically. For example: when mouse movement happens, in other words, hover happens or focus happens. `<head> <style type = "text/css"> input:focus { font-family: 'lucida calligraphy'; color: purple; font-size:100; } input:hover { font-family: 'lucida handwriting';color: violet; font-size:40; } </style> </head> <body> <form action = " " > <p><label> NAME: <input type = "text" /></label></p> </form> </body>`

Q6) Explain the following [10]

- 1) Math Object 2) Number Object 3) Date Object 4) type of operator

THE Math OBJECT The Math object provides a collection of properties of Number objects and methods that operate on Number objects. The Math object has methods for the trigonometric functions, such as sin (for sine) and cos (for cosine), as well as for other commonly used mathematical operations. Among these are floor, to truncate a number; round, to round a number; and max, to return the largest of two given numbers.

THE NUMBER OBJECT The Number object includes a collection of useful properties that have constant values. Following table lists the properties of Number. These properties are referenced through Number.

Property	Meaning
MAX_VALUE	Largest representable number
MIN_VALUE	Smallest representable number
NaN	Not a number
POSITIVE_INFINITY	Special value to represent infinity
NEGATIVE_INFINITY	Special value to represent negative infinity
PI	The value of π

Any arithmetic operation that results in an error (e.g., division by zero) or that produces a value that cannot be represented as a double-precision floating-point number, such as a number that is too large (an overflow), returns the value "not a number," which is displayed as NaN. If NaN is compared for equality against any number, the comparison fails. The Number object has a method, toString, which it inherits from Object but overrides. The toString method converts the number through which it is called to a string. Example: `var price = 427, str_price; ... str_price = price.toString();`

THE typeof OPERATOR The typeof operator returns the type of its single operand. typeof produces "number", "string", or "boolean" if the operand is of primitive type Number, String, or Boolean, respectively. If the operand is an object or null, typeof produces "object". If the operand is a variable that has not been assigned a value, typeof produces "undefined", reflecting the fact that variables themselves are not typed. Notice that the typeof operator always returns a string. The operand for typeof can be placed in parentheses, making it appear to be a function. Therefore, typeof x and typeof(x) are equivalent.

THE Date OBJECT

A Date object is created with the new operator and the Date constructor, which has several forms. `var today = new Date();` The date and time properties of a Date object are in two forms: local and Coordinated Universal Time (UTC, which was formerly named Greenwich Mean Time). Table 4.5 shows the methods, along with the descriptions, that retrieve information from a Date object.

Method	Returns
toLocaleString	A string of the Date information
getDate	The day of the month
getMonth	The month of the year, as a number in the range of 0 to 11
getDay	The day of the week, as a number in the range of 0 to 6
getFullYear	The year
getTime	The number of milliseconds since January 1, 1970
getHours	The number of the hour, as a number in the range of 0 to 23
getMinutes	The number of the minute, as a number in the range of 0 to 59
getSeconds	The number of the second, as a number in the range of 0 to 59
getMilliseconds	The number of the millisecond, as a number in the range of 0 to 999

Q7) Discuss all general syntactic characteristics of Java Script. [10]

Scripts can appear directly as the content of a <script> tag.

- The type attribute of <script> must be set to "text/javascript".

The JavaScript script can be indirectly embedded in an XHTML document with the src attribute of a <script> tag, whose value is the name of a file that contains the script—for example, <script type = "text/javascript" src = "tst_number.js"> </script>. Notice that the script element requires the closing tag, even though it has no content when the src attribute is included.

In JavaScript, identifiers, or names, must begin with a letter, an underscore (_), or a dollar sign (\$). Subsequent characters may be letters, underscores, dollar signs, or digits. There is no length limitation for identifiers. JavaScript has 25 reserved words. In addition, JavaScript has a large collection of predefined words, including alert, open, java, and self.

JavaScript has two forms of comments, both of which are used in other languages. First, whenever two adjacent slashes (//) appear on a line, the rest of the line is considered a comment. Second, /* may be used to introduce a comment, and */ to terminate it, in both single- and multiple-line comments.

