🎲 Simulating a Dice Roll in JavaScript

When rolling a die, we want a **random** number between **1 and 6**. Here's how we can achieve that using Math.random().

1️⃣ Basic Dice Roll Simulation

```
var diceRoll = Math.floor(Math.random() * 6) + 1;
console.log(diceRoll);
```

◆ How It Works

1. Math.random() generates a random decimal between **0.0000000000000000 and 0.9999999999999999**.
2. Multiplying by **6** gives a range of **0 to 5.9999999999999999**.
3. Math.floor() rounds down to the nearest integer (**0 to 5**).
4. Adding **1** shifts the range to **1 to 6**.

2️⃣ Rolling Multiple Dice

Want to roll **two dice** at the same time?

```
var dice1 = Math.floor(Math.random() * 6) + 1;
var dice2 = Math.floor(Math.random() * 6) + 1;
console.log(`🎲 Dice 1: ${dice1}, 🎲 Dice 2: ${dice2}`);
```

3️⃣ Simulating Multiple Rolls

Want to simulate rolling a die **10 times**?

```
for (var i = 0; i < 10; i++) {
    console.log(`Roll ${i + 1}:`, Math.floor(Math.random() * 6) + 1);
}
```

4️⃣ Creating a Function for Dice Rolls

Encapsulating this logic in a function makes it reusable:

```
function rollDice(sides) {
    return Math.floor(Math.random() * sides) + 1;
```

```
}

console.log(rollDice(6)); // 🎲 Simulating a standard 6-sided dice
console.log(rollDice(20)); // 🎲 Simulating a 20-sided D&D dice
```

5️⃣ Simulating a Dice Game (e.g., Highest Roll Wins)

```
var player1 = rollDice(6);
var player2 = rollDice(6);

console.log(`Player 1 rolled: ${player1}`);
console.log(`Player 2 rolled: ${player2}`);

if (player1 > player2) {
    console.log("🎉 Player 1 Wins!");
} else if (player2 > player1) {
    console.log("🎉 Player 2 Wins!");
} else {
    console.log("🤝 It's a tie!");
}
```

🔥 Recap

Method	Description
Math.random()	Generates a random decimal between 0 and 1
Math.floor()	Rounds down to the nearest integer
Math.random() * 6 + 1	Generates a random integer from **1 to 6**
Custom function	rollDice(sides) allows rolling **any sided dice**