ChatGPT - JS Basics and Operators

JavaScript Reference Notes

Variable Declaration

1. `var` (Global Scope):

```javascript
var a = 1;
console.log(a); // Output: 1
```

   ⚠ Avoid using `var` to prevent scope-related logic errors.

2. `const` and `let` (Block Scope):

```javascript
const b = 2;
let c = 3;
console.log(b, c); // Output: 2 3
```

Arithmetic Operators

```javascript
var num1 = 10;
var num2 = 5;
console.log(num1 + num2); // Output: 15
console.log(num1 - num2); // Output: 5
console.log(num1 * num2); // Output: 50
console.log(num1 / num2); // Output: 2
console.log(num1 % num2); // Output: 0
```

Assignment Operators

javascript

```javascript
var num1 = 10;
console.log(num1 += 5); // Output: 15
console.log(num1 -= 5); // Output: 10
console.log(num1 *= 2); // Output: 20
console.log(num1 /= 2); // Output: 10
```

Comparison Operators

javascript

```javascript
console.log(10 == 5); // Output: false
console.log(10 === 5); // Output: false
console.log(10 != 5); // Output: true
console.log(10 !== 5); // Output: true
```

Logical Operators

javascript

```javascript
console.log(true && true); // Output: true
console.log(true || false); // Output: true
console.log(!true); // Output: false
```

Data Types

- **Number:**

  javascript

  ```javascript
  var num = 10;
  console.log(num); // Output: 10
  ```

- **String**:

```javascript
var str = "Hello, World!";
console.log(str); // Output: Hello, World!
```

- **Boolean**:

```javascript
var bool = true;
console.log(bool); // Output: true
```

- **Null**:

```javascript
var nullVar = null;
console.log(nullVar); // Output: null
```

- **Undefined**:

```javascript
var undefinedVar;
console.log(undefinedVar); // Output: undefined
```

- **Type Check (`typeof`)**:

```javascript
console.log(typeof undefined); // Output: undefined
console.log(typeof null); // Output: object
```

Data Structures

- **Array**:

```javascript
```

```
var colors = ["red", "green", "blue"];
console.log(colors); // Output: ['red', 'green', 'blue']
```

- **Object**:

```javascript
var person = { name: "John Doe", age: 30 };
console.log(person); // Output: { name: 'John Doe', age: 30 }
```

- **Set** (Unique values):

```javascript
var set = new Set([1, 2, 3]);
console.log(set); // Output: Set { 1, 2, 3 }
```

- **Map** (Key-value pairs):

```javascript
var map = new Map([["key1", "value1"], ["key2", "value2"]]);
console.log(map); // Output: Map { 'key1' => 'value1', 'key2' => 'value2' }
```

Control Structures

- **If-else**:

```javascript
if (true) {
    console.log('Condition is true'); // Output: Condition is true
}
```

- **Switch Statement**:

```javascript
switch (1) {
    case 0: console.log('Value is 0');
```

```
    case 1: console.log('Value is 1');
    default: console.log('Default case');
}
// Output: Value is 1, Default case
```

Loops

1. **For loop:**

```javascript
for (var i = 0; i < 3; i++) {
    console.log(i); // Output: 0, 1, 2
}
```

2. **While loop:**

```javascript
var i = 0;
while (i < 3) {
    console.log(i); // Output: 0, 1, 2
    i++;
}
```

3. **Do-while loop:**

```javascript
var i = 0;
do {
    console.log(i); // Output: 0, 1, 2
    i++;
} while (i < 3);
```

Functions

1. **Declaration:**

```javascript
function add(a, b) {
    return a + b;
}
console.log(add(1, 2)); // Output: 3
```

2. **Expression:**

```javascript
var add = function(a, b) {
    return a + b;
};
console.log(add(1, 2)); // Output: 3
```

3. **Arrow Function:**

```javascript
var add = (a, b) => a + b;
console.log(add(1, 2)); // Output: 3
```

Exception Handling

```javascript
try {
    var x = 5 / 0;
    console.log(x); // Output: Infinity
} catch (error) {
    console.log('Error:', error.message);
}
```

Module Imports

```javascript
import { add, multiply } from './math.js';
console.log(add(5, 10)); // Output: 15
console.log(multiply(5, 10)); // Output: 50
```

Console vs Document

- `console.log` : Outputs to the browser's developer console.
- `document` : Used for DOM manipulation to control webpage content.