Numbers

In JavaScript, numbers are fundamental data types used to represent numeric values. Here's a brief explanation of numbers in JavaScript:

Data Type: Numbers in JavaScript are represented using the `Number` data type. They can hold both integer and floating-point values.

Numeric Literals: Numeric literals are written numbers in JavaScript code. They can be integers (whole numbers) or floating-point numbers (numbers with decimal points).

Arithmetic Operations: JavaScript supports various arithmetic operations on numbers, including addition (`+`), subtraction (`-`), multiplication (`*`), division (`/`), and modulus (`%`).

Mathematical Functions: The `Math` object in JavaScript provides a set of built-in mathematical functions and constants. These include functions like `Math.sqrt()` (square root), `Math.sin()` (sine), `Math.cos()` (cosine), and constants like `Math.PI` (the mathematical constant π).

Special Values: JavaScript has special values such as `NaN` (Not-a-Number), which represents the result of an undefined or unrepresentable mathematical operation, and `Infinity` and `-Infinity`, which represent positive and negative infinity respectively.

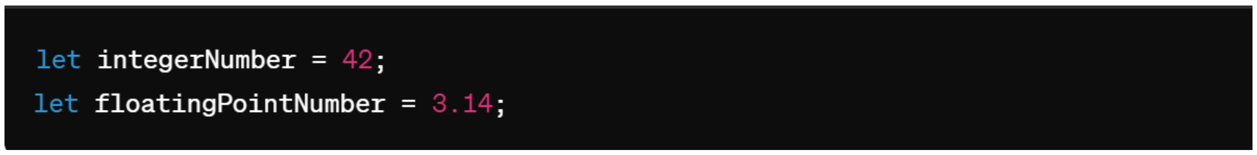
Number Methods: The `Number` object in JavaScript provides methods for working with numbers, including formatting methods like `toFixed()` and `toPrecision()`, conversion methods like `toString()`, and parsing methods like `parseInt()` and `parseFloat()`.

Number Constants: JavaScript defines several constants related to numbers, such as `Number.MAX_VALUE` (the largest representable number), `Number.MIN_VALUE` (the smallest positive number greater than zero), and `Number.EPSILON` (the smallest difference between two representable numbers).

Numbers in JavaScript are represented as 64-bit floating-point values according to the IEEE 754 standard, commonly known as "double-precision" floating-point numbers

Integer and Floating-Point Numbers:

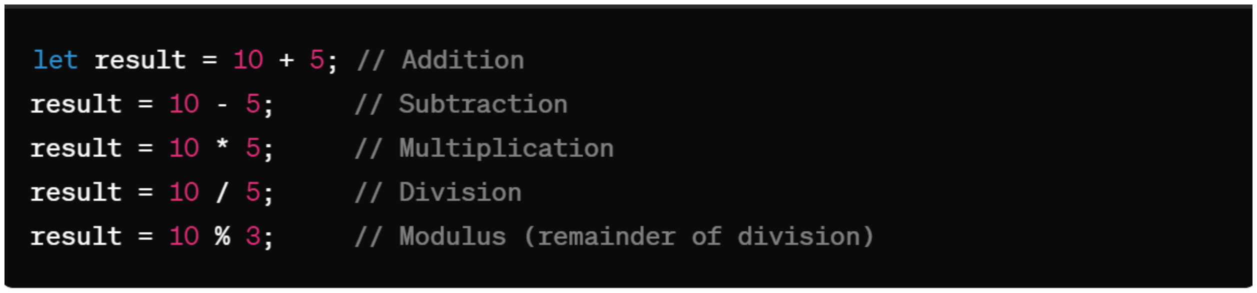
JavaScript supports both integer and floating-point numbers. All numbers in JavaScript, whether integer or floating-point, are represented using the `Number` data type.



```
let integerNumber = 42;
let floatingPointNumber = 3.14;
```

Numeric Literals:

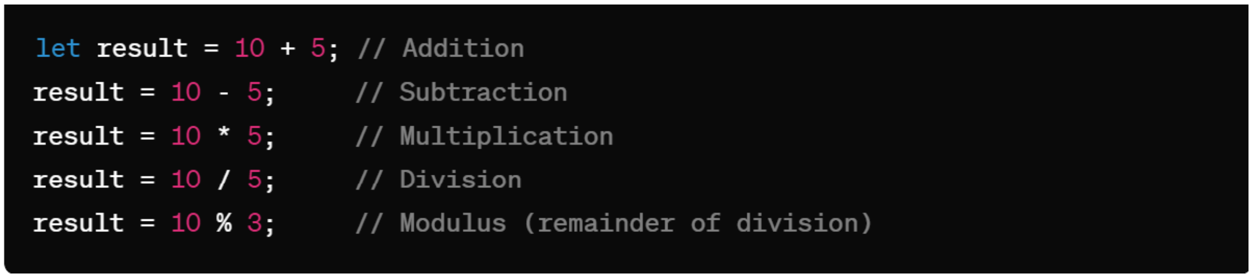
Numeric literals are simply written numbers in JavaScript code. They can be integers or floating-point numbers. For example:



```
let result = 10 + 5; // Addition
result = 10 - 5;     // Subtraction
result = 10 * 5;     // Multiplication
result = 10 / 5;     // Division
result = 10 % 3;     // Modulus (remainder of division)
```

Mathematical Operations:

