⬛ LECTURE 7 — NUMBERS & MATH IN JAVASCRIPT

📘 **JavaScript Core — Numbers, Precision & Math Object**
🧠 *First Principles • Calculations • Real-Life Usage*

⬛ FIRST PRINCIPLE — "NUMBER HOTA KYA HAI?"

🔷 Simple Soch (Real-Life)

Paise, likes, marks, score, distance —
👉 **sab numbers hote hain**

JavaScript me **numbers** ka use:

- calculations ke liye
- comparison ke liye
- logic banane ke liye

🟨 **IMPORTANT**

JavaScript me **number ek hi type** hota hai
(integer + decimal alag-alag nahi)

⬛ NUMBERS IN JAVASCRIPT — TWO WAYS

① Primitive Number

```
let num1 = 231;

console.log(typeof num1); // "number"
```

② Number Object (Wrapper Class)

```
let num2 = new Number(231);

console.log(typeof num2); // "object"
```

🟥 DIFFERENCE (VERY IMPORTANT)

Point	Primitive	Object
Creation	`231`	`new Number(231)`
typeof	`"number"`	`"object"`
Memory	Stack	Heap
Use	✅ Recommended	❌ Avoid

🟨 COMPARISON BEHAVIOR

```
let num1 = 231;

let num2 = new Number(231);

let num3 = new Number(231);


console.log(num1 == num2); // true

console.log(num2 == num3); // false
```

🧠 Reason

- `==` → object ko primitive me convert kar deta hai
- Object vs Object → **memory address compare hota hai**

📌 Best Practice

❌ `new Number()` avoid karo
✅ Primitive number use karo

⬛⬛ NUMBER METHODS (IMPORTANT) ⬛⬛

① `toFixed(n)` — Fixed Decimal Places

```
let num = 231.689;

console.log(num.toFixed(1)); // "231.7"
```

🟨 **Use Case**

Currency, price, GST, percentage

---

② `toPrecision(n)` — Total Length Format

```
let num = 231.689;

console.log(num.toPrecision(5)); // "231.69"
```

🧠 **Difference**

- `toFixed()` → decimal control
- `toPrecision()` → total digits control

---

③ `toString()` — Number → String

```
let num = 42;

console.log(num.toString()); // "42"
```

🟨 **Use Case**

UI display, concatenation, form data

---

④ `toExponential(n)` — Scientific Notation

```
let num = 12345.6789;
```

```
console.log(num.toExponential(2)); // "1.23e+4"
```

🟨 **Use Case**

Very large / scientific values

⬛⬛ MATH OBJECT IN JAVASCRIPT ⬛⬛

🔷 **First Principle**

JavaScript ek **global `Math` object** deta hai
jo **advanced mathematical operations** ke liye hota hai.

⬛ MATH CONSTANTS

```
console.log(Math.PI);   // 3.14159...
console.log(Math.E);    // 2.71828...
console.log(Math.LN10); // 2.30258...
```

🟨 **Use**

Geometry, scientific formulas

⬛ ROUNDING METHODS

```
console.log(Math.round(4.6)); // 5
console.log(Math.floor(4.9)); // 4
console.log(Math.ceil(4.1));  // 5
```

Method	Kaam
round	Nearest integer

floor	Neeche ki taraf
ceil	Upar ki taraf

⬛ POWER & ROOTS

```
console.log(Math.pow(2, 3)); // 8
console.log(Math.sqrt(16));  // 4
```

🟨 **Tip**

`x ** y` bhi power ke liye use kar sakte ho

⬛ MIN & MAX VALUES

```
console.log(Math.min(5, 2, 9)); // 2
console.log(Math.max(5, 2, 9)); // 9
```

🟨 **Real-Life**

Lowest price, highest score

⬛ RANDOM NUMBERS — Math.random() ⬛

🔷 First Principle

Computer **true random** nahi hota
👉 Isliye **pseudo-random numbers** generate karta hai

Basic Usage

```
console.log(Math.random());      // 0 to <1
console.log(Math.random() * 10); // 0 to <10
```

INTEGER RANGES

```
Math.floor(Math.random() * 10);      // 0–9

Math.floor(Math.random() * 10) + 1;  // 1–10

Math.floor(Math.random() * 10) + 11; // 11–20
```

⬛ GENERAL FORMULA (VERY IMPORTANT)

```
Math.floor(Math.random() * (max - min + 1)) + min
```

Example: 30–40

```
Math.floor(Math.random() * (40 - 30 + 1)) + 30;
```

🟨 **Use Cases**

- Games (dice, cards)
- Random colors
- Random IDs

⬛⬛SHORT SUMMARY — EK NAZAR ME ⬛⬛

✅ Numbers → Primitive & Object
✅ Prefer → Primitive numbers
✅ Methods → `toFixed`, `toPrecision`, `toString`, `toExponential`
✅ Math Object → constants + methods
✅ Rounding → round, floor, ceil
✅ Random numbers → `Math.random()`
✅ Universal formula → `(Math.random() * (max - min + 1)) + min`

⬛⬛FINAL THOUGHT ⬛⬛

🧠 **Numbers + Math = JavaScript ka calculation engine**
Isko master kar liya →
👉 **Games, finance, analytics sab easy**