

| Regularity<br>(2) | Content<br>(4) | Viva-voce<br>(2) | Timely<br>Submission (2) | Total<br>(10) | Dated Sign of<br>Subject Teacher |
|-------------------|----------------|------------------|--------------------------|---------------|----------------------------------|
|                   |                |                  |                          |               |                                  |

**Assignment No. 03**

Date of Performance: .....

**Title:** Create form in HTML with all form elements apply form validations (e.g. Email, mobile, Pin code, Password).

**Objective:** To study different HTML form elements & JavaScript for Validation.

**Theory: Hyper Text Markup Language (HTML)** is the standard markup language used to create web pages. HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <tr> and </tr>, although some tags represent *empty elements* and so are unpaired, for example <br>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*).

**HTML Form Elements**➤ **HTML <textarea> Tag**

- The <textarea> tag defines a multi-line text input & can hold unlimited number of characters
- The size of a text area can be specified by the cols and rows attributes & can also be specified through CSS height and width properties
- Syntax/Example:

```
<textarea rows="4" cols="50">
```

At w3schools.com you will learn how to make a website. We offer free tutorials in all web development technologies.

```
</textarea>
```

- Attributes:

| Attribute | Value    | Description  |
|-----------|----------|--|
| cols      | number   | Specifies the visible width of a text area           |
| disabled  | disabled | Specifies that a text area should be disabled        |
| name      | text     | Specifies a name for a text area                     |
| rows      | number   | Specifies the visible number of lines in a text area |

➤ **HTML <input > Tag**

- The <input> tag specifies an input field where the user can enter data
- <input> elements are used within a <form> element to declare input controls that allow users to input data
- Syntax/Example:

```
<input type="text" name="name" value="Enter your name">
```

- An input field can vary in many ways, depending on the type attribute:

| Attribute | Value  | Description  |
|-----------|--|--|
| accept    | audio/*<br>video/*<br>image/*  | Specifies the types of files that the server accepts (only for type="file")  |
| align     | Left/ right/ top/ middle/<br>bottom  | Specifies the alignment of an image input (only for type="image")  |
| alt       | text   | Specifies an alternate text for images (only for type="image")   |
| checked   | Checked  | Specifies that an <input> element should be pre-selected when the page loads (for type="checkbox" or type="radio") |
| name      | text   | Specifies the name of an <input> element   |
| size      | number   | Specifies the width, in characters, of an <input> element  |
| src       | URL  | Specifies the URL of the image to use as a submit button (only for type="image")                                   |
| type      | Button/ checkbox/ color/ date<br>/datetime / email/ file/ hidden/<br>image/ month/ number/<br>password/ radio/ range/<br>reset/ search/ submit/ text/<br>time/ url/ week | Specifies the type <input> element to display  |
| value     | text   | Specifies the value of an <input> element  |

#### ➤ HTML <select> Tag

- The <select> element is used to create a drop-down list
- The <option> tags inside the <select> element define the available options in the list
- Syntax/Example:

```
<select>
  <option value="">text</option>
</select>
```

- Attributes:

| Attribute | Value    | Description   |
|-----------|----------|---|
| disabled  | disabled | Specifies that a drop-down list should be disabled        |
| multiple  | multiple | Specifies that multiple options can be selected at once   |
| name      | name     | Defines a name for the drop-down list                     |
| size      | number   | Defines the number of visible options in a drop-down list |

#### ➤ HTML <option> Tag

- The <option> tag defines an option in a select list
- The <option> elements go inside a <select> or <datalist> element
- Syntax/Example:

```
<option value="">text</option>
```

- Attributes:

| Attribute | Value    | Description   |
|-----------|----------|---|
| disabled  | disabled | Specifies that an option should be disabled                         |
| label     | text     | Specifies a shorter label for an option                             |
| selected  | selected | Specifies that an option should be pre-selected when the page loads |
| value     | text     | Specifies the value to be sent to a server                          |

### ➤ HTML <button> Tag

- The <button> tag defines a clickable button
- Different from <input> as inside <button> element you can put content, like text or images.
- Syntax/Example:

```
<button type="button">Click Me</button>
```

- Attributes:

| Attribute | Value                 | Description                                |
|-----------|-----------------------|--|
| disabled  | disabled              | Specifies that a button should be disabled |
| name      | name                  | Specifies a name for the button            |
| type      | Button/ reset/ submit | Specifies the type of button               |
| value     | text                  | Specifies an initial value for the button  |

**JavaScript (JS)** is a dynamic computer programming language and increasingly considered an "assembly" language (a compiler target). It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side network programming (with Node.js), game development and the creation of desktop and mobile applications. JavaScript is classified as a prototype-based scripting language with dynamic typing and has first-class functions. It is the language for HTML, Web, computers, servers, laptops, tablets, etc.

### Why Study JavaScript?

JavaScript is one of **3** languages all web developers **MUST** learn:

1. **HTML** to define the content of web pages
2. **CSS** to specify the layout of web pages
3. **JavaScript** to program the behavior of web pages

### The <script> Tag

- In HTML, JavaScripts must be inserted between <script> and </script> tags.
- JavaScripts can be put in the <body> and in the <head> section of an HTML page.
- The <script> and </script> tells where the JavaScript starts and ends.
- The lines between <script> and </script> contain the JavaScript code:

```
<script>
    function myFunction()
    {
        document.write("Hello");
    }
</script>
```

### JavaScript in <head> or <body>

- You can place any number of scripts in an HTML document.
- Scripts can be placed in the <body> or in the <head> section of HTML, and/or in both.
- Scripts can be at bottom of the <body> section of a web page. This can reduce display time.
- Sometimes you will see all JavaScript functions in the <head> section.
- Separating HTML and JavaScript, by putting all the code in one place, is always a good habit.

