Good Morning

EVERYONE

Let's Recap

Q 1. Are you **ready** ?

Let's Recap

Q 1. What is a **JavaScript variable?**

Options:

(A) A built-in function

(B) A reserved keyword

(C) A container for storing data values

(D) A data type

Let's Recap

Q 1. What is a **JavaScript variable?**

Options:

(A) A built-in function

(B) A reserved keyword

(C) **A container for storing data values**

(D) A data type

Let's Recap

Q 2. How is a **JavaScript string data type** different from a **JavaScript number data type**?

Options:

(A) Strings are used for arithmetic operations, while numbers are used for text manipulation.

(B) Strings can hold both text and numerical values, while numbers can only hold numeric data.

(C) Both of above

(D) None of above

Let's Recap

Q 2. How is a **JavaScript string data type** different from a **JavaScript number data type**?

Options:

(A) Strings are used for arithmetic operations, while numbers are used for text manipulation.

(B) **Strings can hold both text and numerical values, while numbers can only hold numeric data.**

(C) Both of above

(D) None of above

Let's Recap

Q 3. What is the purpose of the **typeof operator** in JavaScript?

Options:

(A) It checks if a variable is defined.

(B) It converts a variable to a specific data type.

(C) It returns the data type of a value or expression.

(D) None of the above

Let's Recap

Q 3. What is the purpose of the **typeof operator** in JavaScript?

Options:

(A) It checks if a variable is defined.

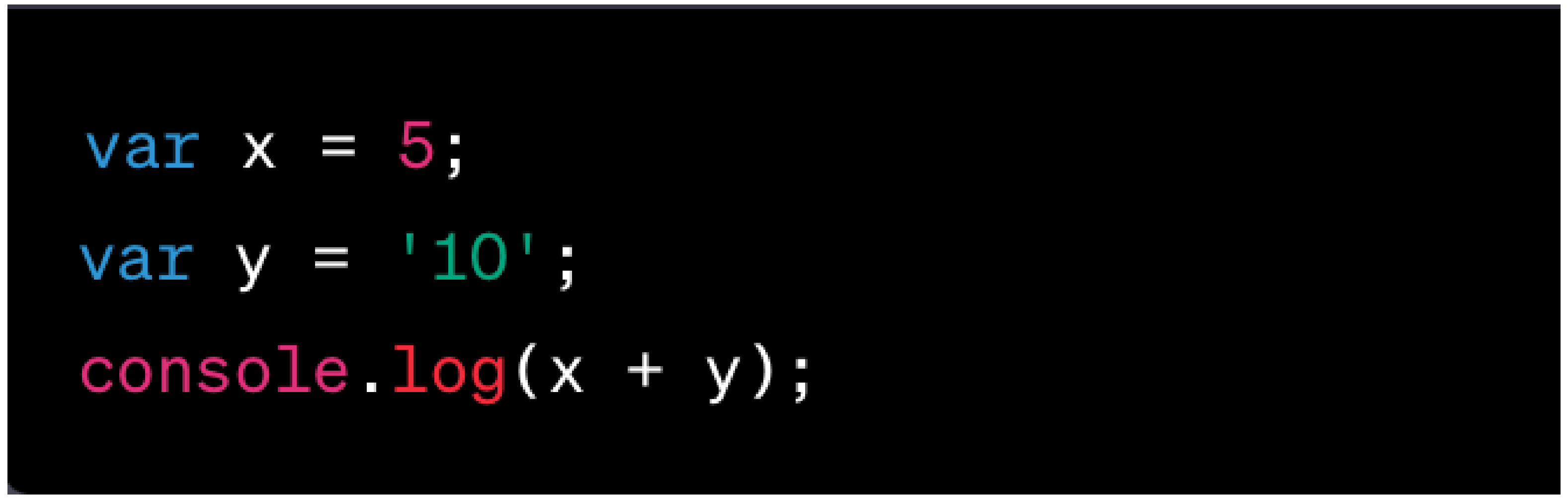
(B) It converts a variable to a specific data type.

