JavaScript (Part 1)

Dr R Rakesh
Professor

History of JavaScript

History

- Brendan Eich at Netscape, 1995
- Need for interactivity in web pages
- Mocha, LiveWire, LiveScript, then JavaScript
- Not taken seriously at first
  - Called "beginner's language" or "toy"

Brendan Eich
CTO, Mozilla Corporation

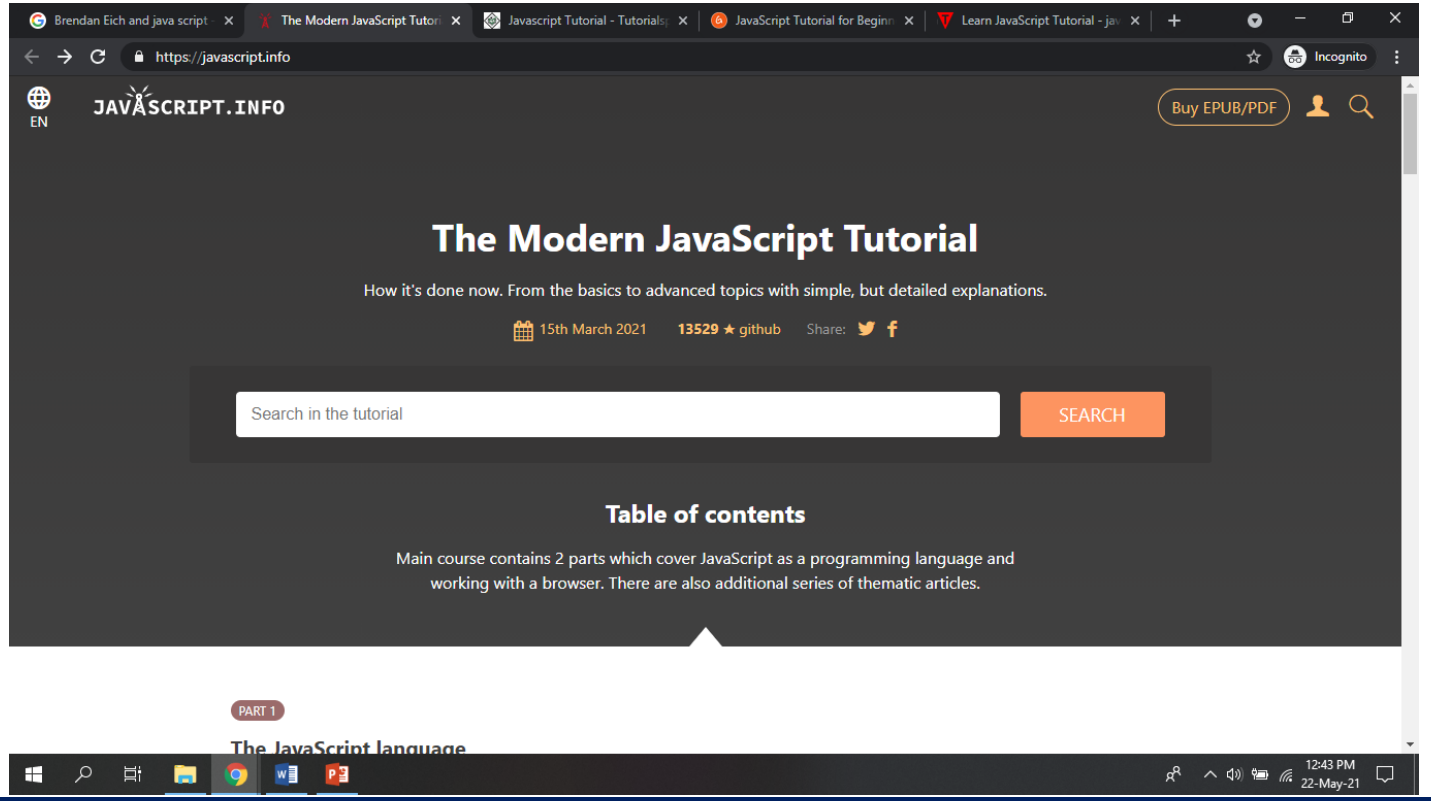
Netscape and Sun Microsystems decided to standardize the language along with the help of the European Computer Manufacturers Association, who would host the standard.
This standardized language was called ECMAScript

- Microsoft reverse-engineered JavaScript to create their own version, called JScript to avoid copyright issues with Sun Microsystems who owned the Java trademark and had licensed it to Netscape.

