

 Cập nhật tháng 8 năm 2024

[Bài Đọc] Truy cập vào thuộc tính trong đối tượng

Thuộc tính của đối tượng trong JavaScript có thể được truy cập và thao tác bằng hai cách chính: **dấu chấm (.)** và **dấu ngoặc vuông ([])**. Mỗi cách có ưu điểm riêng và phù hợp với từng trường hợp cụ thể.

1. Truy cập thuộc tính bằng dấu chấm (.)

- Cách này là trực quan và dễ đọc nhất khi tên thuộc tính đã biết trước và tuân theo quy tắc đặt tên hợp lệ của JavaScript.
- Cú pháp

`object.property`

Ví dụ:

```
const person = {  
  name: "Alice",  
  age: 25,  
  greet: function() {  
    console.log(...data: "Hello!");  
  },  
};  
  
console.log(...data: person.name); // Alice  
console.log(...data: person.age);  // 25  
person.greet();                     // Hello!
```

- Lưu ý
 - Tên thuộc tính phải là một **identifier hợp lệ** trong JavaScript (không chứa dấu cách, không bắt đầu bằng số, không chứa ký tự đặc biệt như - hoặc @).
 - Không sử dụng được nếu thuộc tính là một biến hoặc có ký tự đặc biệt trong tên.

2. Truy cập thuộc tính bằng dấu ngoặc vuông ([])

- Cách này linh hoạt hơn và cho phép truy cập các thuộc tính mà tên của chúng có ký tự đặc biệt, chứa dấu cách, hoặc không biết trước.
- Cú pháp

`object["propertyName"]`

Ví dụ:

```
const person = {
  'full name': 'Alice Johnson',
  age: 25,
  country: 'USA',
};

console.log(...data: person['full name']); // Alice Johnson
console.log(...data: person['age']); // 25
console.log(...data: person['country']); // USA
```

- Truy cập thuộc tính động
 - Khi tên thuộc tính được lưu trữ trong một biến, sử dụng dấu ngoặc vuông là cách duy nhất.

```
const person = {
  'full name': 'Alice Johnson',
  age: 25,
  country: 'USA',
};

console.log(...data: person['full name']); // Alice Johnson
console.log(...data: person['age']); // 25
console.log(...data: person['country']); // USA

const property = "age";
console.log(...data: person[property]); // 25
```

- Lưu ý
 - Dấu ngoặc vuông yêu cầu tên thuộc tính được cung cấp dưới dạng chuỗi (string) hoặc biểu thức trả về chuỗi.
 - Phù hợp khi làm việc với dữ liệu JSON hoặc các đối tượng phức tạp.

3. Xử lý khi thuộc tính không tồn tại

- Khi truy cập một thuộc tính không tồn tại trong đối tượng, JavaScript trả về **undefined** thay vì lỗi

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

4. Xử lý an toàn khi truy cập thuộc tính (Optional Chaining)

- JavaScript cung cấp cú pháp ?. để tránh lỗi khi truy cập thuộc tính trên đối tượng **null** hoặc **undefined**

```
const person = null;  
console.log(...data: person?.name); // undefined, không gây lỗi
```

5. Tài liệu tham khảo

<https://dmitripavlutin.com/access-object-properties-javascript/>

<https://javascript.info/optional-chaining>

Danh sách các bài học