(C) **It returns the data type of a value or expression.**

(D) None of the above

Let's Recap

Q 4. What will the following JavaScript code output?



```
var x = 5;
var y = '10';
console.log(x + y);
```

Options:

(A) 510

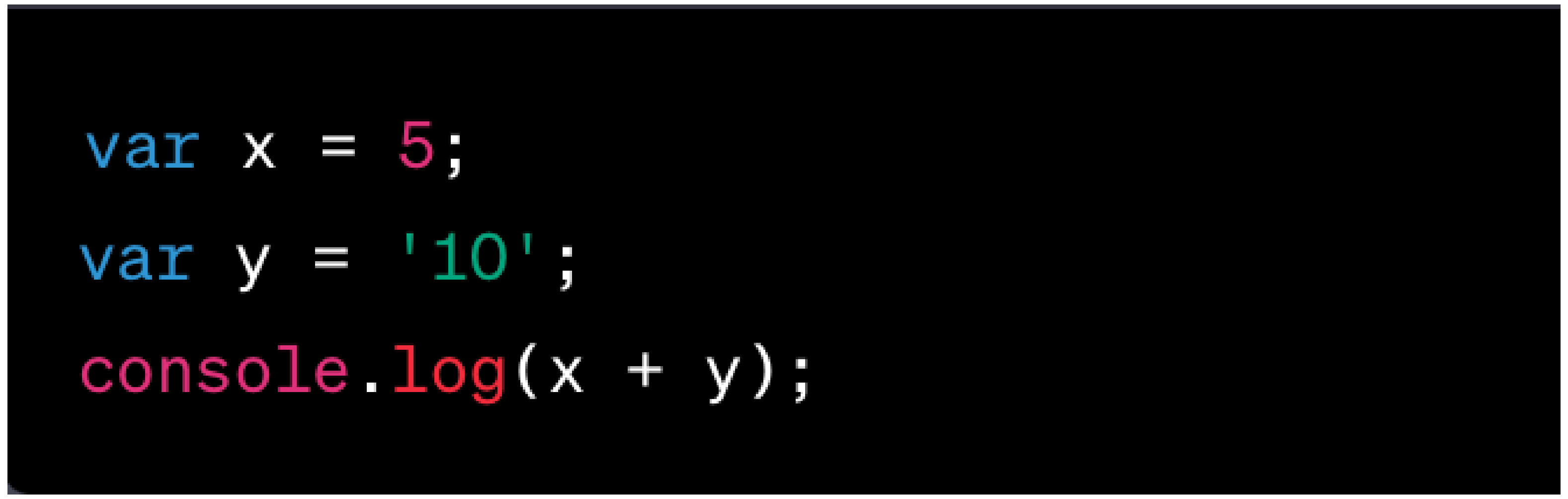
(B) 15

(C) 105

(D) Error

Let's Recap

Q 4. What will the following JavaScript code output?



```
var x = 5;
var y = '10';
console.log(x + y);
```

Options:

(A) **510**

(B) 15

(C) 105

(D) Error

Let's Recap

Q 5. Which of the following is not a valid **JavaScript variable name?**

Options:

(A) myVariable

(B) 123variable

(C) _variable

(D) $variable

Let's Recap

Q 5. Which of the following is not a valid **JavaScript variable name?**

Options:

(A) myVariable

[✓ (B) **123variable**]

