

[Open in app](#)

Medium



★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



# A Beginner's Guide to Constructors in JavaScript: From Functions to Classes.

3 min read · Jan 25, 2024



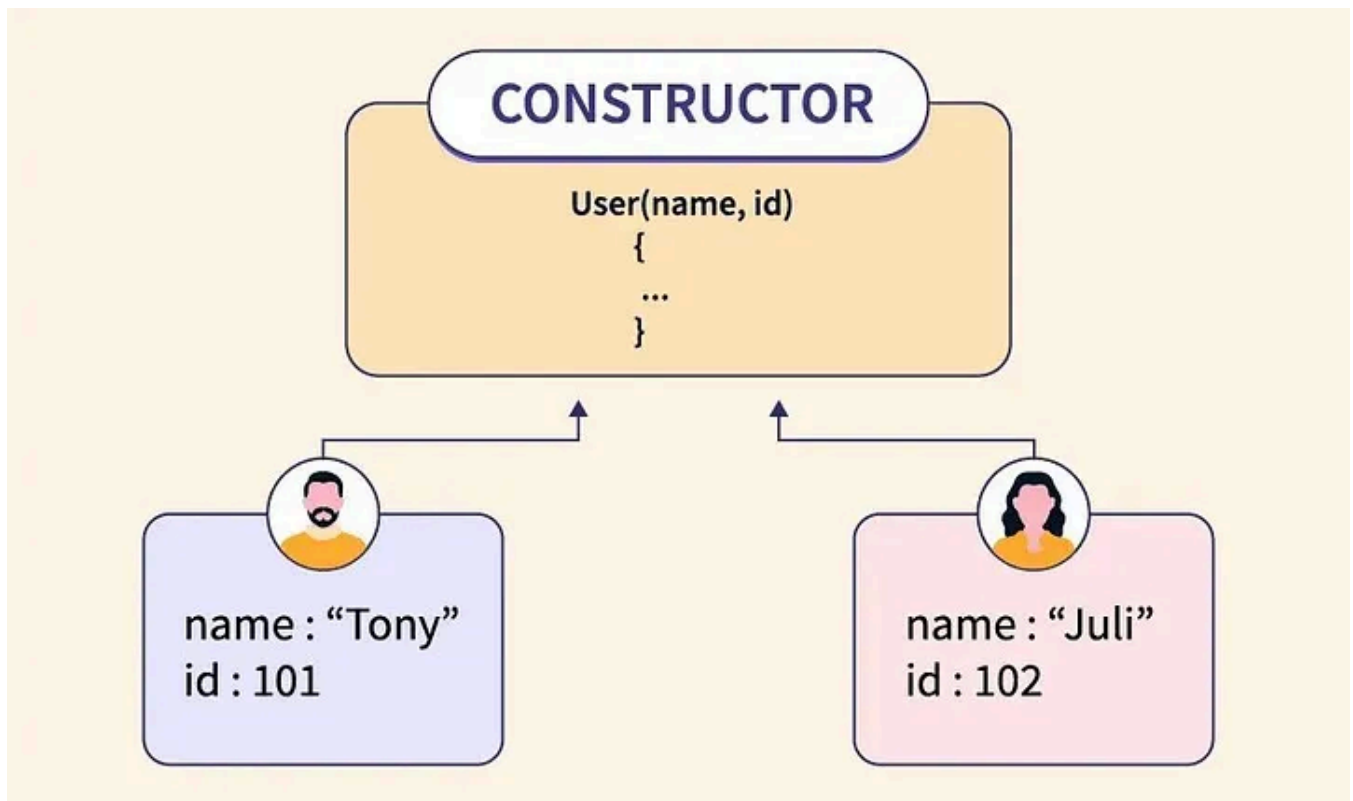
devtalib

[Follow](#)

Listen

Share

More



The term “constructor” in JavaScript can refer to two different things, depending on the context:

## 1. Constructor Functions (Pre-ES6):

When working with JavaScript, especially before ES6 (ECMAScript 2015) introduced the `class` syntax, constructors played a crucial role. They were functions used to

create and set up new objects. Let's dive into what constructors are and how they work.

Constructors are functions designed to create new objects. They define properties and behaviors that will belong to the new object. Think of them as a blueprint for creating multiple objects of the same type.

## The Mechanics of Constructors

### 1. Instantiating New Objects:

Constructors are typically used in conjunction with the `new` keyword. This keyword triggers JavaScript to create a new empty object, set the prototype of this object to the constructor function's prototype, execute the constructor with `this` bound to the new object, and finally return the new object.

### 2. The `this` Keyword in Action:

Inside a constructor, the `this` keyword refers to the new object being created. You can add properties and methods to this new object using `this`. These properties are then assigned to each instance of the object created using the constructor.

```
function Person(name, age) {  
  this.name = name;  
  this.age = age;  
  
  this.describe = function() {  
    return `${this.name} is ${this.age} years old.`;  
  };  
}  
  
const person1 = new Person("Mohi", 25);  
console.log(person1.describe()); // Outputs: "Mohi is 25 years old."
```

Here, `Person` is a constructor function that initializes new objects with `name` and `age` properties and a `describe` method. When we create a new instance of `Person`, these properties and methods are attached to the `person1` object.

## 2. Class Constructors (ES6 onwards):

In JavaScript, with the introduction of ES6 (ECMAScript 2015), the concept of classes was added as a part of its syntax. Within these classes, the `constructor` method became a key player. It's a special method for setting up new objects, much like the

constructor functions used before ES6, but now neatly integrated within the class structure. This was a significant enhancement for developers coming from class-based languages like Java or C++.

- **Definition:**

Class constructors are part of the ES6 class syntax. They are special methods for creating and initializing an object created with a `class`.

- **Syntax:**

A class constructor is defined using the `constructor` keyword inside a class. The `constructor` method can take parameters and is used to set up the properties of the class.

- **Using `new` Keyword:**

Similar to constructor functions, instances of a class are created using the `new` keyword.

```
class Person {
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }

  greet() {
    return `Hello, my name is ${this.name} and I am ${this.age} years old.`
  }
}

const person1 = new Person("Mohi", 25);
console.log(person1.greet()); // "Hello, my name is Mohi and I am 25 years old."
```

In this example, `Person` is our constructor. It creates new person objects with a name, age, and a method to say hello.

### Key Differences:

- Pre-ES6 constructors are regular functions, while class constructors are specific methods within a class.

- Class constructors can access and utilize class features like inheritance through `super` keyword.

### Additional Points:

- Both types of constructors can receive arguments, which are used to initialize the object's properties.
- Constructors can also be used with built-in objects like `Array` and `Object`.

### Examples:

`new Array(3)` creates an array with 3 empty slots. `[ <3 empty items> ]`

`new Array('a', 'b', 'c')` creates an array with elements `'a'`, `'b'`, and `'c'`.  
`['a', 'b', 'c']`

- Not all classes need an explicit constructor; a default constructor is provided if no custom one is defined.

I hope this explanation clarifies the concept of constructors in JavaScript.

[JavaScript](#)[Javascript Constructor](#)[Class](#)[Js](#)[Javascript Tips](#)[Follow](#)

## Written by devtalib

54 followers · 12 following

### No responses yet



sahil siddiqui