📘 JavaScript Fundamentals – Master Notes

✅ 1. Output in JavaScript

Methods:

```
console.log("Hello, World!");        // Standard output with newline
process.stdout.write("No newline");   // Writes without a newline
console.table({ name: "Swagat", age: 22 }); // Outputs data in table format
console.warn("This is a warning");    // Shows a warning message
console.error("This is an error");    // Shows an error message
```

✅ 2. JavaScript Data Types

🔹 Primitive Data Types:

- **String** – Textual data
  ```
  let name = "Swagat";
  ```
- **Number** – All numbers (integers and floats)
  ```
  let age = 22;
  ```
- **Boolean** – `true` or `false`
  ```
  let isLoggedIn = true;
  ```
- **Undefined** – Variable declared but not assigned
  ```
  let score;
  ```
- **Null** – Intentional absence of any value
  ```
  let user = null;
  ```
- **Symbol** – Unique identifiers
  ```
  let sym1 = Symbol("id");
  let sym2 = Symbol("id");
  console.log(sym1 === sym2); // false
  ```

- **BigInt** – Large integers
  let big = 1234567890123456789012345678901234567890n;

🔸 **Reference (Non-Primitive) Types:**

- **Object**
  let user = { name: "Swagat", age: 22 };

- **Array**
  let fruits = ["apple", "banana"];

- **Function**
  function greet() {
  return "Hello!";
  }

✅ 3. Variables and Constants

```
var a = 10;    // function-scoped
let b = 20;    // block-scoped
const c = 30;  // block-scoped, cannot be reassigned
```

🧠 Best Practice:

- Use `let` and `const` only.

- Prefer `const` unless you need to reassign.

✅ 4. Arithmetic Operators

```
let a = 10, b = 3;

console.log(a + b); // Addition
console.log(a - b); // Subtraction
console.log(a * b); // Multiplication
console.log(a / b); // Division
console.log(a % b); // Remainder
console.log(a ** b); // Exponentiation (10^3 = 1000)
```

✅ 5. Assignment Operators

```
let x = 10;

x += 5; // x = x + 5
x -= 2; // x = x - 2
x *= 3; // x = x * 3
x /= 4; // x = x / 4
x %= 3; // x = x % 3
x **= 2; // x = x ** 2
```

✅ 6. Comparison Operators

```
let x = 5, y = "5";

console.log(x == y);   // true (loose equality)
console.log(x === y);  // false (strict equality)
console.log(x != y);   // false
console.log(x !== y);  // true
console.log(x > 3);    // true
console.log(x <= 5);   // true
```

🧠 Tip: Always use `===` and `!==` to avoid unexpected type coercion.

✅ 7. Logical Operators

```
let isAdmin = true;
let hasPaid = false;

console.log(isAdmin && hasPaid); // false (AND)
console.log(isAdmin || hasPaid); // true (OR)
console.log(!isAdmin);          // false (NOT)
```

✅ Use case:

```
if (isAdmin && hasPaid) {
  console.log("Access granted");
}
```

✅ 8. Operator Precedence

```
let result = 2 + 3 * 4;        // 2 + (3*4) = 14
let result2 = (2 + 3) * 4;     // (2+3) * 4 = 20
```

🧠 Use parentheses to ensure correct order of evaluation.

✅ 9. Increment & Decrement

```
let a = 5;

console.log(a++); // 5 (post-increment, returns value then increments)
console.log(++a); // 7 (pre-increment, increments first then returns)
```

Same goes for decrement (`--a`, `a--`).

✅ 10. Mutability: Primitive vs Reference

```
let x = 100;
let y = x;
y = 200;
console.log(x); // 100 – primitives are copied by value

let obj1 = { name: "Swagat" };
let obj2 = obj1;
obj2.name = "Nanda";
console.log(obj1.name); // "Nanda" – objects are copied by reference
```

✅ 11. String Operations

```
let fname = "Swagat";
let greet = `Hello, ${fname}!`; // Template literal

console.log(fname.length);
console.log(fname.toUpperCase());
console.log(fname.slice(0, 3)); // "Swa"
```

✅ 12. Typeof Operator

```
console.log(typeof "hello");     // string
console.log(typeof 123);         // number
console.log(typeof true);        // boolean
console.log(typeof null);        // object (weird but true)
console.log(typeof undefined);   // undefined
console.log(typeof Symbol());    // symbol
console.log(typeof {});          // object
```

Would you like these exported to a `.docx` or `.pdf`? Or want me to continue this with functions, conditionals, loops, arrays, etc.?