(C) _variable

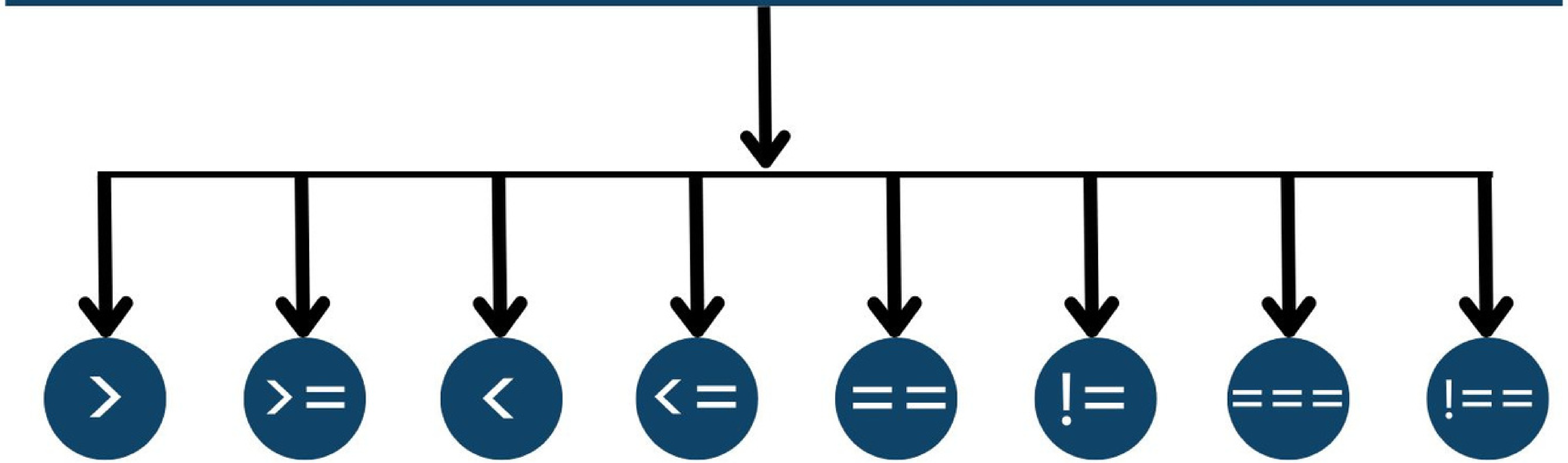
(D) $variable

TODAY AGENDA

1 JS Relational operators

2 JS Comparison operators

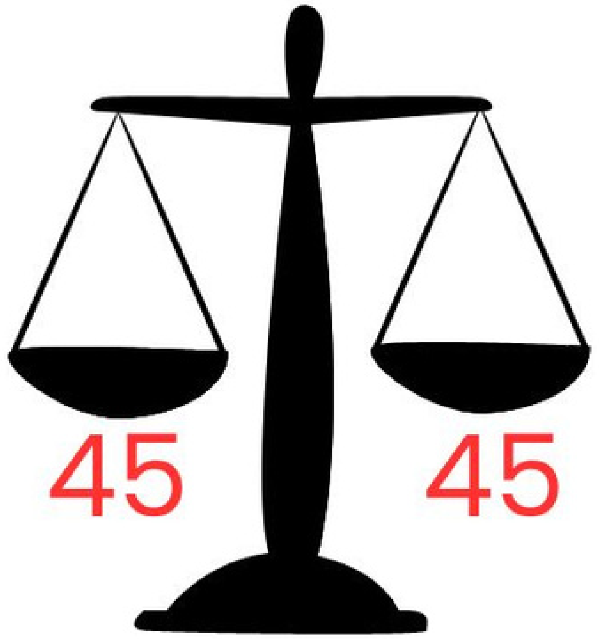
Javascipt Relational and Comparison Operators



