Satyaansh Softech

Project Based Learning

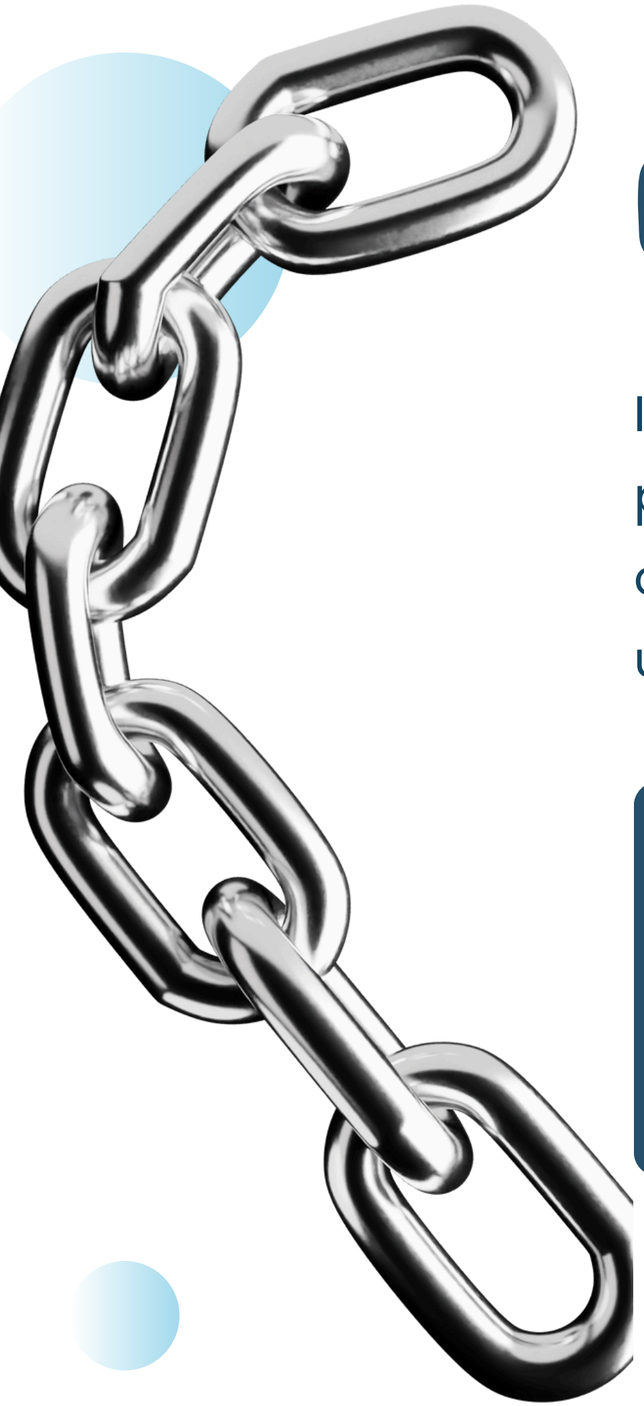
- NEW Something new for everyone — beginner or pro
- 100% Project-Based Learning
- 24/7 Every video = Contains assignments

www.satyaanshsoftech.com

Satyaansh Softech Pvt. Ltd.

Optional Chaining

By Satyawan Panchal



Optional Chaining

Introduction in **ECMAScript 2020**, is a feature that simplifies the process of accessing properties and methods of nested objects or arrays when intermediate properties may be null or undefined.

The optional chaining operator
→ (?.)
It allows you to access properties.

