

II. Thuộc Tính và Phương Thức của Object

Truy cập thuộc tính:

Dấu chấm (.)

Dấu chấm (.) là cách phổ biến nhất để truy cập thuộc tính của một Object.

```
const person = { name: "John", age: 30 };  
console.log(person.name); // John
```

Dấu ngoặc vuông ([])

Dấu ngoặc vuông ([]) cho phép truy cập thuộc tính động hoặc tên thuộc tính có khoảng trắng, ký tự đặc biệt.

```
const person = { "full name": "John Doe", age: 30 };  
console.log(person["full name"]); // John Doe
```

Thêm, sửa và xóa thuộc tính

- Thêm thuộc tính:

```
const person = {};  
person.name = "Alice";  
console.log(person); // { name: "Alice" }
```

- Sửa thuộc tính:

```
person.name = "Bob";  
console.log(person.name); // Bob
```

- Xóa thuộc tính:

```
delete person.name;  
console.log(person); // {}
```

Enhanced Object Literal

```
const name = "Alice";
const age = 30;
const person = { name, age };
console.log(person); // { name: "Alice", age: 30 }
```

Thuộc tính động (Computed Property)

Computed Property giúp tạo thuộc tính từ biến.

```
const key = "email";
const user = { [key]: "user@example.com" };
console.log(user.email); // user@example.com
```

Định nghĩa phương thức trong Object

Object có thể chứa các phương thức (hàm bên trong Object).

```
const person = {
  name: "John",
  greet() {
    console.log(`Hello, my name is ${this.name}`);
  }
};
person.greet(); // Hello, my name is John
```

this trong Object – Cách hoạt động và các vấn đề thường gặp

- Sử dụng **this** trong Object:

```
const person = {
  name: "Alice",
  sayHello() {
    return `Hello, I am ${this.name}`;
  }
};
console.log(person.sayHello()); // Hello, I am Alice
```

- Vấn đề thường gặp khi dùng **this**:

```
const person = {
  name: "John",
  greet() {
    setTimeout(function() {
      console.log(`Hello, ${this.name}`); // undefined
    }, 1000);
  }
};
```

```
    }, 1000);  
  }  
};  
person.greet();
```

Giải pháp sử dụng arrow function:

```
const person = {  
  name: "John",  
  greet() {  
    setTimeout(() => {  
      console.log(`Hello, ${this.name}`); // Hello, John  
    }, 1000);  
  }  
};  
person.greet();
```