`>` `>=` `<` `<=` `==` `!=` `===` `!==`

45 < 90

45 == 45



1

JS Relational operators

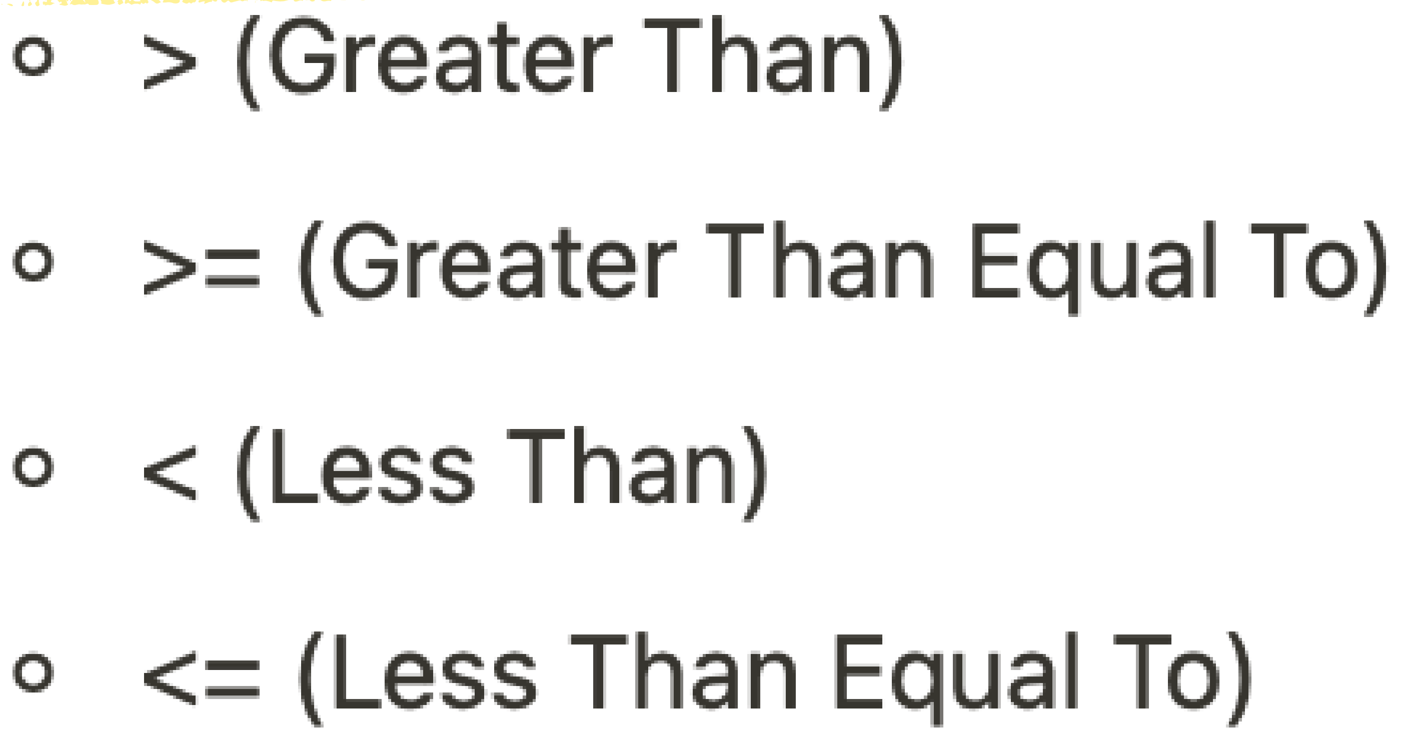
1 JS Relational operators

- **Relational operators** in JavaScript are used to compare two values and return a Boolean result (either true or false).
- These operators are often used to establish the relationship between two operands.

1 JS Relational operators

- **Relational operators** in JavaScript are used to compare two values and return a Boolean result (either true or false).
- These operators are often used to establish the relationship between two operands.

