Bài: Kiểu dữ liệu chuỗi trong JavaScript (Phần 2) - Template literals trong Js, Mối liên hệ giữa Number-String và Number

Xem bài học trên website để ủng hộ Kteam: Kiểu dữ liệu chuỗi trong JavaScript (Phần 2) - Template literals trong Js, Mối liên hệ giữa Number-String và Number

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 cho các bạn về kiểu dữ liệu chuỗi trong Javascript

Ở bài này, các bạn sẽ được mở rộng hơn kiến thức của mình về chuỗi: **Template literals trong Javascript, Mối liên hệ giữa Number-String và số**

Nội dung

Để nắm chắc kiến thức bài này, bạn cần có:

- Kiến thức cơ bản về chuỗi trong Javascript
- Các toán tử cơ bản với số trong Javascript

Những nội dung mà Kteam sẽ đề cập đến:

- Template literals trong Javascript
- Mối liên hệ giữa số và chuỗi trong Javascript

Template literals trong JavaScript

Ở bài trước, Kteam vẫn còn nợ bạn một câu trả lời, về cách dùng cả dấu " và ' trong cùng một chuỗi mà không gây lỗi.

Ở bài này, Kteam sẽ "trả bài" cho các bạn. Cụ thể hơn chúng ta cùng đi vào nội dung sau

Template literals, là một tính năng mới trong **ES6** (JavaScript 2015), Template literals (Còn được gọi là template string) là những chuỗi mà được đặt trong cặp `` (**backtick**).

Về mặt cơ bản, nó không khác so với những chuỗi thông thường, điều đặc biệt chỉ là việc cho phép sử dụng cả dấu ' và " trong cùng một chuỗi. Đó là lý do tại sao nó lại có thể được dùng nếu trong chuỗi yêu cầu nhiều các dấu ' và ".

Ví dụ:

Javascript:

```
a = `how'Kteam"`
// `how'Kteam"`
b = `doooooo`
// 'doooooo'
typeof(a)
// 'string'
typeof(b)
// 'string'
```

Nhưng không có gì sinh ra chỉ để cho có cả. Ngoài những điều cơ bản, nó còn cho phép nhúng một biểu thức, một hàm hay một giá trị nào đó vào trong một chuỗi:

Copyright © Howkteam.com

Ví dụ:

Javascript:

```
a = "hiii ${1+2}" // Dùng chuỗi thông thường
// 'hiii ${1+2}'
a = `hiii ${1+2}` // Dùng template string
// 'hiii 3'
a = `hiii ${[1, 2, 3] + 1}` // Mảng + number   string
// 'hiii 1,2,31'
function a() {return 12} // Định nghĩa một hàm
// Undefined
t = `hiii ${a()}` // dùng template string kết hợp hàm
// 'hiii 12'
```

Bên cạnh đó, việc sử dụng cặp `` còn cho phép bạn khởi tạo một chuỗi trên nhiều dòng. Thoạt tiên, hãy lấy ví dụ với những loại chuỗi thông thường:

Javascript:

```
a = 'howKteam // nhấn enter
// a = 'howKteam
    ^^^^^^^^^

// Uncaught SyntaxError: Invalid or unexpected token
a = "howKteam
// a = "howKteam
    ^^^^^^^^^

// Uncaught SyntaxError: Invalid or unexpected token
```

Một điều có thể thấy rõ ràng là chuỗi thông thường không cho phép việc khởi tạo một chuỗi trên nhiều dòng mà không sử dụng **escape character**.

Tất nhiên, để khởi tạo, đơn giản là thêm **escape character** vào rồi.

Javascript:

```
a = 'howKteam\nFree education'
// 'howKteam\nFree education'
a
// 'howKteam\nFree education'
console.log(a)
// howKteam
// Free education
// undefined
a = 'howKteam\n' + 'Free education'
// 'howKteam\nFree education'
console.log(a)
// howKteam
// Free education
// undefined
```

Nhưng, để cho việc khởi tạo được tường minh hơn, thì còn một cách khác để thực hiện, đó là dùng cặp ``:

Javascript:

```
b = `howKteam
// Free education`
// 'howKteam\nFree education'
console.log(b)
// howKteam
// Free education
// undefined
```

Copyright © Howkteam.com

Chắc chắn rồi, giá tri của 2 chuỗi được khởi tạo bằng 2 cách là tương tự nhau:

Javascript:

```
a == b
// true
a
// 'howKteam\nFree education'
b
// 'howKteam\nFree education'
```

Mối liên hệ giữa chuỗi và số trong JavaScript

Như đã quy ước, ta sẽ gọi các chuỗi chỉ gồm các ký tự số là **Number-String**. Vì Number-String là một chủ đề khá thú vị, cần được tìm hiểu một cách độc lập.

Đối với Number-String, Javascript sẽ cho phép nó thực hiện các toán tử giống như kiểu dữ liệu số (ngoại trừ **phép cộng** sẽ nối các chuỗi với nhau):

Ví dụ:

Javascript:

```
var a = '100';
// undefined
var b = '200';
// undefined
a - b;
// -100
a + b;
// '100200'
a * b;
// 20000
a / b;
// 0.5
a % b;
// 100
```

Và tất nhiên là các toán tử giữa Number-String và số đều được cho phép:

Javascript:

```
var a = '100';
// undefined
a == 100;
// True
a === 100; // Trả về false vì ban đầu, a có kiểu là string
// False
a * 100;
// 10000
'10000' / 2.5
// 4000
```

> **Lưu ý:** Khi thực hiện cộng một Number-String với một số, thì JS sẽ chuyển toàn bộ các số về dạng chuỗi và sau đó nối với nhau.

Javascript:

```
12345 + '6789';
// '123456789'
'6789' + 12345
// '678912345'
```

Copyright © Howkteam.com

Math đối với một Number-String:

Javascript:

```
Math.cbrt("182")
// 5.667051108097064
Math.sin("82")
// 0.31322878243308516
Math.round("1.82")
// 2
```

Kết luận

Qua bài này, các bạn đã hiểu rõ hơn về "template literals" và cách sử dụng chuỗi giống như một số trong Js

Bài tiếp theo, chúng ta sẽ cùng tìm hiểu Các phương thức của chuỗi.

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**".

Copyright © Howkteam.com