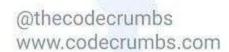


JavaScript Object Essentials







Property Access

Dot Notation: The most common way to access properties. If you have an object person with a property name, you can access it with **person.name**

Bracket Notation: Useful when property names are dynamic or include characters that can't be used in dot notation.

Example: person["first name"]



Object Methods

Objects can store functions as properties. These functions are typically called methods when associated with objects.

Within an object method, **this** refers to the object that the method belongs to. However, the value of **this** can change depending on the context in which a function is called.

```
const person = {
  name: 'John',
  greet: function () {
    console.log('Hello, ' + this.name + '!');
  },
};

person.greet();
// Outputs: Hello, John!
```



Object Keys

Object.keys() Returns an array of an object's own enumerable property names.

```
const car = { make: 'Toyota', model: 'Camry' };
console.log(Object.keys(car));
// Outputs: ["make", "model"]
```



Object Values

Object.values() Returns an array of an object's own enumerable property values.

```
const car = { make: 'Toyota', model: 'Camry' };
console.log(Object.values(car));
// Outputs: ["Toyota", "Camry"]
```



Object Entries

Object.entries() Returns an array of [key, value] pairs for an object.

```
const car = { make: 'Toyota', model: 'Camry' };
console.log(Object.entries(car));
// Outputs: [["make", "Toyota"], ["model", "Camry"]]
```



Cloning Objects

Copying objects in JavaScript can be tricky since objects are reference types. To create a shallow copy, you can use the **Object.assign()** method or the spread operator.

```
const carCopy = Object.assign({}, car);

// OR

const carCopy2 = { ...car };
```



Nested Objects

Objects can contain other objects, leading to a hierarchical structure. Accessing nested objects requires chaining property access.

```
const student = {
  name: 'Alex',
  address: {
    street: '123 Main St',
    city: 'Anytown'
  },
};

console.log(student.address.city);
// Outputs: Anytown
```

How many do you know?

@thecodecrumbs www.codecrumbs.com