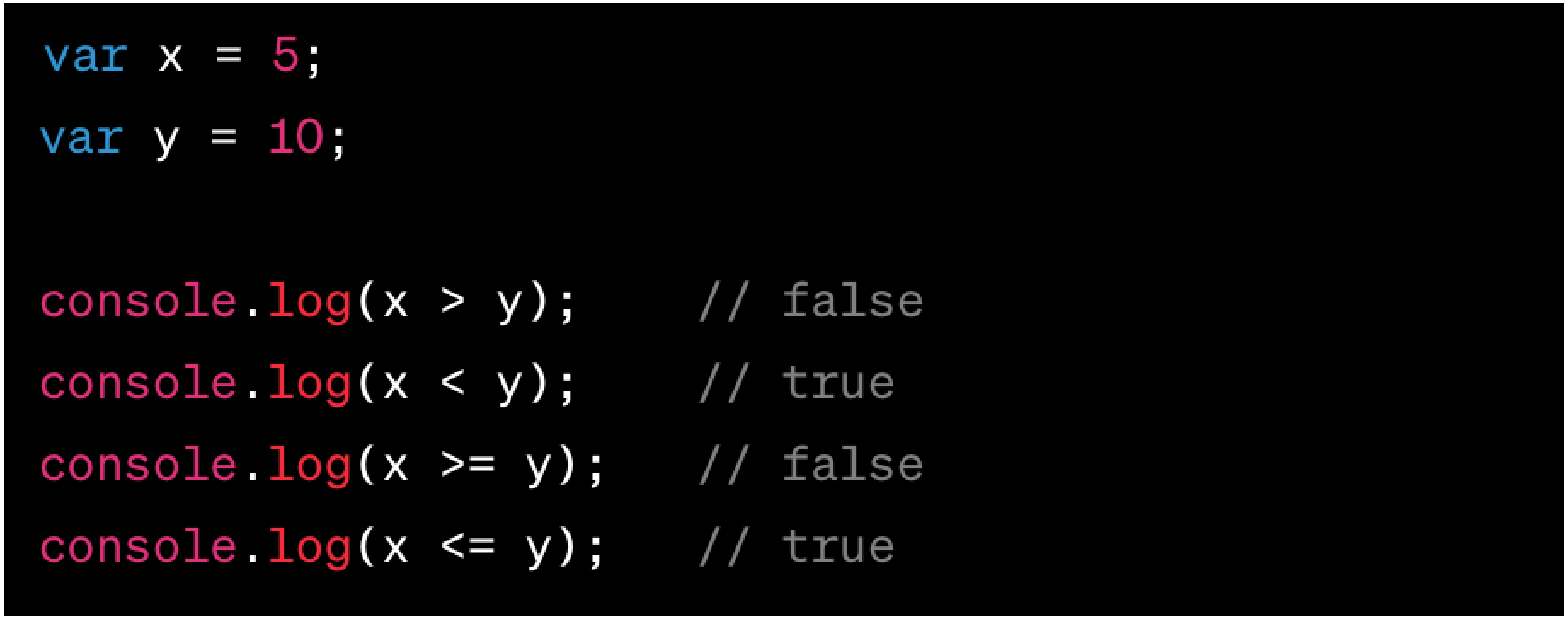
There are **4 relationship operators**



- > (Greater Than)
- >= (Greater Than Equal To)
- < (Less Than)
- <= (Less Than Equal To)

1 JS Relational operators

Example

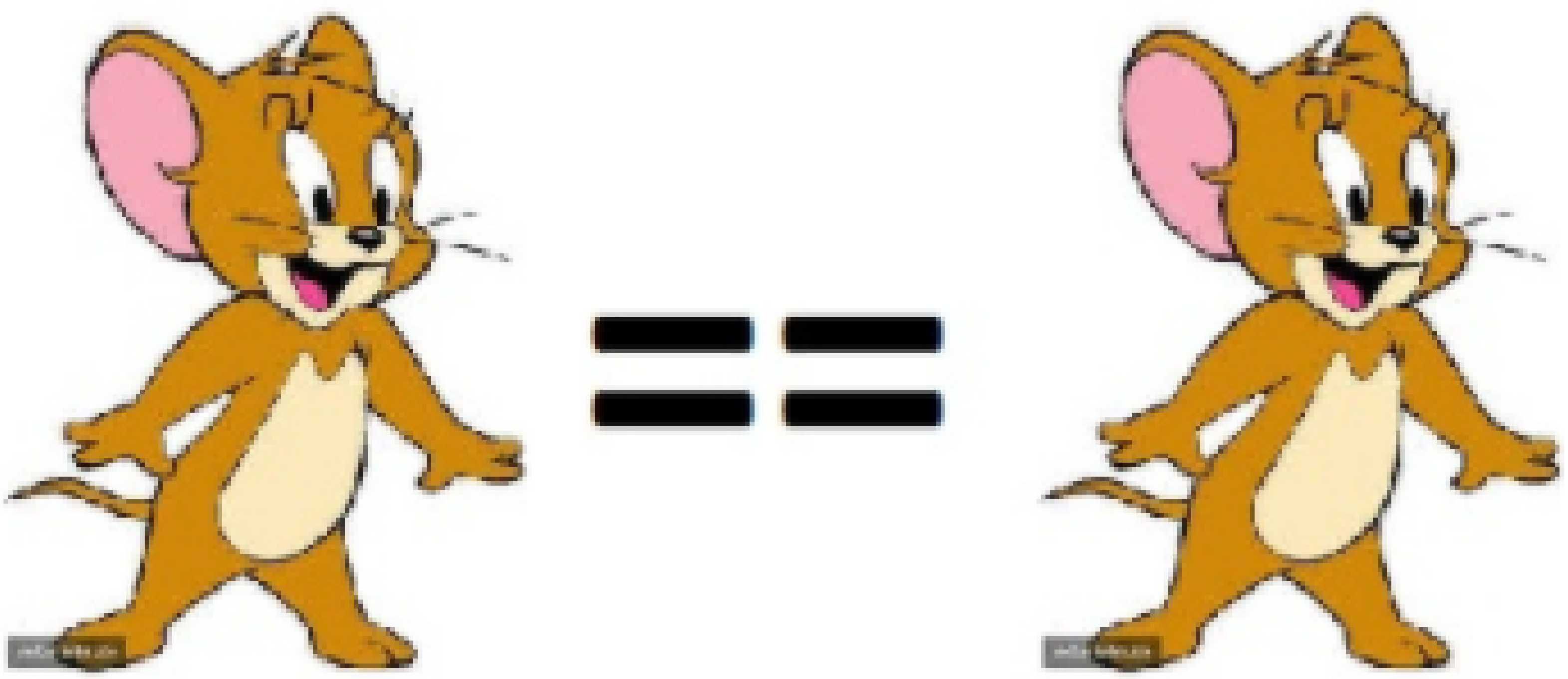


```
var x = 5;
var y = 10;

console.log(x > y);     // false
console.log(x < y);     // true
console.log(x >= y);    // false
console.log(x <= y);    // true
```

