JAVASCRIPT BASICS

1. Variables

* **Variables** are containers that hold data values. They are used to store, manipulate, and display information within a program.
* In short, a variable is like a memory unit that we can access by typing the name of the variable.
* Each variable has a unique **name**and a **value**that can be of different types.
* Variables are containers that store information for later use.
* We can assign a value to a variable with the assignment operator =.
* cannot declare a const-variable without assigning a value to it directly.
* It is considered good practice to use const for variables whose value will never change.

1. Numbers:
   * JavaScript has a number type that can represent both integers and floating-point numbers.
2. String:
   * A **char** is a single character (For example: 1, 6, %, b, p, ., T, etc.)
   * The **string** type is a special type that consists of multiple **char**s.
   * To initialize a string value in a variable, enclose it within single or double quotation marks:
3. Boolean:

* A **bool** (Boolean) type has only 2 possible values: true or false.
* Booleans are building blocks for logic in the programs we write.

1. Naming conventions:

* In JavaScript, there are certain rules and conventions for naming variables:
* Variable names are case-sensitive, it means MyVariable & Myvariable are considered different.
* Variable names can only contain letters(a-z, A-Z), numbers(0-9), underscores(\_), and dollar signs($).
* Variable name must starts with a letters, underscore or dollar sign, they can’t begin with a number.
* Certain words are reserved by javascript and can’t be used as variable names such as **let, const, function, if, else etc**..
* It’s a common practice to use camelCase for variable names, where each word except the 1st starts with a capital letter(e.g., **m**y**V**ariable**N**ame).

1. Empty Variables:

* In JavaScript, it's possible to declare variables without assigning them a specific value. This can be useful when you know you'll use a variable later in your code, but you don't yet know its initial value.
* To declare an empty variable in JavaScript, you can use the let keyword followed by the variable name, without assigning any value
* creates a variable named myVariable but doesn't assign any value to it. The variable is said to be **uninitialized.**

1. Constants:

* In JavaScript, sometimes you need variables that *never change*. This is where const comes in! It’s used to create constants—values that stay the same after being set.
* Use the const keyword, just like let, but once you set its value, you **cannot change it** later.

1. Concept of truthy & falsy value in Javascript and how it relates to javascript Booleans:

* Truthy values:
  + A value is considered "truthy" if it evaluates to true when coerced to a Boolean. Most values in JavaScript are truthy, including:
    - Non-zero numbers (e.g. `1`, `-1`, `3.14`).
    - Non-empty strings (e.g.`”hello”`, ””`).
    - Objects (e.g. `{ }`, `[ ]`).
    - Functions.
* Falsy values:
* A value is considered "falsy" if it evaluates to false when coerced to a Boolean. The following values are falsy in JavaScript:
  + `false`.
  + `0` (zero).
  + `” ”`(empty string).
  + `null`.
  + `undefined`.
  + `NaN` (Not-a-Number).

1. Operators: part-1
2. Arithmetic operators: ~it performs mathematical calculations.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Operation** | **Example** |
| **+** | Addition | 3 + 2 = 5 |
| **-** | Subtraction | 3 – 2 =1 |
| **\*** | Multiplication | 3 \* 2 = 6 |
| **/** | Division | 4 / 2 = 2 |

* Arithmetic shortcut Assignment operators:
* Javascript created a shortcut for self-arithmetic operations.
* This operation is valid for all arithmetic operations:

|  |  |
| --- | --- |
| **Operator** | **Shortcut** |
| **+** | **+=** |
| **-** | **-=** |
| **\*** | **\*=** |
| **/** | **/=** |
| **%** | **%=** |

1. Modulo Operator:

* The Modulo operator % tells you what’s left over after dividing one number by another. Result = dividend % divisor.
* **Dividend**: The number being divided.
* **Divisor**: The number that divides the dividend.
* **Result**: The reminder of the division

1. Comparison operator:

* Comparison operators are used to compare two operands.
* Sometimes we need to check whether an operand is bigger/smaller/... than another operand. The following table shows possible operators for comparison:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Meaning** | **Example** |
| **==** | Equal | 1 == 2 returns **false** |
| **!=** | Not Equal | 1 != 2 returns **true** |
| **>** | Greater Than | 1 > 2 returns **false** |
| **<** | Lower Than | 1 < 2 returns **true** |
| **>=** | Greater or Equal | 1 >= 2 returns **false** |
| **<=** | Lower or Equal | 1 <= 2 returns **true** |