

MODULE – 03

JavaScript

BASIC CONCEPT OF JAVASCRIPT

- Web pages made using only HTML are somewhat static with no interactivity and negligible user involvement.
- HTML tags are just instructions on document and the display of the document is dependent on the browser.
- Interactive pages cannot be built with only HTML, we need a programming language. So Netscape came out with a client-side language called as JavaScript.

WHAT IS JAVASCRIPT?

- JavaScript is the scripting language of the Web.
- JavaScript is used in Web pages to add functionality, validate forms, detect browsers, and much more.
- JavaScript is the most popular scripting language on the internet and works on all major browsers available like IE, Firefox, Chrome etc..

WHAT IS NEED OF JAVASCRIPT?

- For putting dynamic content into an HTML Page.
- For client side Validation.
- For storing and retrieving client's information in the form of Cookies.

HOW DOES JAVASCRIPT WORKS?

- When a JavaScript is inserted into an HTML document, the Internet browser will read the HTML and interpret the JavaScript.
- The JavaScript can be executed immediately, or at a later event.

EMBEDDING JS IN HTML

- In HTML you can use script tag for embedding javascript.
 - <SCRIPT>.....</SCRIPT>
- Attributes for script tag
 - src :- Specifies the url of external script file
 - type :- Specifies the media type of the script
 - async, reffererpolicy & many more...
- Use the <NOSCRIPT> tag to specify alternate content for browsers that do not support JavaScript.

WHERE TO WRITE JS?

- Head Section
 - If you want the Scripts to be executed when they are called, or whenever a user action happens, put script tag in the head section.
- Body Section
 - If you want the Scripts to be executed when the page loads then put script tag in the body section
- External File
 - If you want to run the same JavaScript on several pages, without having to write the same script on every page, you can write a JavaScript in an external file.

WHERE TO WRITE JS?

- Head Section
 - If you want the Scripts to be executed when they are called, or whenever a user action happens, put script tag in the head section.
- Body Section
 - If you want the Scripts to be executed when the page loads then put script tag in the body section
- External File
 - If you want to run the same JavaScript on several pages, without having to write the same script on every page, you can write a JavaScript in an external file.

HELLO WORLD USING JS

- Let's implement the hello world program using JavaScript.

STATEMENT, SEMICOLON & COMMENTS

- Statements are syntax constructs and commands that perform actions.
 - `alert('Hello to the'); alert('JS World');`
- A semicolon may be omitted in most cases when a line break exists.
`alert("Hello")`
`alert("World")`
- Comments
 - `//Single Line Comment`
 - `/* Multi Line Comment */`

VARIABLES IN JS

- Why Variables?
- A variable is a “named storage” for data.
- A variable should be declared only once, repeated declaration cause an error (Exception :- var).
- Variables are case-sensitive.
- Variable name can't be a reserved keyword. Eg:- let, break, if, function etc.
- There are two limitations on variable names in JavaScript:
 - The name must contain only letters, digits, or the symbols \$ and _.
 - The first character must not be a digit.
- Types of Variable
 - let
 - var
 - const

LET KEYWORD

- Variable Declaration
 - `let name;`
- Variable Assignment
 - `name = "Neelkanth";`
- Variable Declaration with Assignment
 - `let surname = "Patel";`
- Multiple Variables
 - `let name = "Neelkanth", surname="Patel";`

VAR KEYWORD

- The var keyword is used in all JavaScript code from 1995 to 2015.
- There are subtle differences between let and var, we'll cover them in detail later.
- Variable Declaration
 - `var name;`
- Variable Assignment
 - `var = "Neelkanth";`
- Both Declaration & Assignment
 - `var age = 27`

CONST KEYWORD

- To declare a constant (unchanging) variable, use `const` instead of `let`.
 - `const myBirthday = '18.04.1995';`
- There is a widespread practice to use constants as aliases for difficult-to-remember values that are known prior to execution.
- Such constants are named using capital letters and underscores.

DATATYPES

- A value in JavaScript is always of a certain type. For example, a string or a number.
- There are eight basic data types in JavaScript.
- We can put any type in a variable.
 - For example, a variable can at one moment be a string and then store a number:

NUMBER

- The number type represents both integer and floating point numbers.
- There are many operations for numbers, e.g. multiplication *, division /, addition +, subtraction -, etc.
- Besides regular numbers, there are so-called “special numeric values” which also belong to this data type: Infinity, -Infinity and NaN.

BIGINT

- In JavaScript, the “number” type cannot safely represent integer values larger than $(2^{53}-1)$ (that’s 9007199254740991), or less than $-(2^{53}-1)$ for negatives.
- BigInt type was recently added to the language to represent integers of arbitrary length.
- A BigInt value is created by appending n to the end of an integer:

STRING

- A string in JavaScript must be surrounded by quotes.
- In JavaScript, there are 3 types of quotes.
 - Double quotes: "Hello".
 - Single quotes: 'Hello'.
 - Backticks: `Hello`.
- Double and single quotes are “simple” quotes. There’s practically no difference between them in JavaScript.
- Backticks are “extended functionality” quotes. They allow you to embed variables and expressions into a string by wrapping them in \${...}.

BOOLEAN

- The boolean type has only two values: true and false.
- This type is commonly used to store yes/no values: true means “yes, correct”, and false means “no, incorrect”.

NULL

- The special null value does not belong to any of the types described above.
- It forms a separate type of its own which contains only the null value.
- It's just a special value which represents “nothing”, “empty” or “value unknown”.

UNDEFINED

- The special value `undefined` also stands apart. It makes a type of its own, just like `null`.
- The meaning of `undefined` is “value is not assigned”.
- If a variable is declared, but not assigned, then its value is `undefined`:

OBJECTS

- The object type is special.
- All other types are called “primitive” because their values can contain only a single thing (be it a string or a number or whatever).
- In contrast, objects are used to store collections of data and more complex entities.

TYPEOF

- The `typeof` operator returns the type of the argument. It's useful when we want to process values of different types differently or just want to do a quick check.
- A call to `typeof x` returns a string with the type name: