

Mastering JavaScript Prototypes

SWIPE TO LEARN

RAJ BHENDADIYA

What is a Prototype?

- A prototype is like a blueprint for objects.
- It defines properties and methods objects can inherit.

The Prototype Chain

- Objects search for properties and methods in a chain.
- If not found, they look up to their prototype.
- This continues until the root object, typically Object.prototype.

Setting Prototypes

- There are various ways to set prototypes.
- You can use constructor functions, `Object.create()`, ES6 classes, or `Object.setPrototypeOf()`.



JS prototype.js

```
function Animal(name) {  
  this.name = name;  
}
```

```
Animal.prototype.makeSound = function () {  
  console.log(`${this.name} makes a sound`);  
};
```

```
const cat = new Animal('Whiskers');  
cat.makeSound(); // Output: Whiskers makes a sound
```

- In this example, we create an object cat using a constructor function Animal.
- The Animal constructor function takes a name parameter and assigns it to the name property of the object.
- We also add a method makeSound to the Animal prototype, which allows every instance created from this constructor to make a sound when called.



JS prototype.js

```
const animalPrototype = {  
  speak() {  
    console.log(`${this.name} makes a sound`);  
  },  
};  
  
const cat = Object.create(animalPrototype);  
cat.name = 'Whiskers';  
  
cat.speak(); // Output: Whiskers makes a sound
```

- In this example, we define a prototype object `animalPrototype`.
- This object has a `speak` method. Then, we create an object `cat` using `Object.create()` and specify `animalPrototype` as its prototype.
- We set the `name` property of `cat` to '`Whiskers`'.
- This way, `cat` inherits the `speak` method from its prototype, allowing it to make a sound.



JS prototype.js

```
function Person(name) {  
    this.name = name;  
}  
  
function Programmer(name, language) {  
    Person.call(this, name);  
    this.language = language;  
}  
  
Programmer.prototype = Object.create(Person.prototype);  
Programmer.prototype.code = function () {  
    console.log(` ${this.name} is coding in ${this.language}`);  
};  
  
const coder = new Programmer('Alice', 'JavaScript');  
coder.code(); // Output: Alice is coding in JavaScript
```

- Person takes a name parameter and assigns it to the name property.
- Programmer is derived from Person.
- It uses Person.call() to inherit the name property and introduces its own property, language.
- We create a method code for Programmer objects by adding it to the Programmer prototype.

I'm Raj.

Software Engineer;
passionate for building and shipping
scalable software.

I am open to full-stack development
projects. (JavaScript, Python, AI/ML,
DevOps)

Let's connect and expand our
professional network together.

I'm excited to collaborate, exchange
ideas, and contribute to impactful
projects.