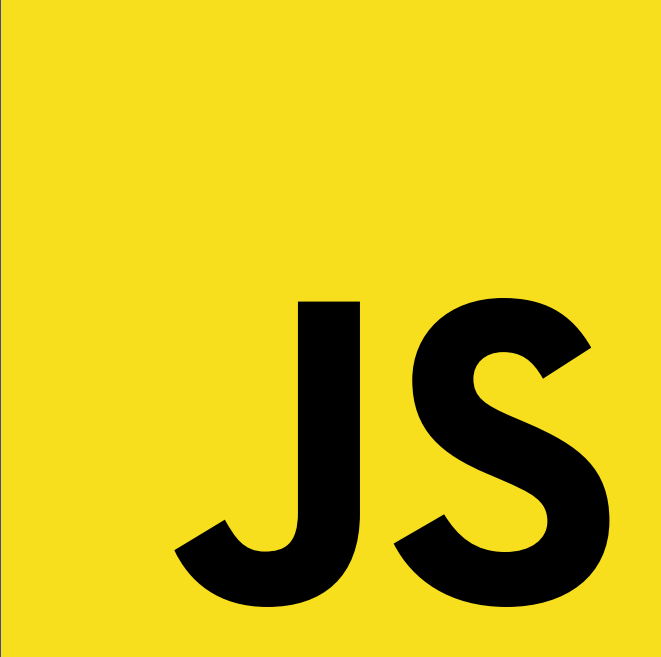


JavaScript Fundamental

The JavaScript logo, featuring the letters 'JS' in a bold, black, sans-serif font, centered within a bright yellow square. This yellow square is positioned on the right side of the image, overlapping a stack of three blue squares with rounded corners. The blue squares are arranged in a slightly offset, layered fashion, creating a sense of depth. The entire graphic is set against a dark blue background that transitions into a lighter blue gradient on the right side.

JS

JAVASCRIPT DATE OBJECT

The JavaScript date object can be used to get year, month and day.

`getDate()`

It returns the integer value between 1 and 31 that represents the day for the specified date on the basis of local time.

`getDay()`

It returns the integer value between 0 and 6 that represents the day of the week on the basis of local time.

`getFullYear()`

It returns the integer value that represents the year on the basis of local time.

`getHours()`

It returns the integer value between 0 and 23 that represents the hours on the basis of local time.

`getMilliseconds()`

It returns the integer value between 0 and 999 that represents the milliseconds on the basis of local time.

JAVASCRIPT DATE OBJECT

The JavaScript date object can be used to get year, month and day.

`getMinutes()`

It returns the integer value between 0 and 59 that represents the minutes on the basis of local time.

`getMonth()`

It returns the integer value between 0 and 11 that represents the month on the basis of local time.

`getSeconds()`

It returns the integer value between 0 and 60 that represents the seconds on the basis of local time.

JAVASCRIPT DATE OBJECT

The JavaScript date object can be used to get year, month and day.

-

`getUTCDate()`

It returns the integer value between 1 and 31 that represents the day for the specified date on the basis of universal time.

`getUTCDay()`

It returns the integer value between 0 and 6 that represents the day of the week on the basis of universal time.

`getUTCFullYear()`

It returns the integer value that represents the year on the basis of universal time.

`getUTCHours()`

It returns the integer value between 0 and 23 that represents the hours on the basis of universal time.

`getUTCMinutes()`

It returns the integer value between 0 and 59 that represents the minutes on the basis of universal time.

`getUTCMonth()`

It returns the integer value between 0 and 11 that represents the month on the basis of universal time.

JAVASCRIPT MATH

The JavaScript math object provides several constants and methods to perform mathematical operation.

`abs()`

It returns the absolute value of the given number.

`ceil()`

It returns a smallest integer value, greater than or equal to the given number.

`floor()`

It returns largest integer value, lower than or equal to the given number.

`max()`

It returns maximum value of the given numbers.

`min()`

It returns minimum value of the given numbers.

`random()`

It returns random number between 0 (inclusive) and 1 (exclusive).

`round()`

It returns closest integer value of the given number.

JAVASCRIPT NUMBER OBJECT

The JavaScript number object enables you to represent a numeric value. It may be integer or floating-point.

`isFinite()`

It determines whether the given value is a finite number.

`isInteger()`

It determines whether the given value is an integer.

`parseFloat()`

It converts the given string into a floating point number.

`parseInt()`

It converts the given string into an integer number.

`toFixed()`

It returns the string that represents a number with exact digits after a decimal point.

`toString()`

It returns the given number in the form of string.

WINDOW OBJECT

The window object represents a window in browser. An object of window is created automatically by the browser.

`alert()`

displays the alert box containing message with ok button.

`confirm()`

displays the confirm dialog box containing message with ok and cancel button.

`prompt()`

displays a dialog box to get input from the user.

`open()`

opens the new window.

