

# JavaScript-Notes

## \*\*JavaScript Fundamentals\*\*

### 1. Variables:

Variables in JavaScript are containers used to store data values. They allow you to label and store data for later use and manipulation. Variables can hold various types of data like numbers, text, arrays, objects, etc.

#### Types of Variable Declarations

- **let**: Used for variables that can be reassigned. It is block-scoped, meaning it is only accessible within the block where it is defined.
- **const**: Used for variables that will not be reassigned. It is also block-scoped.
- **var**: The older way to declare variables. It is function-scoped, not recommended for modern JavaScript because of scope issues.

#### Examples

##### 1. Using **let**:

```
let age = 25;  
console.log(age); // Output: 25
```

```
age = 30;  
console.log(age); // Output: 30
```

##### 2. Using **const**:

```
const birthYear = 1995;  
console.log(birthYear); // Output: 1995
```

```
// birthYear = 2000; // This will cause an error as `const` cannot be reassigned
```

##### 3. Using **var**:

```
var city = "New York";  
  
console.log(city); // Output: New York  
  
city = "Los Angeles";  
  
console.log(city); // Output: Los Angeles
```

## 2. Data Types

Data types specify the kind of value a variable holds. JavaScript has several built-in data types, including primitive types (simple values) and non-primitive types (complex structures).

### Primitive Data Types

1. **Number**: Represents numerical values.
2. **String**: Represents text enclosed in single ( ' '), double ( " "), or backticks ( ` ` ).
3. **Boolean**: Represents a logical entity and can have only two values: **true** or **false**.
4. **Undefined**: When a variable is declared but not assigned a value.
5. **Null**: Represents an empty or unknown value.
6. **Symbol**: Used for creating unique identifiers.
7. **BigInt**: Represents large integers.

### Non-Primitive Data Types

1. **Object**: Used to store collections of data and complex entities.
2. **Array**: A type of object that stores a list of values.

### Examples

- **Number, String, and Boolean**

```
let age = 30; // Number

let name = "Alice"; // String

let isStudent = true; // Boolean

console.log(age); // Output: 30

console.log(name); // Output: Alice

console.log(isStudent); // Output: true
```

- **Undefined and Null**

```
let city;

console.log(city); // Output: undefined

let result = null;

console.log(result); // Output: null
```

- **Array**

```
let colors = ["red", "green", "blue"];  
  
console.log(colors); // Output: ["red", "green", "blue"]  
  
console.log(colors[1]); // Output: green (accessing array element by index)
```

Adding and Removing Elements in Array:

**push**: Adds an element to the end of the array.

```
let fruits = ["apple", "banana"];  
  
fruits.push("grape");  
  
console.log(fruits); // ["apple", "banana", "grape"]
```

**pop**: Removes the last element from the array.

```
let fruits = ["apple", "banana"];  
  
fruits.pop();  
  
console.log(fruits); // ["apple"]
```

**unshift**: Adds an element to the beginning of the array.

```
let fruits = ["apple", "banana"];  
  
fruits.unshift("grape");  
  
console.log(fruits); // ["grape", "apple", "banana"]
```

**shift**: Removes the first element.

```
let fruits = ["apple", "banana"];  
  
fruits.shift();  
  
console.log(fruits); // ["banana"]
```

- **Object**

```
let person = {  
  name: "Alice",  
  age: 30,  
  city: "Paris"  
};  
  
console.log(person); // Output: { name: "Alice", age: 30, city: "Paris" }  
  
console.log(person.name); // Output: Alice
```

### 3. Operators

Operators are symbols or keywords used to perform operations on variables and values. They allow you to manipulate data, perform calculations, and make decisions.

#### Types of Operators

1. **Arithmetic Operators:** Used for mathematical calculations.
2. **Comparison Operators:** Used to compare two values.
3. **Logical Operators:** Used to combine multiple conditions.

#### Arithmetic Operators:

Operator	Description	Example	Result
+	Addition	5 + 5	10
-	Subtraction	5 - 5	0
/	Division	20/5	4
%	Remainder	9 % 4	1
*	Multiplication	2 * 2	4
**	Exponentiation	3**2	9

#### Example:

Division 20/5 4

Exponentiation 3\*\*2 9

```

let x = 10;
let y = 5;
console.log(x + y); // Output: 15
console.log(x - y); // Output: 5
console.log(x * y); // Output: 50
console.log(x / y); // Output: 2
console.log(x % y); // Output: 0
console.log(x ** 2); // Output: 100 (10 squared)

```

## Comparison Operators:

Operator	Description	Example	Result
==	Equal to	5 == '5'	true
===	Strict Equal to	5 === '5'	false
!=	Not equal to	5 != '5'	false
!==	Strict not equal to	5 !== '5'	true
>	Greater than	10 > 5	true
<	Less than	3 < 5	true
>=	Greater than or equal to	7 >= 7	true
<=	Less than or equal to	5 <= 8	true

## Example

```

let a = 10;
let b = 20;

console.log(a == b); // Output: false
console.log(a === 10); // Output: true
console.log(a != b); // Output: true
console.log(a > b); // Output: false
console.log(a <= b); // Output: true

```

## Logical Operators:

Operator	Description	Example	Result
&&	Logical AND	True && False	false
	Logical OR	True    False	true
!	Logical NOT	!true	false

## Example

```
let isAdult = true;
```

```
let hasPermission = false;
```

```
console.log(isAdult && hasPermission); // Output: false
```

```
console.log(isAdult || hasPermission); // Output: true
```

```
console.log(!isAdult); // Output: false
```