The following XHTML comment used to hide JavaScript uses the normal beginning syntax, <!--. The use of semicolons in JavaScript is unusual. The JavaScript interpreter tries to make semicolons unnecessary, but it does not always work.

When the end of a line coincides with what could be the end of a statement, the interpreter effectively inserts a semicolon there. But this can lead to problems. For example, return x;

The interpreter will insert a semicolon after return, making x an invalid orphan. The safest way to organize JavaScript statements is to put each on its own line whenever possible and terminate each statement with a semicolon. If a statement does not fit on a line, be careful to break the statement at a place that will ensure that the first line does not have the form of a complete statement.

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- hello.html
      A trivial hello world example of XHTML/JavaScript
-->

<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Hello world </title>
  </head>
  <body>
    <script type = "text/javascript">
      <!--
        document.write("Hello, fellow Web programmers!");
      // -->
    </script>
  </body>
</html>
```

Q8) Write a Java script program to find factorial of a number entered by the user. [10]

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Factorial</title>
  </head>
  <body>
    <script type="text/javascript">
      //initialize variables
      var fact=1;
      var n=prompt("Enter a number");
      if(n>0)
      {
        for (i = 1; i <= n; ++i) {
          fact *= i;
        }
        document.write("Factorial of given number is", fact);
      }
      else
        alert("Please enter proper input value");
    </script>
```

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

Q9) Write a XHTML program to create a table with 4 rows and 4 columns, with 2 paragraphs. Design the page by using inline, document and external level style sheet. [10]
Pgm1.xhtml

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Program 2</title>
    <link rel="stylesheet" type="text/css" href="mycss.css" />
    <style type="text/css">

        #two{
          font-style:italic;
        }
        .one{
          font-weight:bold;
          font-family:'Courier New';
        }
    </style>
  </head>
  <body style="background-color:AliceBlue">

    <p class="one" style="margin:0.2in; padding:0.2;background-color:#C0C0C0; border-style:solid;"> The
    Department of MCA was established in 2001.
    The Department offers a postgraduate degree titled 'Master of Computer Applications'.
    The programme is of 3 years (6 semesters) duration. The current student intake for the programme is
    120.</p>

    <p id="two" style="margin:0.1in; padding:0.3;background-color:#C0C0C0; border-style:solid;"> The
    Department of MCA was established in 2001.
    The Department offers a postgraduate degree titled 'Master of Computer Applications'.
    The programme is of 3 years (6 semesters) duration. The current student intake for the programme is
    120.</p>
```

```
<br/>
<table border="1">
  <caption>Fruit Juice Drinks</caption>
  <tr>
    <th></th>
    <th>Apple</th>
    <th>Orange</th>
    <th>Watermelon</th>
  </tr>
  <tr>
    <th>Break fast</th>
    <td>0</td>
    <td>1</td>
    <td>0</td>
  </tr>
  <tr>
    <th>Lunch</th>
    <td>1</td>
    <td>0</td>
    <td>0</td>
  </tr>
  <tr>
    <th>Dinner</th>
    <td>0</td>
    <td>0</td>
```

```

<td>|</td>
</tr>
</table>
</body>
</html>

```

Mycss.css

```

body{
background-color:red;
}

table,th{
border-top-width:medium;
border-top-color:red;
border-top-style:dashed;
border-bottom-width:thick;
border-bottom-color:blue;
border-bottom-style:dotted;
}

p{
font-family:'Arial'
}

```

Q10) Discuss all the methods and properties of string object in Java script [10]
 The String object includes one property, length, and a large collection of methods. The number of characters in a string is stored in the length property as follows:

```
var str = "George"; var len = str.length; //now, len=6
```

Method	Parameters	Result
charAt	A number	Returns the character in the String object that is at the specified position
indexOf	One character, string	Returns the position in the String object of the parameter
substring	Two numbers	Returns the substring of the String object from the first parameter position to the second
toLowerCase	None	Converts any uppercase letters in the string to lowercase
toUpperCase	None	Converts any lowercase letters in the string to uppercase

Consider,
var str = "George";
 Now, **str.charAt(2)** is 'o'
str.indexOf('r') is 3
str.substring(2, 4) is 'org'
str.toLowerCase() is 'george'