`close()`

closes the current window.

`setTimeout()`

performs action after specified time like calling function, evaluating expressions etc.

JAVASCRIPT NAVIGATOR OBJECT

The JavaScript navigator object is used for browser detection. It can be used to get browser information such as appName, appCodeName, userAgent etc.

● ● ● index.html

```
11 document.writeln("<br/>navigator.appCodeName: "+navigator.appCodeName);
12 document.writeln("<br/>navigator.appName: "+navigator.appName);
13 document.writeln("<br/>navigator.appVersion: "+navigator.appVersion);
14 document.writeln("<br/>navigator.cookieEnabled: "+navigator.cookieEnabled);
15 document.writeln("<br/>navigator.language: "+navigator.language);
16 document.writeln("<br/>navigator.userAgent: "+navigator.userAgent);
17 document.writeln("<br/>navigator.platform: "+navigator.platform);
18 document.writeln("<br/>navigator.onLine: "+navigator.onLine);
```


COMMON JAVASCRIPT EVENTS

HTML events are "things" that happen to HTML elements.

`onclick()`

The user clicks an HTML element

`onchange()`

An HTML element has been changed

`onmouseover()`

The user moves the mouse over an HTML element

`onmouseout()`

The user moves the mouse away from an HTML element

`onkeydown()`

The user pushes a keyboard key

`onload()`

The browser has finished loading the page

DOCUMENT OBJECT MODEL-DOM

When a web page is loaded, the browser creates a Document Object Model of the page. With the HTML DOM, JavaScript can access and change all the elements of an HTML document.

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

FINDING HTML ELEMENTS -DOM

●●● index.html

```
4      <p id="MyId"></p>
5      <script>
6      document.getElementById("MyId").innerHTML = "Learn Confidently";
7      </script>
```

●●● index.html

```
4      <p class="MyClass"></p>
5      <script>
6      document.getElementsByClassName("MyClass")[0].innerHTML = "Learn Confidently";
7      </script>
```

FINDING HTML ELEMENTS -DOM

● ● ● index.html

```
4      <p></p>
5      <script>
6      document.getElementsByTagName("p")[0].innerHTML = "Learn Confidently";
7      </script>
```

● ● ● index.html

```
4      <p name="MyName"></p>
5      <script>
6      document.getElementsByName("MyName")[0].innerHTML = "Learn Confidently";
7      </script>
```


HTML DOM DOCUMENT

- Display all name value pairs of cookies in a document
- Display the domain name of the server that loaded the document
- Display the date and time the document was last modified
- Display the title of a document
- Display the full URL of a document
- Replace the content of a document
- Open a new window, and add some content
- Display the number of elements with a specific tag name

HTML DOM DOCUMENT

●●● index.html

```
4  <p id="demo">Click the button to display the cookies associated with this document.</p>
5  <button onclick="myFunction()">Try it</button>
6
7  <script>
8      function myFunction() {
9          document.getElementById("demo").innerHTML =
10             "Cookies associated with this document: " + document.cookie;
11      }
12 </script>
```

HTML DOM DOCUMENT

●●● index.html

```
4   <button onclick="myFunction()">Try it</button>
5   <p id="demo"></p>
6   <script>
7       function myFunction() {
8           document.getElementById("demo").innerHTML = document.domain;
9       }
10  </script>
```

HTML DOM DOCUMENT

●●● index.html

```
4 <p>This document was last modified <span id="demo"></span>.</p>  
5 <script>  
6     document.getElementById("demo").innerHTML = document.lastModified;  
7 </script>
```


HTML DOM DOCUMENT

● ● ● index.html

```
1  <html>
2    <head>
3      <title>Learn Confidently</title>
4    </head>
5  <body>
6
7    <p id="demo"></p>
8    <script>
9      document.getElementById("demo").innerHTML =
10      "The title of this document is: " + document.title;
11    </script>
12
13  </body>
14 </html>
```

HTML DOM DOCUMENT

●●● index.html

```
7    <p>The full URL of this document is: <br><span id="demo"></span>.</p>
8    <script>
9        document.getElementById("demo").innerHTML = document.URL
10    </script>
```

HTML DOM DOCUMENT

●●● index.html

```
7   <p id="demo">Click the button to replace this document with new content.</p>
8   <button onclick="myFunction()">Try it</button>
9   <script>
10      function myFunction() {
11          document.open("text/html","replace");
12          document.write("<h2>Learning about the HTML DOM is fun!</h2>");
13          document.close();
14      }
15  </script>
```

HTML DOM DOCUMENT

●●● index.html

```
7   <p>Click the button to open a new window and add some content.</p>
8   <button onclick="myFunction()">Try it</button>
9
10  <script>
11      function myFunction() {
12          var w = window.open();
13          w.document.open();
14          w.document.write("<h2>Hello World!</h2>");
15          w.document.close();
16      }
17  </script>
```