Outline

01 Objects

02 Ways to create

03 Object Methods

04 Uses of objects

Along with examples

WAYS TO CREATE OBJECTS

Object literal

```
const obj = { key: "value" };
```

new Object()

```
const obj = new Object();
obj.key = "value";
```

WAYS TO CREATE OBJECTS

Constructor function

```
function Person(name) {
 this.name = name;
}
const p1 = new Person("Riya")
```

Class

(syntactic sugar)

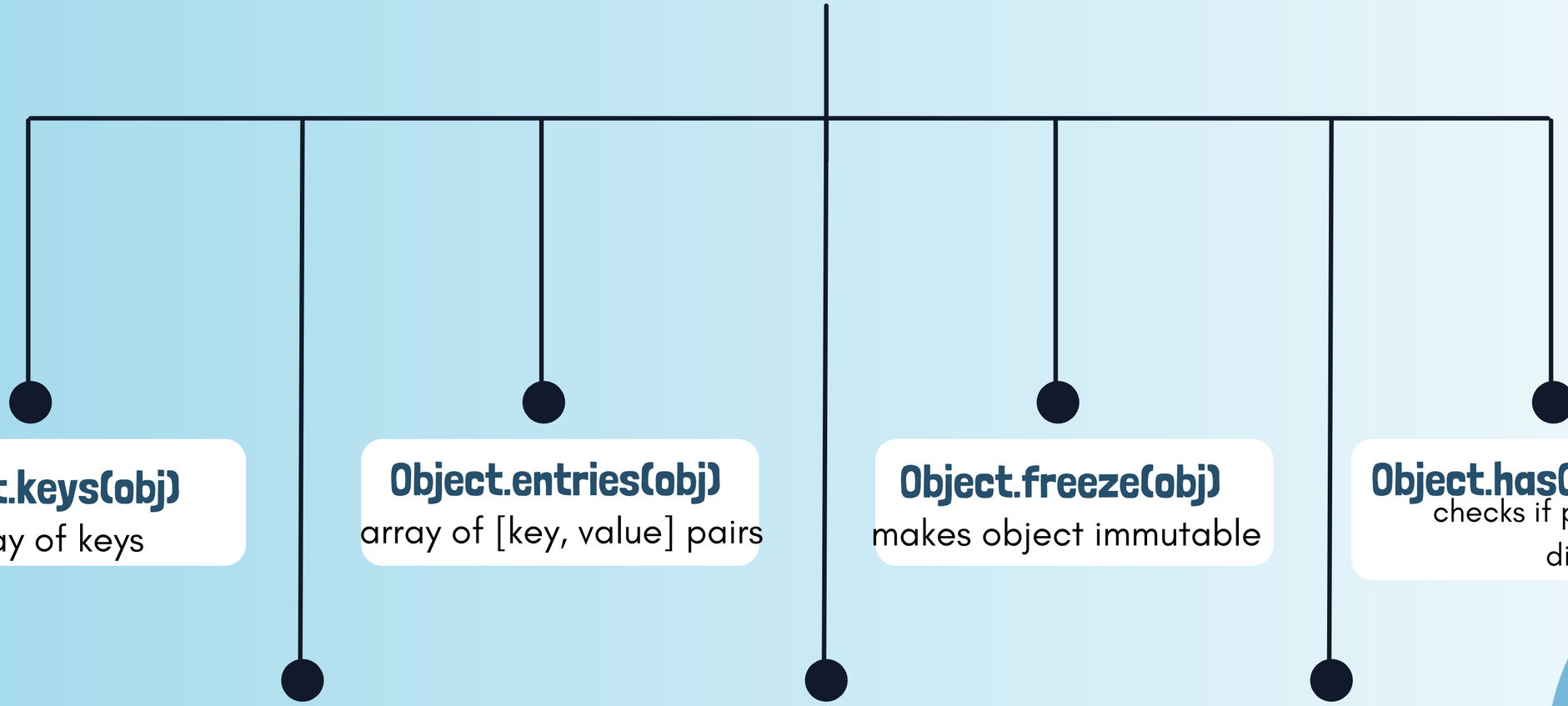
```
class Person {
constructor(name) {
        this.name = name;
    }
 }
```

WAYS TO CREATE OBJECTS

Object.create()

```
const proto = {
greet() {
        return "Hello";
    }
};
const obj = Object.create(proto);
```

