

# Random Numbers with JavaScript

```
// Generate a random integer between min (inclusive) and max (inclusive)
function getRandomInt(min, max) {
  min = Math.ceil(min);
  max = Math.floor(max);
  return Math.floor(Math.random() * (max - min + 1)) + min;
}

const randomNum = getRandomInt(1, 100);
```

Generating Random Numbers with Math.random():	2
Generating Random Decimals:	3
Generating Random Boolean Values:	3
Seeding Random Numbers (Deterministic Randomness):	4
10 coding exercises	5
Exercise 1: Calculate Factorial	6
Exercise 2: Find Largest Number in an Array	7
Exercise 3: Palindrome Check	8
Exercise 4: Reverse a String	9
Exercise 5: JavaScriptFun	10

Random numbers are a fundamental concept in programming and are often used in JavaScript for various purposes, such as generating game scenarios, creating random user experiences, or conducting simulations. In JavaScript, you can generate random numbers using the built-in Math object or by utilizing the Math.random() method.

Learn more about JavaScript with Examples and Source Code Laurence Svekis  
Courses <https://basescripts.com/>

## Generating Random Numbers with Math.random():

The Math.random() method in JavaScript generates a random floating-point number between 0 (inclusive) and 1 (exclusive). It provides a pseudo-random number, which means it's not truly random but appears random enough for most applications.

To generate a random number within a specific range, you can use mathematical operations. Here's how you can generate random integers within a given range:

```
// Generate a random integer between min (inclusive)
and max (inclusive)
function getRandomInt(min, max) {
    min = Math.ceil(min);
    max = Math.floor(max);
    return Math.floor(Math.random() * (max - min + 1))
+ min;
}

const randomNum = getRandomInt(1, 100);
console.log(randomNum); // Outputs a random integer
between 1 and 100 (inclusive)
```

In this example, `Math.random()` is used to generate a random decimal number between 0 and 1. By applying some mathematical operations, you can transform it into an integer within the desired range.

## Generating Random Decimals:

If you need a random decimal number within a specific range, you can adapt the previous function:

```
// Generate a random decimal number between min
(inclusive) and max (inclusive)
function getRandomDecimal(min, max) {
    return Math.random() * (max - min) + min;
}

const randomFloat = getRandomDecimal(0.5, 2.5);
console.log(randomFloat); // Outputs a random decimal
between 0.5 (inclusive) and 2.5 (exclusive)
```

## Generating Random Boolean Values:

To generate random true/false or 1/0 values (boolean), you can use `Math.random()` along with comparison operators:

```
function getRandomBoolean() {  
    return Math.random() < 0.5; // 50% chance of true,  
    50% chance of false  
}
```

```
const randomBool = getRandomBoolean();  
console.log(randomBool); // Outputs a random true or  
false
```

## Seeding Random Numbers (Deterministic Randomness):

If you need reproducible random numbers, you can "seed" the random number generator using a specific value. However, JavaScript's `Math.random()` does not provide a built-in seeding mechanism. To achieve deterministic randomness, you may need to use external libraries or write your own custom random number generator.

In summary, random numbers are a versatile tool in JavaScript programming. You can use `Math.random()` to generate random numbers for various purposes, such as games, simulations, or randomized algorithms. Understanding how to manipulate and use random numbers is a valuable skill for any JavaScript developer.

# 10 coding exercises

Exercise to help you practice generating random numbers in JavaScript:

1. **Random Number Range:**

Write a function that generates a random number between two given integers, min and max.

2. **Dice Roll Simulator:**

Create a program that simulates rolling a six-sided die. Print the result of each roll.

3. **Random Color Generator:**

Build a function that generates a random RGB color code (e.g., "#RRGGBB") and displays it on the screen.

4. **Coin Flip Simulator:**

Write a function that simulates flipping a coin (heads or tails). Print the result of each flip.

5. **Random Password Generator:**

Create a function that generates a random password of a specified length. The password should include a mix of uppercase letters, lowercase letters, numbers, and special characters.

6. **Random Quote Generator:**

Build a program that selects and displays a random quote from a predefined list of quotes each time it's run.

7. **Random Name Selector:**

Write a script that randomly selects a name from an array of names and prints it. It should ensure that the same name isn't selected twice in a row.