HTML DOM DOCUMENT

● ● ● index.html

```
7    <p></p>
8    <p></p>
9    <p></p>
10   <p id="demo"></p>
11   <input type="button" onclick="getElements()" value="How many elements tag p?">
12   <script>
13       function getElements() {
14           var x = document.getElementsByTagName("p");
15           document.getElementById("demo").innerHTML = x.length;
16       }
17   </script>
```

HTML DOM FINDING FORM INPUT VALUE

●●● index.html

```
7   <input id="num1"/><br>
8   <input id="num2"/><br>
9   <button onclick="AddTwo()">Add</button>
10  <script>
11      function AddTwo() {
12          var x = document.getElementById("num1").value;
13          var y = document.getElementById("num2").value;
14          var z=parseFloat(x)+parseFloat(y);
15          alert(z);
16      }
17  </script>
```

DOM HTML CSS MANUPULATION

● ● ● index.html

```
7   <h1 id="MyId">Learn Confidently</h1>
8   <button onclick="CSSClassManipulation()">Manipulate</button>
9   <script>
10      function CSSClassManipulation() {
11          var x = document.getElementById("MyId")
12          x.classList.add('text-primary')
13      }
14  </script>
```

DOM HTML CSS MANUPULATION

● ● ● index.html

```
7   <h1 class="text-primary" id="MyId">Learn Confidently</h1>
8   <button onclick="CSSClassManipulation()">Manipulate</button>
9   <script>
10      function CSSClassManipulation() {
11          var x = document.getElementById("MyId")
12          x.classList.remove('text-primary')
13      }
14  </script>
```


DOM CREATE ELEMENT APPEND ELEMENT

●●● index.html

```
7   <ol id="MyList"></ol>
8   <input id="Item"/>
9   <button onclick="AppendElement()">Append</button>
10  <script>
11      function AppendElement() {
12          var Item = document.getElementById("Item").value;
13          var MyList = document.getElementById("MyList");
14          let li = document.createElement("li");
15          li.innerHTML=Item;
16          MyList.appendChild(li)
17      }
18  </script>
```

DOM CHANGING ATTRIBUTE VALUE

●●● index.html

```
7   
8   <button onclick="ChangeSrc()">Change Image Src</button>
9   <script>
10      function ChangeSrc() {
11          var image = document.getElementById("image")
12          image.src="https://cdn.rabbil.com/photos/images/2022/11/04/whyChoose.png"
13
14      }
15  </script>
```

DOM QUERY SELECTOR

●●● index.html

```
7   <h1>H1</h1>
8   <h2 id="MyId">H2</h2>
9   <h3 class="MyClass">H3</h3>
10  <h4 name="MyName">H4</h4>
11  <input placeholder="..." />
12  <button onclick="Change()">Change</button>
13
14  <script>
15      function Change() {
16          document.querySelector("h1").innerHTML='Hello H1'
17          document.querySelector("#MyId").innerHTML='Hello H2'
18          document.querySelector('.MyClass').innerHTML='Hello H3'
19          document.querySelector('h4[name="MyName"]').innerHTML='Hello H4'
20          document.querySelector('input').placeholder='New Placeholder'
21      }
22  </script>
```

AJAX

AJAX is a developer's dream, because you can:

- Update a web page without reloading the page
- Request data from a server - after the page has loaded
- Receive data from a server - after the page has loaded
- Send data to a server - in the background

AJAX PACKAGES

- The XMLHttpRequest object build in with javascript
- Fetch API supported by modern browser's
- Axios package
- JQuery Ajax
- Superagent

AJAX FETCH API GET REQUEST

●●● index.html

```
8   <button onclick="FetchGetData()">Fetch Get Data</button>
9
10  <script>
11      function FetchGetData() {
12
13          var url="https://crud.teamrabbil.com/api/v1/ReadProduct"
14          var requestOptions = {method: 'GET'};
15
16          fetch(url, requestOptions)
17              .then(response => response.json())
18              .then(result => console.log(result))
19              .catch(error => console.log('error', error));
20
21      }
22  </script>
```

AJAX FETCH API POST REQUEST

index.html

```
8   <button onclick="FetchPostData()">Fetch Post Data</button>
9
10  <script>
11      function FetchPostData() {
12
13          var url="https://crud.teamrabbil.com/api/v1/CreateProduct"
14
15          var data={ Img:"A",ProductCode:"B",ProductName:"C",Qty:"D",TotalPrice:"E",UnitPrice:"F"}
16          var requestOptions = {
17              method: 'POST',
18              headers: {'Accept': 'application/json','Content-Type': 'application/json'},
19              body: JSON.stringify(data)
20          };
21
22          fetch(url, requestOptions)
23              .then(response => response.json())
24              .then(result => console.log(result))
25              .catch(error => console.log('error', error));
26      }
27  </script>
```

