**Topic:  Number & Math methods**

**Number methods**

**toFixed()** method formats a number using fixed-point notation, which means it returns a string representation of the number with a specified number of decimal places. This is useful for rounding numbers to a certain number of decimal places.

let num = 123.45678;

console.log(num.toFixed(2)); // "123.46" - rounded to two decimal places

console.log(num.toFixed(3)); // "123.457" - rounded to three decimal places

console.log(num.toFixed(0)); // "123" - no decimal places, rounds to nearest integer

let num1 = 150;

console.log(num1.toFixed(2)); // "150.00" - Adds two decimal places with zeros

console.log(num1.toFixed(0)); // "150" - No change

**parseInt():** Parses a string argument and returns an integer.

let str = "123";

console.log(parseInt(str)); // Output: 123

**parseFloat():** Parses a string argument and returns a floating point number.

**// Example 1: Basic Parsing**

console.log(parseInt("42")); // 42 (simple integer parsing)

console.log(parseFloat("42")); // 42 (simple float parsing)

console.log(parseFloat("42.5")); // 42.5 (floating-point parsing)

**// Example 2: Handling Leading and Trailing Whitespaces**

console.log(parseInt("   123   ")); // 123 (leading/trailing spaces ignored)

console.log(parseFloat("   123.45   ")); // 123.45 (leading/trailing spaces ignored)

**// Example 3: Parsing with Non-Numeric Characters**

console.log(parseInt("123abc")); // 123 (parsing stops at "abc")

console.log(parseFloat("123.45abc")); // 123.45 (parsing stops at "abc")

**// Example 4: Strings Starting with Non-Numeric Characters**

console.log(parseInt("abc123")); // NaN (no leading numeric characters)

console.log(parseFloat("abc123.45")); // NaN (no leading numeric characters)

**// Example 5: Handling Float Strings in parseInt()**

console.log(parseInt("3.14")); // 3 (truncates the decimal part)

console.log(parseFloat("3.14")); // 3.14 (returns the full float)

**// Example 6: Parsing Strings with Exponential Notation**

console.log(parseInt("1e4")); // 1 (stops at "e")

console.log(parseFloat("1e4")); // 10000 (interpreted as 1 \* 10^4)

**// Example 8: Large and Small Numbers**

console.log(parseInt("999999999999999999999")); // 1e+21 (very large integer)

console.log(parseFloat("999999999999999999999")); // 1e+21 (very large float)

console.log(parseInt("0.0001")); // 0 (fractional part ignored)

console.log(parseFloat("0.0001")); // 0.0001 (floating-point number)

**isNaN():** Checks if a value is NaN (Not-a-Number). If it is number it returns false if it is not a number it returns true.

NUMBER – False

Not a Number - True

console.log(isNaN("hello")); // true

Output: true console.log(isNaN(123)); // Output: false

**5. Number method:** The Number constructor converts a value to a number.

console.log(Number("123")); // 123

console.log(Number("123abc")); // NaN

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

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

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

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

**Type Coercion:**

isNaN() first tries to convert the parameter to a number, and then tests if the resulting value is NaN.

isNaN(NaN); // true

isNaN(undefined); // true

isNaN({}); // true

isNaN(true); // false

isNaN(false);//false

isNaN(null); // false

isNaN(37); // false

**// Strings**

isNaN("37"); // false: "37" is converted to the number 37 which is not NaN

isNaN("37.37"); // false: "37.37" is converted to the number 37.37 which is not NaN

isNaN("37,5"); // true

isNaN("123ABC","jhkhk"); // true: Number("123ABC") is NaN

isNaN(""); // false: the empty string is converted to 0 which is not NaN

isNaN(" "); // false: a string with spaces is converted to 0 which is not NaN

**// Dates**

isNaN(new Date()); // false; Date objects can be converted to a number (timestamp)

isNaN(new Date().toString()); // true; the string representation of a Date object cannot be parsed as a number

**// Arrays**

isNaN([]); // false; the primitive representation is "", which coverts to the number 0

isNaN([1]); // false; the primitive representation is "1"

isNaN([1, 2]); // true; the primitive representation is "1,2", which cannot be parsed as number

**Math methods**

**1. Math.abs():** Returns the absolute value of a number.

        console.log(Math.abs(10)); // 10

        console.log(Math.abs(-10)); // 10

        console.log(Math.abs(0)); // 0

        console.log(Math.abs(-0)); // 0

        console.log(Math.abs("-42")); // 42 (string converted to number)

        console.log(Math.abs(null)); // 0 (null converted to 0)

        console.log(Math.abs("Hello")); // NaN (string that can't be converted to a number)

**2. Math.ceil():** Rounds a number up to the next largest integer.

        console.log(Math.ceil(4.2)); // 5

        console.log(Math.ceil(-4.2)); // -4

        console.log(Math.ceil(0)); // 0

        console.log(Math.ceil(7.004)); // 8

        console.log(Math.ceil(-7.004)); // -7

**3. Math.floor():** Rounds a number down to the previous largest integer.

        console.log(Math.floor(4.7)); // 4

        console.log(Math.floor(-4.7)); // -5

        console.log(Math.floor(0)); // 0

        console.log(Math.floor(7.999)); // 7

        console.log(Math.floor(-7.999)); // -8

**4. Math.round()**

Rounds a number to the nearest integer. If the fractional part is 0.5 or greater, the argument is rounded to the next higher integer.

        console.log(Math.round(4.5)); // 5

        console.log(Math.round(4.4)); // 4

        console.log(Math.round(-4.5)); // -4

        console.log(Math.round(-4.6)); // -5

        console.log(Math.round(7.999)); // 8

        console.log(Math.round(-7.999)); // -8

**5. Math.trunc()**

Returns the integer part of a number by removing any fractional digits.

    console.log(Math.trunc(4.9)); // 4

    console.log(Math.trunc(-4.9)); // -4

    console.log(Math.trunc(0)); // 0

    console.log(Math.trunc(7.004)); // 7

    console.log(Math.trunc(-7.004)); // -7

**6. Math.max()** Returns the largest of zero or more numbers

    console.log(Math.max(1, 2, 3)); // 3

    console.log(Math.max(-1, -2, -3)); // -1

    console.log(Math.max(1, 2, 3, 10, 20)); // 20

**7. Math.min():** Returns the smallest of zero or more numbers.

    console.log(Math.min(1, 2, 3)); // 1

    console.log(Math.min(-1, -2, -3)); // -3

    console.log(Math.min(1, 2, 3, 10, 20)); // 1

**8. Math.pow():** Returns the base raised to the power of the exponent.

    console.log(Math.pow(2, 3)); // 8 (2^3)

    console.log(Math.pow(5, 2)); // 25 (5^2)

    console.log(Math.pow(4, 0.5)); // 2 (square root of 4)

    console.log(Math.pow(-7, 2)); // 49 (negative base, even exponent)

**9. Math.sqrt():** Returns the square root of a number.

    console.log(Math.sqrt(16)); // 4

    console.log(Math.sqrt(9)); // 3

    console.log(Math.sqrt(0)); // 0

**10. Math.random():** Returns a pseudo-random number between 0 (inclusive) and 1 (exclusive).

    console.log(Math.random()); // Random number between 0 and 1

    console.log(Math.random() \* 10); // Random number between 0 and 10

    console.log(Math.floor(Math.random() \* 10)); // Random integer between 0 and 9

    console.log(Math.floor(Math.random() \* 100) + 1); // Random integer between 1 and 100