

Bài: Kiểu dữ liệu Symbol trong JavaScript

Xem bài học trên website để ủng hộ Kteam: [Kiểu dữ liệu Symbol trong JavaScript](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở bài trước, Kteam đã giới thiệu đến các bạn về kiểu dữ liệu boolean trong JavaScript.

Trong bài này, chúng ta sẽ cùng nhau tìm hiểu về một kiểu dữ liệu khác: **symbol**

Nội dung

Để nắm được nội dung của bài này, bạn cần có kiến thức về:

- Kiểu dữ liệu số trong JavaScript
- Kiểu dữ liệu chuỗi trong JavaScript

Những nội dung mà chúng ta sẽ cùng tìm hiểu:

- Giới thiệu khái quát về Symbol trong JavaScript, tác dụng của Symbol
- Các thuộc tính, phương thức của Symbol

Kiểu dữ liệu symbol trong Js

Nếu bạn dùng google và dịch từ "**symbol**" sang tiếng Việt, thì nó có nghĩa là ký hiệu, hay là vật tượng trưng.

Ở một khía cạnh nào đó, ta cũng có thể hiểu symbol trong JavaScript theo nghĩa tương tự.

Trong JavaScript, symbol là một trong 7 loại kiểu dữ liệu "nguyên thủy" (tiếng Anh: **primitive**). 7 loại này bao gồm số (**number**, **bigint**), chuỗi (**string**), **boolean**, **undefined**, **symbol** và **null**.

Cấu trúc của symbol:

Một symbol sẽ bao gồm một key được đặt trong cụm từ "symbol":

```
symbol(<key>)
```

- **<key>** có thể là một số, một chuỗi hay bất kì kiểu dữ liệu nào khác, miễn là nó hợp lệ

Khởi tạo một symbol

Để khởi tạo một giá trị kiểu symbol, ta dùng constructor Symbol:

```
Symbol(<value>)
```

Trong đó: **<value>** có thể có hoặc không. Nhưng nếu có, **<value>** phải là một giá trị hợp lệ, và nó có thể mang bất kỳ kiểu dữ liệu nào. Ở ví dụ bên dưới có sử dụng function – một kiến thức mới mẻ. Có thể hiểu đơn giản, là nó chứa 1 dòng lệnh bên trong dùng để tái sử dụng. **Kteam** sẽ nói rõ hơn về function trong các bài tiếp theo.

Ví dụ:**Javascript:**

```
a = Symbol('a')
// Symbol(a)
b = Symbol('1')
// Symbol(1)
t = Symbol(true)
// Symbol(true)
function f() { return 1;}
// Undefined
Symbol(f)
// Symbol(function f() { return 1;})
```

Sau khi khởi tạo, các giá trị này đều thuộc kiểu symbol:

Javascript:

```
typeof(a)
// 'symbol'
typeof(b)
// 'symbol'
typeof(t)
// 'symbol'
```

Đặc điểm của symbol

Tính độc nhất

Đối với kiểu dữ liệu symbol, thì mỗi giá trị là **độc nhất** trong một chương trình, có nghĩa là, **không thể tồn tại hai giá trị symbol bằng nhau**, ngoại trừ trường hợp ta dùng phép **gán**.

Javascript:

```
a = Symbol()
// Symbol()
b = Symbol()
// Symbol()
a == b
// false
```

Mời các bạn tiếp tục xem các ví dụ bên dưới:

Javascript:

```
a = Symbol(123)
// Symbol(123)
b = Symbol(123)
// Symbol(123)
a == b
// false
```

Javascript:

```
a = Symbol(true)
// Symbol(true)
b = Symbol(true)
// Symbol(true)
a == b
// false
```

Các giá trị symbol khi ép sang các kiểu dữ liệu khác:

Kiểu dữ liệu tương ứng	Giá trị	Ví dụ
Boolean	True	<code>Boolean(Symbol())</code> <code>//true</code>
Number	Không thể chuyển từ symbol sang number	<code>Number(Symbol(123))</code> <code>// Uncaught TypeError:</code> <code>Cannot convert a Symbol</code> <code>value to a number</code> <code>at Number (<anonymous>)</code>
String	Chuỗi tương ứng	<code>String(Symbol())</code> <code>// 'Symbol()'</code>
Array	Một array bao gồm các symbol	<code>Array(Symbol(123))</code> <code>// [Symbol(123)]</code> <code>Array(Symbol(123),</code> <code>Symbol('Kteam'))</code> <code>// [Symbol(123),</code> <code>Symbol(Kteam)]</code>

Sử dụng symbol như là một thuộc tính của object

Vì tính độc nhất của symbol, nên nó có thể được dùng làm thuộc tính của một object.

Javascript:

```
sb = Symbol('aa')  
// Symbol(aa)  
obj = {  
  Name: "Kter",  
  Age: 18,  
  [sb]: "Dodoodo"  
}  
// { Name: 'Kter', Age: 18, [Symbol(aa)]: 'Dodoodo' }  
console.log(obj[sb]);  
// Dodoodo  
// undefined
```

Kết luận

Qua bài này, các bạn đã biết về kiểu dữ liệu symbol trong JavaScript

Bài sau, Kteam sẽ cùng bạn củng cố qua [Bài tập kiểu dữ liệu symbol trong JavaScript](#)

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.