Links to learn JavaScript

- https://www.javatpoint.com/javascript-tutorial

https://www.guru99.com/interactive-javascript-tutorials.html



JavaScript Intro

- **JavaScript** is a lightweight, interpreted **programming** language.
- It is designed for creating network-centric applications.
- It is complimentary to and integrated with Java.
- **JavaScript** is very easy to implement because it is integrated with HTML.
- It is open and cross-platform.

- JavaScript usage has now extended to mobile app development, desktop app development, and game development.
- It helps you developing great front-end as well as back-end softwares using different Javascript based frameworks like jQuery, Node.JS etc.

Where to Script ? <script>

- In HTML, JavaScript code is inserted between <script> and </script> tags.
- JavaScript Functions and Events
- A JavaScript function is a block of JavaScript code, that can be executed when "called" for.

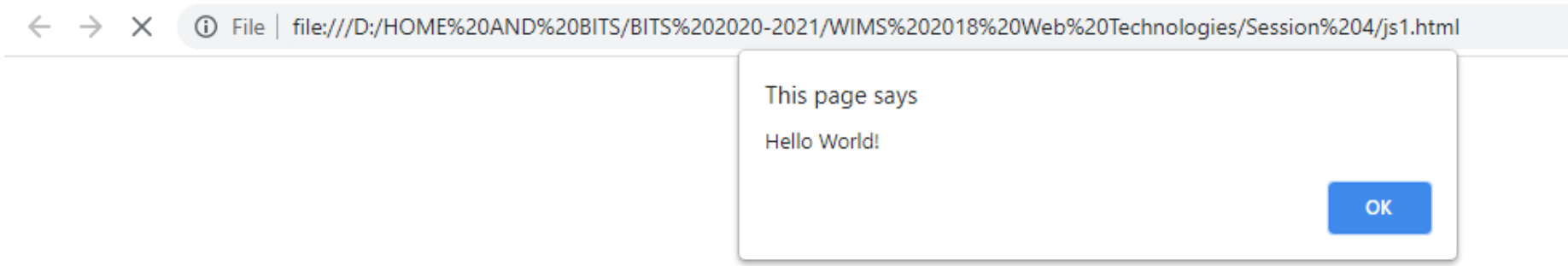
- For example, a function can be called when an event occurs, like when the user clicks a button.

Hello World using Javascript

```
<html>
<head>
        <title>My First JavaScript code!!!</title>
        <script type="text/javascript">

                alert("Hello World!");

        </script>
</head>
<body>
</body>
</html>
```



- **Note:** type="text/javascript" is not necessary in HTML5.

JAVA SCRIPT – methods to call the <script>

JavaScript in <head>

- You can place any number of scripts in an HTML document.
- Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

JavaScript Output

JavaScript Output

- JavaScript can "display" data in different ways:

  - Writing into an HTML element, using innerHTML.
  - Writing into the HTML output using document.write().
  - Writing into an alert box, using window.alert().
  - Writing into the browser console, using console.log().

Using innerHTML

- To access an HTML element, JavaScript can use the document.getElementById(id) method.

- The id attribute defines the HTML element. The innerHTML property defines the HTML content:

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

<h1>My First Web Page</h1>
<p>My First Paragraph</p>

<p id=“Regenesys"

></p> <script>
document.getElementById(“Regenesys").innerHTML = 35 +
36; </script>

</body>
</html>
```

Using document.write()

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

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

Using window.alert()

- You can skip the window keyword.

- In JavaScript, the window object is the global scope object, that means that variables, properties, and methods by default belong to the window object.
- This also means that specifying the window keyword is optional:

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

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

JavaScript Variable

JavaScript Variable: Declare, Assign a Value with Example

- Variables are used to store values (name = "John") or expressions (sum = x + y).

Declare Variables in JavaScript

- Before using a variable, you first need to declare it.
- You have to use the keyword var to declare a variable like this:
  - var name;
- Assign a Value to the Variable
- You can assign a value to the variable either while declaring the variable or after declaring the variable
  - var name = "John";

    or
  - var name;
  - name = "John";

JavaScript Variable: Declare, Assign a Value with Example