2

JS Comparison operators



1 JS Comparison operators

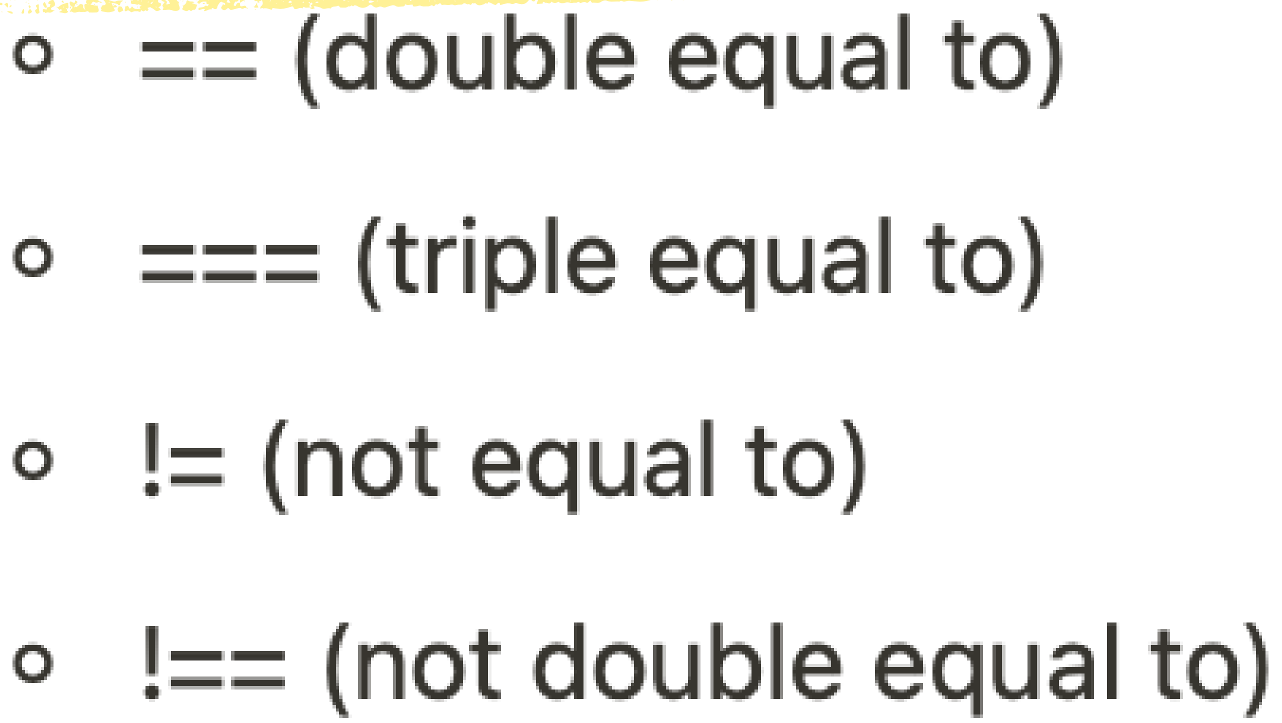
- **Comparison operators** are used to compare two values or expressions and return a Boolean result, indicating whether the comparison is true or false.