OBJECT METHODS



Object.keys(obj)
array of keys

Object.values(obj)
array of values

Object.entries(obj)
array of [key, value] pairs

Object.assign(target, source)
copies properties

Object.freeze(obj)
makes object immutable

Object.seal(obj)
prevents adding/removing properties but allows modification

Object.hasOwn(obj, prop)
checks if property exists directly

Uses of Objects

1. Store structured data (person, product, settings, etc.).
2. Group related functionality (methods inside).
3. Represent real-world entities (students, employees, etc.).
4. Use as a map/dictionary (key–value storage).
5. Backbone of OOP in JavaScript.

Misconceptions

1. ❌ Objects and arrays are different

   ✅ Arrays are also objects with numeric keys.

2. ❌ Objects are unordered.

   ✅ Since ES2015, object keys maintain order (integers sorted, strings insertion order).

Misconceptions

3. ❌ this always refers to object itself

   ✅ It depends on how function is called.

4. ❌ Objects can only have string keys

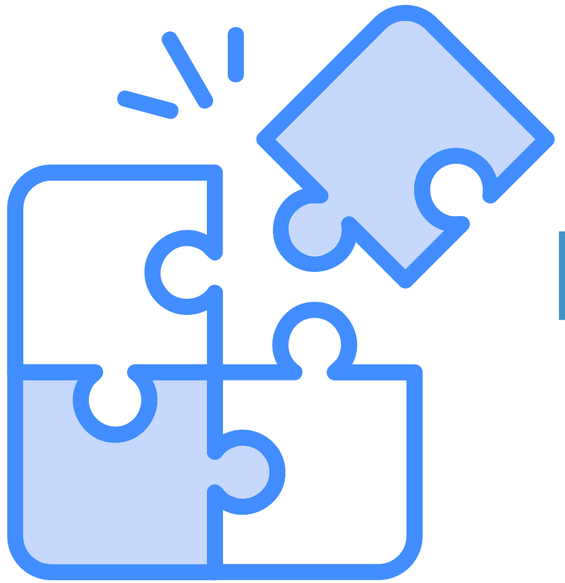
   ✅ Keys can also be symbols.

FACTS

1. Object.freeze() is shallow → nested objects are still mutable
2. Property names are always strings or symbols → numbers are converted to strings
3. You can make objects without prototypes (Object.create(null)).
4. Functions in JS are also objects → they can have properties

Common Mistakes

Wrong	Right
person = { name: "Kanak" };	const person = { name: "Kanak" };
const obj2 = obj1;	const obj2 = { ...obj1 }; OR const obj2 = structuredClone(obj1);
obj.name = "Varsha"; // when frozen	const newObj = { ...obj, name: "Varsha" };
for (let key in obj) { ... }	Object.keys(obj).forEach(key => {...});
greet: () => console.log(this.name)	greet() { console.log(this.name); }
const copy = { ...obj }; // shallow	const copy = structuredClone(obj);
const obj = { toString: "Hello" };	const obj = Object.create(null); obj.name = "Kanak";
user.profile.name	user?.profile?.name
{ a: 1 } === { a: 1 } // false	JSON.stringify(obj1) === JSON.stringify(obj2)



HANDS-ON PRACTICAL

QUESTIONS FOR YOU

01 How are arrays also objects in JavaScript?

02 Create an object car with properties brand, model, and year. Add a method getDetails() that returns a string describing the car.

03 Why does this fail? How can it be fixed?

```
const person = {
  name: "Kanak",
  greet: () => console.log(this.name)
};
person.greet();
```

04 Use Object.keys(), Object.values(), and Object.entries() on the object below and write the outputs:

```
const student = { name: "Riya", age: 20, grade: "A" };
```

PROVIDING WINGS TO YOUR DREAMS

FOR ONLINE/OFFLINE SOFTWARE TRAINING

+91 70822 52235

www.satyaanshsoftech.com

satyaanshsofttechpvtltd@gmail.com

132101, Panipat, India