Learn more about JavaScript with Examples and Source Code Laurence Svekis  
Courses <https://basescripts.com/>

#### 8. **Random Sentence Generator:**

Create a program that generates random sentences by combining random words from predefined lists of nouns, verbs, and adjectives.

#### 9. **Randomized Quiz Questions:**

Build a quiz application that randomizes the order of questions and answer choices each time it's taken.

#### 10. **Randomized Card Shuffle:**

Implement a card shuffling function that shuffles a deck of cards represented as an array of card objects (e.g., [{ rank: 'Ace', suit: 'Spades' }, ...]). Ensure that the cards are shuffled randomly.

These exercises cover a range of applications for random number generation in JavaScript, from simple simulations to more complex scenarios like creating random passwords or randomized quiz questions. They will help you become more comfortable with generating and using random numbers in your JavaScript programs.

## Exercise 1: Calculate Factorial

Description: Write a JavaScript function to calculate the factorial of a given integer. The factorial of a non-negative integer  $n$  is the product of all positive integers less than or equal to  $n$ .

Summary: The function will take an integer  $n$  as input and return the factorial of  $n$ .

```
function calculateFactorial(n) {
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis  
Courses <https://basescripts.com/>

```
    if (n === 0 || n === 1) {
      return 1;
    } else {
      let factorial = 1;
      for (let i = 2; i <= n; i++) {
        factorial *= i;
      }
      return factorial;
    }
  }
}
```

// Example usage:

```
const result = calculateFactorial(5); // Calculates 5!
console.log(result); // Outputs 120
```

## Exercise 2: Find Largest Number in an Array

Description: Write a JavaScript function to find the largest number in an array of numbers.

Summary: The function will take an array of numbers as input and return the largest number in the array.

```
function findLargestNumber(arr) {
  if (arr.length === 0) {
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis  
Courses <https://basescripts.com/>

```
        return null;
    } else {
        let largest = arr[0];
        for (let i = 1; i < arr.length; i++) {
            if (arr[i] > largest) {
                largest = arr[i];
            }
        }
        return largest;
    }
}
```

```
// Example usage:
const numbers = [10, 5, 8, 21, 3];
const largestNumber = findLargestNumber(numbers);
console.log(largestNumber); // Outputs 21
```

## Exercise 3: Palindrome Check

Description: Write a JavaScript function that checks if a given string is a palindrome. A palindrome is a word, phrase, or sequence that reads the same backward as forward (ignoring spaces, punctuation, and capitalization).



Summary: The function will take a string as input and return true if it's a palindrome, or false if it's not.

```
function isPalindrome(str) {  
    str = str.toLowerCase().replace(/[^a-zA-Z0-9]/g,  
    ''); // Remove non-alphanumeric characters and convert  
    to lowercase  
    const reversedStr =  
    str.split('').reverse().join('');  
    return str === reversedStr;  
}
```

```
// Example usage:  
const testString = "A man, a plan, a canal, Panama";  
const isPalin = isPalindrome(testString);  
console.log(isPalin); // Outputs true
```

## Exercise 4: Reverse a String

Description: Write a JavaScript function that reverses a given string.

Summary: The function will take a string as input and return a new string with its characters reversed.

```
function reverseString(str) {
    return str.split('').reverse().join('');
}

// Example usage:
const inputString = "Hello, World!";
const reversed = reverseString(inputString);
console.log(reversed); // Outputs "!dlroW ,olleH"
```

## Exercise 5: JavaScriptFun

Description: Write a JavaScript program that prints numbers from 1 to 100. For multiples of 3, print "JavaScript" instead of the number, and for multiples of 5, print "Fun." For numbers that are multiples of both 3 and 5, print "JavaScriptFun."

Summary: The program will loop from 1 to 100 and apply the JavaScriptFun rules to each number.

```
for (let i = 1; i <= 100; i++) {
    if (i % 3 === 0 && i % 5 === 0) {
        console.log("JavaScriptFun");
    } else if (i % 3 === 0) {
        console.log("JavaScript");
    } else if (i % 5 === 0) {
```

```
        console.log("Fun");
    } else {
        console.log(i);
    }
}
```

These coding exercises cover a range of JavaScript concepts, including loops, conditionals, functions, and string manipulation. They are suitable for practicing and reinforcing your JavaScript skills.