

JavaScript



Content

- JavaScript General Concepts

General Concepts

What is JavaScript

- Is a lightweight, interpreted programming language with object-oriented capabilities that allows you to build interactivity into otherwise static HTML pages.
- JavaScript can be used for Client-side developments as well as Server-side developments.
- JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.

General Concepts

What is JavaScript

- **Client-side JavaScript:** It supplies objects to control a browser and its Document Object Model (DOM). For example, when the client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation.
- **Server-side JavaScript:** It supplies objects relevant to running JavaScript on a server. For example, if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.

General Concepts

Using JavaScript

- In HTML, JavaScript code is inserted between `<script>` and `</script>` tags.
- JavaScript can be added to HTML documents in 3 ways:
 - **Inline** - using a `<script>` tag inside the body of the HTML
 - **Internal** - using a `<script>` tag in the `<head>` section
 - **External** - using a `src` (source) attribute of a `<script>` tag to link to an external JavaScript file
 - External scripts should have the file extension **.js**
 - You can place an external script reference in `<head>` or `<body>` as you like.
 - External scripts cannot contain `<script>` tags

General Concepts

Inline JavaScript

- Using Inline JavaScript, the `<script>` tag is placed inside the body of the HTML, with all the JavaScript code inside the `<script>` tag

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript in Body</h2>
```

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

```
<script>
```

```
document.getElementById("demo").innerHTML = "My First  
JavaScript";
```

```
</script>
```

```
</body>
```

```
</html>
```

General Concepts

Internal JavaScript

- Using Internal JavaScript, the `<script>` tag is placed inside the `<head>` of the HTML, with all the JavaScript code inside the `<script>` tag

```
<!DOCTYPE html>
```

```
<html>
```

```
<head><title> JavaScript in Head</title>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "My First  
JavaScript";
```

```
</script> </head>
```

```
<body>
```

```
<h2>Internal JavaScript</h2>
```

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

```
</body>
```

```
</html>
```

General Concepts

External JavaScript

- Using External JavaScript, the JavaScript code is placed in a separate file and the src (source) attribute is used to <script> tag is placed inside the <head> of the HTML, with all the JavaScript code inside the <script> tag

```
<!DOCTYPE html>
```

```
<html>
```

```
<head><title> External JavaScript in Head </title>
```

```
<script src="/myjs/myjsScript.js"></script>
```

```
</head>
```

```
<body>
```

```
<h2>External JavaScript</h2>
```

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

```
</body>
```

```
</html>
```


JavaScript Syntax

JavaScript Variables

- Variables are used to store data values.
- JavaScript Variables can be declared in 4 ways:
 - Automatically: When first used
 - Using **var** : Was used in all JavaScript code from 1995 to 2015
 - Using **let** : Were added to JavaScript in 2015
 - Using **const** : Should only be used in code written for older browsers
- It is considered good programming practice to always declare variables before use
- JavaScript is **case sensitive**
- The variables `lastName` and `lastname`, are two different variables

JavaScript Syntax

JavaScript Basic Operators

- Arithmetic operators (+, -, * , /, %) to compute values
- Assignment operator (=) to assign values to variables

JavaScript Syntax

JavaScript Data Types

- JavaScript has 8 Datatypes:

- String - Number - BigInt
- Boolean - Undefined - Null
- Symbol - Object

- **Note:** JavaScript variable can hold any type of data
- It has dynamic types. This means that the same variable can be used to hold different data types

JavaScript Syntax

JavaScript Expressions

- A combination of values, variables, and operators, which computes to a value:
- For example:
 - `5 * 10` evaluates to 50
 - `"John" + " " + "Doe"`, evaluates to "John Doe"

JavaScript Syntax

JavaScript Comments

- Code after double slashes `//` or between `/*` and `*/` is treated as a comment
- Comments are ignored, and will not be executed

JavaScript Syntax

Conditional Statements

- JavaScript have the following 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

JavaScript Syntax

Control structure Statements

- JavaScript supports different kinds of loops:
 - **for** - loops through a block of code a number of times
 - **for/in** - loops through the properties of an object
 - **for/of** - loops through the values of an iterable object
 - **while** - loops through a block of code while a specified condition is true
 - **do/while** - also loops through a block of code while a specified condition is true

JavaScript Syntax

JavaScript functions

- A JavaScript function is a block of code designed to perform a particular task
- A function is executed when "something" invokes it (calls it). Example:
 - When an event occurs (eg. when a user clicks a button)
 - When it is invoked (called) from JavaScript code
 - Automatically (self invoked)

JavaScript Syntax

JavaScript Events

- HTML events are "things" that happen to HTML elements. It can be something the browser does, or something a user does.
- When JavaScript is used in HTML pages, JavaScript can "react" on these events.
- For example:
 - An HTML web page has finished loading
 - An HTML input field was changed
 - An HTML button was clicked
- HTML allows event handler attributes, ***with JavaScript code***, to be added to HTML elements.
- JavaScript lets you execute code when events are detected.

JavaScript Syntax

JavaScript Events Handlers

- Event handlers can be used to handle and verify user input, user actions, and browser actions.
- For example:
 - Things that should be done every time a page loads
 - Things that should be done when the page is closed
 - Action that should be performed when a user clicks a button
 - Content that should be verified when a user inputs data

JavaScript Syntax

JavaScript Events Handlers

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Event Handling Example</title>
</head>
<body>
  <h1>Event Handling Example</h1>
  <button id="myButton">Click Me!</button>

  <script src="script.js"></script>
</body>
</html>
```

