JavaScript

JavaScript is a **programming language** used primarily to add interactivity to websites. It runs on the browser, enabling dynamic changes to HTML and CSS and interaction with server data. JavaScript can validate form inputs, create animations, manipulate content, and more. It's also essential in frameworks like React, Angular, and Node.js.

---

2. Alerts

An **alert** is a method used to display messages in a popup box on the browser. It's useful for notifying users or showing messages temporarily.

```
javascript
Copy code
alert("This is an alert!");
```

---

3. Data Types

JavaScript has various **data types** to represent different kinds of values:

- **Number**: Represents numeric values (e.g., `42`, `3.14`).
- **String**: Represents text values, enclosed in single (`' '`), double (`" "`), or backticks (`` ` ` ``).
- **Boolean**: Represents true or false values.
- **Object**: Complex data type to store collections of values (e.g., `{ name: 'Alice', age: 25 }`).
- **Array**: A type of object for lists (e.g., `[1, 2, 3]`).
- **Undefined**: Represents variables that are declared but not assigned.
- **Null**: An intentional empty value.
- **Symbol**: Represents unique, immutable values (often used as object keys).
- **BigInt**: Represents large integers that exceed the range of the Number type.

---

4. Variables

Variables are used to store data. In JavaScript, you declare variables using `let`, `const`, or the older `var`.

- **let**: Declares variables that can be reassigned.
- **const**: Declares variables with values that cannot be reassigned.
- **var**: An older way to declare variables (less commonly used).

```
javascript
```

```
Copy code
let name = "Alice";
const age = 25;
var city = "New York";
```

---

5. String Concatenation

String concatenation is combining two or more strings into one.

- Using the + operator:

  ```
  javascript
  Copy code
  let greeting = "Hello, " + "world!";
  ```

- Using **Template Literals** (backticks):

  ```
  javascript
  Copy code
  let name = "Alice";
  let greeting = `Hello, ${name}!`;
  ```

---

6. Slicing and Extracting Parts of Strings

- **slice()**: Extracts part of a string and returns a new string.

  ```
  javascript
  Copy code
  let str = "Hello, world!";
  let slicedStr = str.slice(7, 12); // "world"
  ```

- **substring()**: Similar to `slice()`, but doesn't accept negative indexes.
- **substr()**: Extracts a part of a string, specifying the start and the length.

  ```
  javascript
  Copy code
  let str = "JavaScript";
  let subStr = str.substr(4, 6); // "Script"
  ```

---

7. String Length

The `.length` property returns the number of characters in a string.

```
javascript
Copy code
let text = "Hello!";
console.log(text.length); // 6
```

---

8. Changing Case in Text

To change the case of text in JavaScript:

- **toUpperCase()**: Converts all characters to uppercase.

  ```
  javascript
  Copy code
  let text = "hello";
  let upperText = text.toUpperCase(); // "HELLO"
  ```

- **toLowerCase()**: Converts all characters to lowercase.

  ```
  javascript
  Copy code
  let text = "HELLO";
  let lowerText = text.toLowerCase(); // "hello"
  ```

---

9. Arithmetic and Modulo Operators

JavaScript **arithmetic operators** allow you to perform basic math operations.

- `+` (Addition)
- `-` (Subtraction)
- `*` (Multiplication)
- `/` (Division)
- `%` (Modulo): Returns the remainder of a division.

```
javascript
Copy code
let x = 10;
let y = 3;
console.log(x % y); // 1 (remainder of 10 / 3)
```

---

10. Increment and Decrement Expressions

These expressions are shorthand ways to increase or decrease a variable by 1.

- `++`: Increment operator, increases by 1.
- `--`: Decrement operator, decreases by 1.

```
javascript
Copy code
let count = 5;
count++; // Now count is 6
```

```
count--; // Now count is 5 again
```

---

11. Parameters and Arguments

- **Parameters**: Variables defined in the function declaration, acting as placeholders.

  ```
  javascript
  Copy code
  function greet(name) {
    console.log("Hello, " + name);
  }
  ```

- **Arguments**: Values passed to the function when it is called.

  ```
  javascript
  Copy code
  greet("Alice"); // Here "Alice" is the argument
  ```

---

12. Output and Return Values

- **Output**: Displaying data to the user, typically through methods like `console.log()`.
- **Return Values**: The result of a function is returned to where the function was called. `return` stops the function execution and sends a value back.

```
javascript
Copy code
function add(a, b) {
    return a + b; // Returns the sum
}
let result = add(5, 3); // result is 8
```

In summary, these JavaScript concepts cover the basics of data manipulation, variable management, arithmetic, and working with functions and user input/output. Together, they form the foundation for creating interactive and dynamic web applications.