Bài: Các giá trị Null và Undefined trong JavaScript

Xem bài học trên website để ủng hộ Kteam: Các giá trị Null và Undefined 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, các bạn đã được tìm hiểu về kiểu dữ liệu symbol trong JavaScript

Trong bài này, chúng ta sẽ cùng tìm hiểu về 2 loại giá trị: **null và undefined**

Nội dung

Dưới đây là những yêu cầu trước khi bắt đầu tìm hiểu bài này:

- Các kiến thức về biến trong JavaScript

Những nội dung mà chúng ta sẽ cùng tìm hiểu

- Giá trị Null trong JavaScript
- Giá trị undefined trong JavaScript
- So sánh giữa 2 kiểu dữ liệu trên

Giá trị null trong JavaScript

Giá trị null là một kiểu dữ liệu **nguyên thủy** trong JavaScript, và đơn giản là: nó đại diện cho việc một biến mà không trỏ đến bất kì giá trị nào.

Ở mức độ cơ bản, chúng ta không cần phải hiểu quá sâu về null. Dưới đây là các giá trị của null khi được ép sang các kiểu dữ liệu khác:

Javascript:

```
Boolean(null) // null là một falsy
// false
String(null)
// 'null'
Number(null)
// 0
Symbol(null)
// Symbol(null)
Object(null)
// {}
```

Và, hãy cùng kiểm tra kiểu dữ liệu của null:

Javascript:

```
typeof(null)
// 'object'
```

Ơ kìa ? Vô lí.

Rõ ràng Kteam đã nói rằng null là một kiểu dữ liệu nguyên thủy cơ mà ? Thế nào lại có 'object' ở đây ?

Về mặt bản chất, null được tạo ra để giúp JavaScript tương tác với một loại ngôn ngữ khác, đó chính là Java. Chính việc cố gắng làm cho nó tương thích, đã tạo ra một lỗi trong quá trình triển khai, đó là việc làm cho **typeof(null)** bằng với object.

Bài viết dưới đây (từ stackoverflow) sẽ cho chúng ta cái nhìn rõ hơn về null: Why is null an object and what's the difference between null and undefined?.

Kteam sẽ không đi quá sâu về những gì được đề cập trong bài viết này, vì phương châm của khóa học là tiếp cận những bạn học ở mức cơ bản.

Các toán tử cũng có thể được sử dụng với null, như trong ví dụ dưới đây:

Javascript:

```
4 + null
// 4
null + false
// 0
null + true
// 1
null === null
// true
undefined == null
// true
```

Giá trị undefined trong JavaScript

> **Lưu ý:** Giá trị undefined trong JavaScript khác với giá trị undefined trong toán học.

Cũng giống như *null*, undefined là một giá trị được "đặc cách" là một trong những kiểu dữ liệu "nguyên thủy" trong JavaScript.

Khi nào thì xuất hiện giá trị undefined?

> **Lời tác giả:** Những kiến thức được đề cập dưới đây hầu hết là những thứ mới, do đó các bạn chỉ cần biết là nó sẽ xuất hiện trong một vài trường hợp, thế là đủ.

Trong JavaScript, undefined xuất hiện trong các trường hợp sau:

Khởi tạo một biến không có giá trị ban đầu, hoặc gán giá trị undefined vào cho nó

- **Ví dụ:**

Javascript:

```
var t
// undefined
t
// undefined
var f = undefined
// undefined
f
// undefined
```

Cố gắng truy cập vào một thuộc tính không có sẵn trong một Object

Khi truy cập một thuộc tính không có sẵn trong một Object, thì giá trị của việc truy cập đó là undefined.

Javascript:

```
a = {name: "Kteam"} // Khởi tạo một object a với một thuộc tính là name
// { name: 'Kteam' }
a.name
// 'Kteam'
a.kteam
// undefined
```

Undefined là giá trị mặc định của các tham số trong một hàm

Đối với một hàm có tham số đầu vào, nếu ta không cho nó một giá trị mặc định, thì giá trị của nó trước khi các đối số được truyền vào là undefined.

Giả sử, ta có một hàm **print** với 2 tham số là a và b, cụ thể như sau:

Javascript:

```
function print(a, b) {
   console.log(a);
   console.log(b);
}
// undefined
```

Sau đó, với hàm **print** như trên, ta tiến hạnh gọi **print(3)** (chỉ truyền vào một đối số):

Javascript:

```
print(3)
// 3
// undefined
// undefined
```

- **Kết quả:** nó sẽ in ra 3 dòng như trong ví dụ. Ở đây, ta chỉ xét 2 dòng đầu, vì dòng thứ 3 là một trường hợp khác (Kteam sẽ nhắc đến trong phần tiếp theo).
  2 dòng đầu chính là 2 giá trị của các tham số a và b khi chạy hàm. Ta có thể thấy, khi chỉ có một đối số được truyền vào, thì giá trị của tham số còn lại là undefined.

Ta có thể chứng thực bằng một ví dụ khác. Cũng với hàm **print** kia, nhưng lần này, ta sẽ không truyền vào tham số:

Javascript:

```
print();
// undefined
// undefined
// undefined
```

Khi một hàm được định nghĩa mà không có giá trị trả về

Như Kteam đã đề cập trong ví dụ trên, bạn sẽ thấy 3 dòng, với nội dung dòng cuối là **undefined**.

Đây là trường hợp mà bạn không cho hàm giá trị trả về.

Đối với hàm, nếu có lệnh **return** và theo sau đó là một giá trị, thì giá trị đó chính là giá trị trả về của hàm.

Còn nếu không có lệnh return hoặc lệnh return mà không đi kèm theo giá trị, thì giá trị trả về của hàm là **undefined**.

> **Lưu ý:** Đối với ví dụ bên trên, chỉ khi ta dùng node.js thì mới có thể thấy được 3 dòng (với dòng cuối là giá trị trả về của hàm).

Ví dụ:

Javascript:

```
function print(a, b) { // định nghĩa một hàm
}
// Undefined // giá trị được định nghĩa
print(2, 3); // Gọi hàm và truyền đối số cho hàm
// Undefined // giá trị trả về của hàm
```

Khi thực hiện các phép toán với giá trị undefined, kết quả sẽ là **NaN**.

Sự khác biệt giữa null và undefined trong JavaScript

Nhìn thì khá giống nhau, nhưng null và undefined cũng có điểm khác nhau.

Về mặt ý nghĩa, đối với một biến:

- Khi chúng ta xác định một biến thành *undefined* thì chúng ta đang cố gắng truyền đạt rằng biến đó không tồn tại.
- Khi chúng ta xác định một biến thành *null* thì chúng ta đang cố gắng truyền đạt rằng biến đó trống.

null là object (nó phải là null, đây là một lỗi nhỏ trong JavaScript), còn **undefined** là một giá trị duy nhất:

Javascript:

```
function print(a, b) { // định nghĩa một hàm
}
// Undefined // giá trị được định nghĩa
print(2, 3); // Gọi hàm và truyền đối số cho hàm
// Undefined // giá trị trả về của hàm
```

Chúng có cùng giá trị, nhưng khác kiểu với nhau:

Javascript:

```
null == undefined
// true
null === undefined
// False
```

Khi chuyển về number, null và undefined cũng cho 2 kết quả khác biệt:

Javascript:

```
Number(null)
// 0
Number(undefined)
// NaN
```

Kết luận

Qua bài này, các bạn đã nắm được kiến thức về null và undefined trong JavaScript

Bài tiếp theo, Kteam sẽ giới thiệu với các bạn về Bài tập về các giá trị Null và Undefined 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**".