JavaScript Syntax

JavaScript Events Handlers

```
// Select the button element
const button = document.getElementById('myButton');

// Define the event handler function
function handleClick() {
    alert('Button was clicked!');
}

// Attach the event handler to the button's click event
button.addEventListener('click', handleClick);
```

JavaScript Syntax

Form validation with JavaScript

- JavaScript provides a way to validate form's data on the client's computer before sending it to the web server
 - Basic Validation – The form must be checked to make sure all the mandatory fields are filled in.
 - Data Format Validation – The data that is entered must be checked for correct format and value.
-

JavaScript Syntax

Form validation with JavaScript

```
<html>
  <head>
    <title>Form Validation</title>
    <script type = "text/javascript">
      // Form validation code will come here.
    </script>
  </head>
  <body>
    <form action = "#" name = "myForm" onsubmit = "return(validate());">
      <table cellpadding = "2" cellspacing = "2" border = "1">
        <tr>
          <td align = "right">Name</td>
          <td><input type = "text" name = "Name" /></td>
        </tr>
        <tr>
          <td align = "right">EMail</td>
          <td><input type = "text" name = "EMail" /></td>
        </tr>
        <tr>
          <td align = "right">Zip Code</td>
          <td><input type = "text" name = "Zip" /></td>
        </tr>
        <tr>
          <td align = "right">Country</td>
          <td>
            <select name = "Country">
              <option value = "-1" selected>[choose yours]</option>
              <option value = "1">USA</option>
              <option value = "2">UK</option>
              <option value = "3">INDIA</option>
            </select>
          </td>
        </tr>
        <tr>
          <td align = "right"></td>
          <td><input type = "submit" value = "Submit" /></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

JavaScript Syntax

Form validation with JavaScript

```
<script type = "text/javascript">
  // Form validation code will come here.
  function validate() {

    if( document.myForm.Name.value == "" ) {
      alert( "Please provide your name!" );
      document.myForm.Name.focus() ;
      return false;
    }
    if( document.myForm.Email.value == "" ) {
      alert( "Please provide your Email!" );
      document.myForm.Email.focus() ;
      return false;
    }
    if( document.myForm.Zip.value == "" || isNaN( document.myForm.Zip.value ) ||
      document.myForm.Zip.value.length != 5 ) {

      alert( "Please provide a zip in the format #####." );
      document.myForm.Zip.focus() ;
      return false;
    }
    if( document.myForm.Country.value == "-1" ) {
      alert( "Please provide your country!" );
      return false;
    }
    return( true );
  }
</script>
```