```
<html>
<head>
<title>Variables!!!</title>
<script type="text/javascript">
var one = 22;
var two = 3;
var add = one + two;
var minus = one - two;
var multiply = one * two;
var divide = one/two;
        document.write("First  No: = " + one + "<br />Second  No: = " + two + " <br />");
        document.write(one  + " + " + two + " = " + add + "<br/>");
        document.write(one  + " - " + two + " = " + minus + "<br/>");
        document.write(one  + " * " + two + " = " + multiply + "<br/>");
        document.write(one  + " / " + two + " = " + divide + "<br/>");
</script>
</head>
<body>
</body>
</html>
```

First No: = 22
Second No: = 3
22 + 3 = 25
22 - 3 = 19
22 * 3 = 66
22 / 3 = 7.333333333333333

Creating an Array

- Using an array literal is the easiest way to create a JavaScript Array.
- Syntax:
- const *array_name* = [*item1*, *item2*, ...];
- It is a common practice to declare arrays with the const keyword.
- ECMAScript 2015 (ES6)
- In 2015, JavaScript introduced an important new keyword: const.
- It has become a common practice to declare arrays using const:

Cannot be Reassigned

- An array declared with const cannot be reassigned:
- const cars = ["Saab", "Volvo", "BMW"];
- cars = ["Toyota", "Volvo", "Audi"]; // ERROR
  - Arrays are Not Constants
  - The keyword const is a little misleading.
  - It does NOT define a constant array. It defines a constant reference to an array.

Different methods to declare const array

```
const cars = [
  "Saab",
  "Volvo",
  "BMW"
];
```

```
const cars = [];
cars[0]= "Saab";
cars[1]= "Volvo";
cars[2]= "BMW";
```

```
const cars = new Array("Saab", "Volvo", "BMW");
```

Accessing Array Elements

- You access an array element by referring to the **index number**:

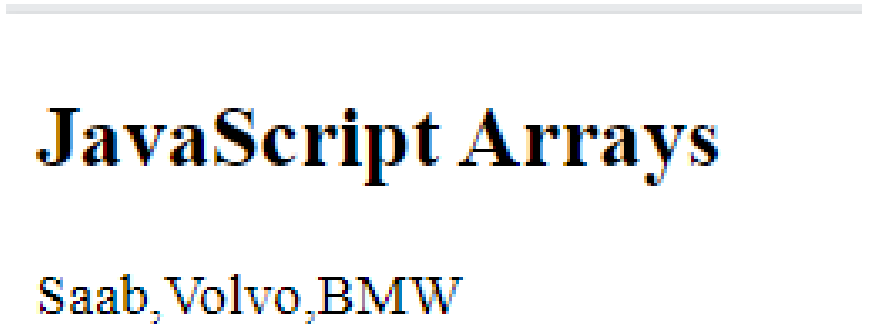
```
<script>
const cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars[0];
</script>
```

Note: Array indexes start with 0.

[0] is the first element. [1] is the second element.

Access the Full Array

- An array is a special variable, which can hold more than one value:
- <!DOCTYPE html>
- <html>
- <body>
- <h2>JavaScript Arrays</h2>
-
- <script>
- const cars = ["Saab", "Volvo", "BMW"];
- document.getElementById("demo").innerHTML = cars;
- </script>
- </body>
- </html>



JavaScript Let

- The let keyword was introduced in ES6 (2015).
- Variables defined with let cannot be Redeclared.
- Variables defined with let must be Declared before use.
- Variables defined with let have Block Scope.

Example

```
let x = "John Doe";

let x = 0;

// SyntaxError: 'x' has already been declared
```

With `var` you can:

Example

```
var x = "John Doe";

var x = 0;
```

If else in JavaScript

```
<html> <body>
    <script>
        var age = 15;
         if( age > 18 )
        {
          document.write("<b>Qualifies for driving</b>"); }
          else
          {
          document.write("<b>Does not qualify for driving</b>");
        }
         </script>
    <p>Set the variable to different value and then try...</p> </body>
</html>
```