JavaScript Day 3 Notes – Type Conversion & Booleans

1. Checking Data Types (typeof)

- typeof tells you what type a variable is.
- **Example:**

js

CopyEdit

let score = "abc";

console.log(typeof score); // Output: string

console.log(typeof(score)); // Same output: string

---

2. Type Conversion (Changing Data Types)

(A) Convert to Number

- **Number()** converts other types to numbers.
- **Examples:**

js

CopyEdit

let valueInNumber = Number("33");

console.log(valueInNumber); // Output: 33

console.log(typeof valueInNumber); // Output: number

Special cases:

js

CopyEdit

console.log(Number("33abc")); // Output: NaN (Not a Number)

console.log(Number(null)); // Output: 0

console.log(Number(undefined)); // Output: NaN

console.log(Number(true)); // Output: 1

console.log(Number(false)); // Output: 0

📌 **Real-life example:**
Imagine a form where users enter their age as a string ("25"). You need to convert it into a number for calculations (e.g., voting eligibility).

(B) Convert to Boolean

- **Boolean()** converts values to true or false.
- **Examples:**

js

CopyEdit

let isLoggedIn = 1;

console.log(Boolean(isLoggedIn)); // Output: true

✅ **Truthful Values:**

- 1, any number except 0, "Pranay" (non-empty string), true ➜ **true**

❌ **Falsy Values:**

- 0, "" (empty string), null, undefined, NaN, false ➜ **false**

📌 **Real-life example:**
Login systems:

- isLoggedIn = 1 ➜ User is logged in (true)
- isLoggedIn = 0 ➜ User is logged out (false)

(C) Convert to String

- **String()** converts any value to a string.
- **Example:**

js

CopyEdit

let someNumber = 33;

let stringNumber = String(someNumber);

console.log(stringNumber); // Output: "33"

console.log(typeof stringNumber); // Output: string

📌 **Real-life example:**

- Converting a number to a string for display purposes (e.g., "Your score is: 33").

---

3. Quick Summary of Conversions

Original Value	Number() Output	Boolean() Output	String() Output
"33"	33	true	"33"

Original Value	Number() Output	Boolean() Output	String() Output
"33abc"	NaN	true	"33abc"
true	1	true	"true"
false	0	false	"false"
null	0	false	"null"
undefined	NaN	false	"undefined"
0	0	false	"0"
1	1	true	"1"

🛠️ JavaScript Day 4 Notes – Operations

🎯 1. String and Number Operations

In JavaScript, when you combine strings and numbers, JavaScript decides whether to treat the numbers as numbers or convert them to strings. Let's break it down:

console.log("1" + 2); // Output: "12"

✅ **Why?**: "1" is a string. When you add a number to a string, JavaScript converts the number to a string and **joins** them. So "1" + 2 becomes "1" + "2" ➔ **"12"**.

console.log(1 + "2"); // Output: "12"

✅ **Why?**: Same thing! The number 1 converts to a string, so it becomes "1" + "2" ➔ **"12"**.

console.log("1" + 2 + 2); // Output: "122"

✅ **Why?**: First, "1" + 2 ➔ **"12"**, then "12" + 2 ➔ **"122"**.

console.log(1 + 2 + "2"); // Output: "32"

✅ **Why?**: Here, 1 + 2 happens first (because numbers add first) ➔ **3**, then **3 + "2" ➔ "32"**.

🔥 2. Unary Operators (+)

Unary + tries to convert things to numbers.

console.log(+true); // Output: 1

✅ **Why?**: true is converted to **1**.

console.log(+""); // Output: 0

✅ **Why?**: An empty string becomes **0**.

🔄 **3. Prefix and Postfix**

✅ **Prefix (++x) →** Increases the value first, then uses it. ✅ **Postfix (x++) →** Uses the value first, then increases it.

let a = 5;

console.log(++a); // Output: 6 (increments first, then prints)

let b = 5;

console.log(b++); // Output: 5 (prints first, then increments)

console.log(b); // Output: 6 (b is now increased)