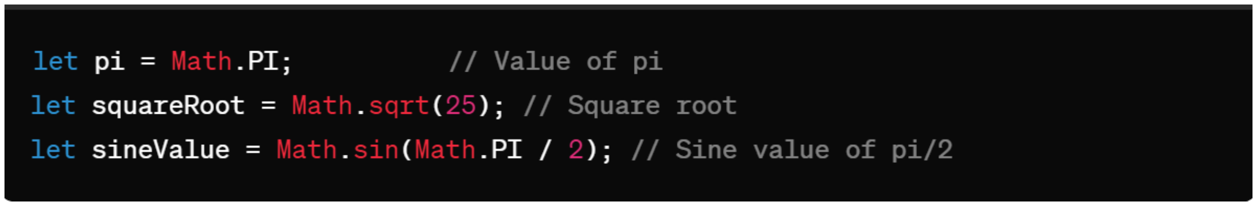
JavaScript supports various mathematical operations, including addition (`+`), subtraction (`-`), multiplication (`*`), division (`/`), and modulus (`%`). For example:



```
let result = 10 + 5; // Addition
result = 10 - 5;     // Subtraction
result = 10 * 5;     // Multiplication
result = 10 / 5;     // Division
result = 10 % 3;     // Modulus (remainder of division)
```

Math Object:

JavaScript provides a built-in `Math` object that contains various mathematical functions and constants. These include trigonometric functions, exponential functions, logarithmic functions, and more. For example:



```
let pi = Math.PI;            // Value of pi
let squareRoot = Math.sqrt(25); // Square root
let sineValue = Math.sin(Math.PI / 2); // Sine value of pi/2
```

NaN (Not a Number):

`NaN` is a special value representing "Not-a-Number." It is returned when a mathematical operation results in an undefined or unrepresentable value. For example:

```
let result = 0 / 0; // NaN
```

Infinity and -Infinity:

`Infinity` represents positive infinity, while `-Infinity` represents negative infinity. These values are returned when a number exceeds the upper or lower limit of representable values. For example:

```
let positiveInfinity = Infinity;
let negativeInfinity = -Infinity;
```

Number Methods:

The `Number` object in JavaScript also provides several methods for working with numbers. These include `toFixed()`, `toPrecision()`, `toString()`, `parseInt()`, `parseFloat()`, and more.

toFixed(): Converts a number into a string, rounding the number to a specified number of decimal places.

```
let num = 3.14159;
console.log(num.toFixed(2)); // Output: "3.14"
```

toPrecision(): Returns a string representing the number to a specified precision.

```
let num = 12345;
console.log(num.toPrecision(2)); // Output: "1.2e+4"
```

toString(): Returns a string representing the specified number.

```
let num = 42;
console.log(num.toString()); // Output: "42"
```

parseInt(): Parses a string argument and returns an integer of the specified radix (the base in mathematical numeral systems).

```
let str = "10";
console.log(parseInt(str)); // Output: 10
```

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

```
let str = "3.14";
console.log(parseFloat(str)); // Output: 3.14
```

isNaN(): Determines whether a value is NaN (Not-a-Number).

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

isFinite(): Determines whether a value is a finite, legal number.

```
console.log(isFinite(42)); // Output: true
console.log(isFinite(Infinity)); // Output: false
```

Number.isInteger(): Determines whether the passed value is an integer.

```
console.log(Number.isInteger(42)); // Output: true
console.log(Number.isInteger(42.5)); // Output: false
```

Number.parseFloat(): Parses an argument and returns a floating point number.

```
let str = "3.14";
console.log(Number.parseFloat(str)); // Output: 3.14
```

Number.parseInt(): Parses a string argument and returns an integer of the specified radix.

```
let str = "10";
console.log(Number.parseInt(str)); // Output: 10
```

These methods are useful for various tasks such as formatting numbers, converting numbers to strings, parsing strings into numbers, and checking number-related properties.

Math

JavaScript provides the `Math` object, which contains a set of properties and methods for mathematical operations.

Math.PI: Returns the mathematical constant π (pi).

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

Math.ceil(): Rounds a number up to the nearest integer.

Math.floor(): Rounds a number down to the nearest integer.

Math.round(): Rounds a number to the nearest integer.

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

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

Math.pow(): Returns the base to the exponent power.

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

Math.random(): Returns a random floating-point number between 0 (inclusive) and 1 (exclusive).

Math.sin(), Math.cos(), Math.tan(): Trigonometric functions to calculate sine, cosine, and tangent of an angle (in radians), respectively.

Math.log(), Math.exp(): Exponential functions to calculate natural logarithm and exponential of a number, respectively.

```javascript
console.log(Math.PI); // Output: 3.141592653589793
console.log(Math.abs(-5)); // Output: 5
console.log(Math.ceil(3.14)); // Output: 4
console.log(Math.floor(3.14)); // Output: 3
console.log(Math.round(3.5)); // Output: 4
console.log(Math.max(10, 20, 30)); // Output: 30
console.log(Math.min(10, 20, 30)); // Output: 10
console.log(Math.pow(2, 3)); // Output: 8
console.log(Math.sqrt(16)); // Output: 4
console.log(Math.random()); // Output: A random number between 0 and 1
console.log(Math.sin(Math.PI / 2)); // Output: 1
console.log(Math.log(Math.E)); // Output: 1
```