1

JS Comparison operators

- **Comparison operators** are used to compare two values or expressions and return a Boolean result, indicating whether the comparison is true or false.

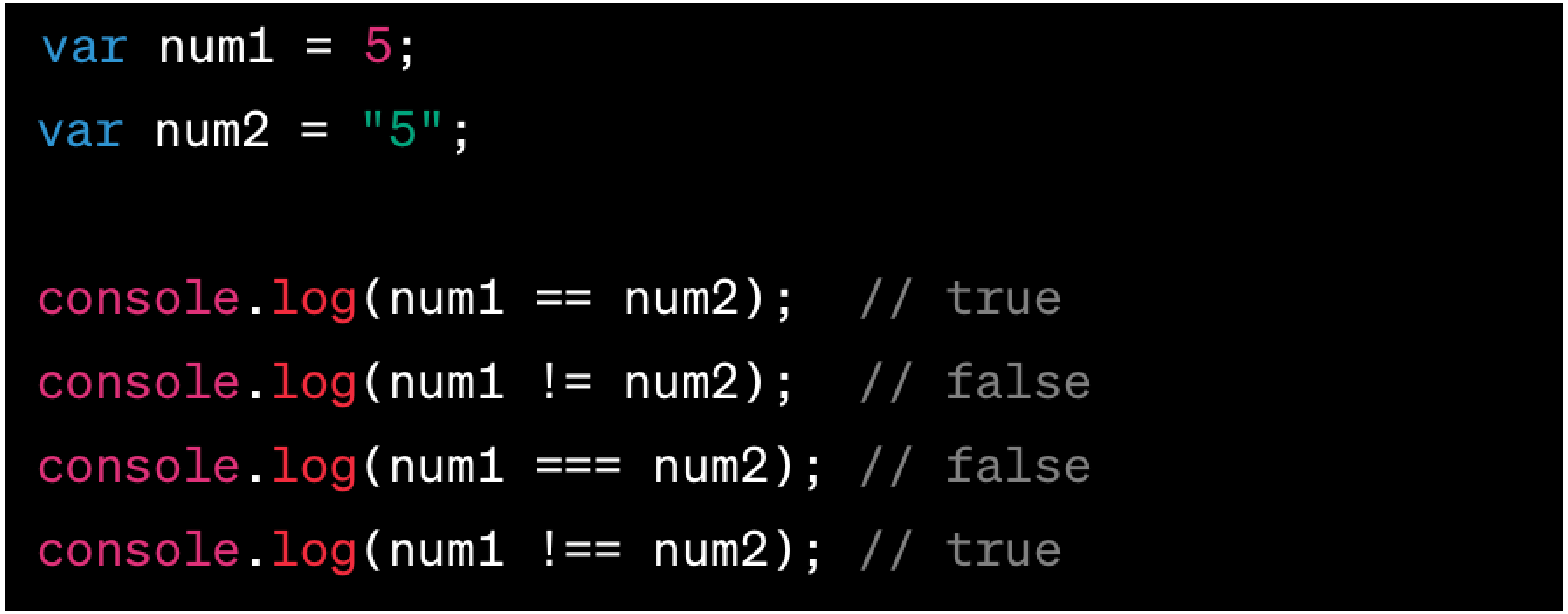
There are **4 Comparison operators**



- == (double equal to)
- === (triple equal to)
- != (not equal to)
- !== (not double equal to)