### External JavaScript

- Scripts can also be placed in external files with **file extension .js**.
- External scripts are practical when the same code is used in many different web pages.
- To use an external script, put the name of the script file in the source (src) attribute of the <script> tag: `<script src="myScript.js"></script>`

### JavaScript Variables

- As with algebra, JavaScript variables can be used to hold values ( $x=5$ ) or expressions ( $z=x+y$ ).
- Variable can have short names (like  $x$  &  $y$ ) or more descriptive names (age, sum, totalVolume).
- Variable names can contain letters, digits, underscores, and dollar signs.
  - ✓ Variable names must begin with a letter or \$ or \_ (but we will not use \$ & \_)
  - ✓ Variable names are case sensitive ( $y$  and  $Y$  are different variables)
  - ✓ Reserved words (like JavaScript keywords) cannot be used as variable names

### JavaScript Data Types

| Data Types      | Objects | Null Data Type |
|-----------------|---------|----------------|
| String, Number  | Object  | Null           |
| Boolean, Object | Date    | Undefined      |
| Function        | Array   | -              |

### The typeof Operator

- We can use the **typeof** operator to find the data type of a JavaScript variable.
- Examples,
  - `typeof "John"` // Returns string
  - `typeof 3.14` // Returns number
  - `typeof NaN` // Returns number
  - `typeof false` // Returns Boolean
  - `typeof [1,2,3,4]` // Returns object
  - `typeof {name:'John', age:34}` // Returns object
  - `typeof new Date()` // Returns object
  - `typeof function () {}` // Returns function
  - `typeof myCar` // Returns undefined (if myCar is not declared)
  - `typeof null` // Returns object

### JavaScript User Defined Functions

- The real power of JavaScript comes from its functions; it has more than 1000 built-in functions.
- Besides the built-in JavaScript functions, we can create our own functions.
- A function is a block of statements that can be used repeatedly in a program.
- A function won't execute immediately when a page gets load but executed by call to function.
- A function name can start with a letter or underscore but not a number.
- A user defined function declaration starts with the keyword "function".

```
function functionName() {
```

```
Code to be executed;
```

```
}
```

### JavaScript Conditional Statements

- **if statement:** executes some code only if a specified condition is true

```
if (condition) {
    code to be executed if condition is true;
}
```

- **if...else statement:** executes some code if a condition is true and another code if the condition is false

```
if (condition) {
    code to be executed if condition is true;
}
Else {
    code to be executed if condition is false;
}
```

- **if...elseif...else statement:** selects one of several blocks of code to be executed

```
if (condition) {
    code to be executed if condition is true;
}
elseif (condition) {
    code to be executed if condition is true;
}
else {
    code to be executed if condition is false;
}
```

- **switch statement:** selects one of many blocks of code to be executed

```
switch(n)
{
    case label1: code to be executed if n=label1;
                break;
    case label2: code to be executed if n=label2;
                break;
    default: code to be executed if n is different from all labels;
}
}
```

### JavaScript Looping Statements

- **While:** loops through a block of code as long as the specified condition is true

```
while (condition is true)
{
    code to be executed;
}
```

- **do...while:** loops through a block of code once, and then repeats the loop as long as the specified condition is true

```
do {
```

- ```

        code to be executed;
    } while (condition is true);
– for: loops through a block of code a specified number of times
    for (init counter; test counter; increment counter) {
        code to be executed;
    }

```

### JavaScript Date Methods

- Date methods let you get and set date values (years, months, days, minutes, seconds, milliseconds)
- Date Get Methods
  - Get methods are used for getting a part of a date. Here are the most common (alphabetically):

| Method            | Description                                       |
|-------------------|---------------------------------------------------|
| getDate()         | Get the day as a number (1-31)                    |
| getDay()          | Get the weekday a number (0-6)                    |
| getFullYear()     | Get the four-digit year (yyyy)                    |
| getHours()        | Get the hour (0-23)                               |
| getMilliseconds() | Get the milliseconds (0-999)                      |
| getMinutes()      | Get the minutes (0-59)                            |
| getMonth()        | Get the month (0-11)                              |
| getSeconds()      | Get the seconds (0-59)                            |
| getTime()         | Get the time (milliseconds since January 1, 1970) |

- Date Set Methods
  - Set methods are used for setting a part of a date. Here are the most common (alphabetically):

| Method            | Description                                        |
|-------------------|----------------------------------------------------|
| setDate()         | Set the day as a number (1-31)                     |
| setFullYear()     | Set the year (optionally month and day yyyy.mm.dd) |
| setHours()        | Set the hour (0-23)                                |
| setMilliseconds() | Set the milliseconds (0-999)                       |
| setMinutes()      | Set the minutes (0-59)                             |
| setMonth()        | Set the month (0-11)                               |
| setSeconds()      | Set the seconds (0-59)                             |
| setTime()         | Set the time (milliseconds since January 1, 1970)  |

**Conclusion:** Thus, we have studied various tags required to design form element in HTML & basics about JavaScript.