1 JS Comparison operators

Example



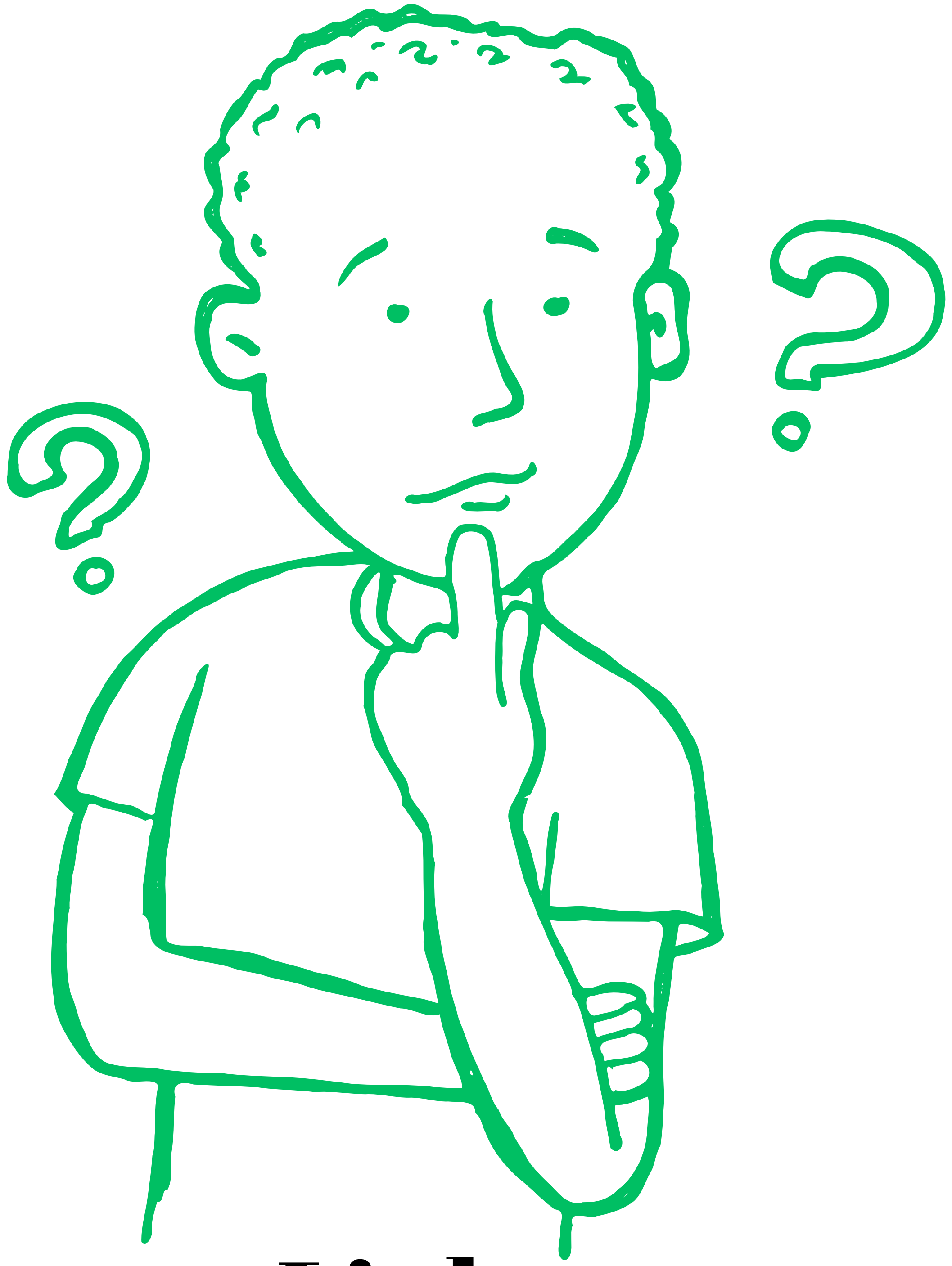
```
var num1 = 5;
var num2 = "5";

console.log(num1 == num2);  // true
console.log(num1 != num2);  // false
console.log(num1 === num2); // false
console.log(num1 !== num2); // true
```

JS Relational VS Comparison operators

Relational Operator	Comparison Operator
Used to establish the r**elationship between two values or expressions, especially for numerical comparisons**.	Used for **comparing two values or expressions.**
Primarily used for **numeric data types**	Can be used for **any data type**.
Do not perform **type coercion**. Values of different data types are not automatically converted.	Perform **type coercion** when comparing values of different data types.
Include **<**, **>**, **<=**, and **>=**.	Include **==**, **!=**, **===**, and **!==**.

Q&A



Link