**Toán tử logical**

&& : sẽ bỏ qua các giá trị true, nhận giá trị false đầu tiên (tất cả các giá trị true thì nhận thằng cuối cùng)

|| : sẽ bỏ qua các giá trị false, nhận giá trị true đầu tiên (tất cả các giá trị false thì)

Ví dụ:

**Text

Description automatically generated**

**Truthy - to bool is true**

Bất cứ giá trị nào trong Javascript khi chuyển đổi sang kiểu dữ liệu boolean mà có giá trị true thì ta gọi giá trị đó là Truthy.

Các giá trị 1, ['BMW'], { name: 'Miu' } và 'hi' được đề cập trong ví dụ dưới đây là Truthy vì khi chuyển sang Boolean ta nhận được giá trị true.

Ví dụ:

console.log(Boolean(1)) // true

console.log(Boolean(['BMW'])) // true

console.log(Boolean({ name: 'Miu' })) // true

console.log(!!'hi') // true

!! là gì? Đơn giản thôi. Toán tử ! là toán tử not (phủ định) nên !! là 2 lần phủ định, mà 2 lần phủ định lại trở thành “khẳng định”. Trong Javascript thì đây là một “tip” để convert (chuyển đổi) mọi kiểu dữ liệu khác sang Boolean.

Ví dụ:

console.log(!!1) // true

console.log(!!'f8') // true

console.log(!!['Mercedes']) // true

Copy

Thêm !! phía trước các giá trị truthy sẽ luôn trả về true.

**Falsy - to bool is false**

Bất cứ giá trị nào trong Javascript khi chuyển đổi sang kiểu dữ liệu boolean mà có giá trị false thì ta gọi giá trị đó là Falsy.

Trong Javascript có 6 giá trị sau được coi là Falsy:

1. false
2. 0 (số không)
3. '' or "" (chuỗi rỗng)
4. null
5. undefined
6. NaN

Ví dụ:

console.log(!!false) // false

console.log(!!0) // false

console.log(!!'') // false

console.log(!!null) // false

console.log(!!undefined) // false

console.log(!!NaN) // false

Copy

**Chú ý!**

Nội dung đã đề cập phía trên đã đầy đủ khi nói về Truthy và Falsy trong Javascript. Tuy nhiên mình vẫn cần nhấn mạnh lại với các bạn rằng:

Ngoài 6 giá trị đã đề cập tới ở phần Falsy thì toàn bộ các giá trị khác đều là Truthy, kể cả những giá trị sau:

1. '0' (một chuỗi chứa số không)
2. ' ' (một chuỗi chứa dấu cách)
3. 'false' (một chuỗi chứa từ khóa false)
4. [] (một array trống)
5. {} (một object trống)
6. function(){} (một hàm “trống”)

Một số người chuyển từ ngôn ngữ khác sang rất có thể sẽ bị nhầm [] (mảng “rỗng”) là falsy, bởi vì trong ngôn ngữ họ đã học trước đó [] là falsy.

Với những người hiểu nhầm [] là falsy sẽ gặp trường hợp khó hiểu sau:

Ví dụ:

var cars = [] // Dù là mảng "rỗng" vẫn là truthy

if (!cars) {

// Họ sẽ thắc mắc: "Tại sao lại không lọt vào đây?"

}

Copy

Vì [] là truthy nên !cars sẽ trả về false. Câu lệnh if sẽ nhận được kết quả của mệnh đề so sánh là false, vì vậy đoạn mã trong if trên sẽ không được lọt vào.

**Ngoại lệ? - document.all**

Trong Javascript (phía trình duyệt) sẽ có sẵn một đối tượng document, và khi bạn thử !!document.all sẽ trả về false. Chẳng lẽ document.all cũng là falsy hay sao?

Bản thân mình cũng thắc mắc điều này nên mình đã search Google “Why document.all is falsy?” và mình đã tìm được câu trả lời [tại đây](https://stackoverflow.com/questions/10350142/why-is-document-all-falsy).

**Tóm tắt câu trả lời:**

document.all là một ngoại lệ chính thức duy nhất theo đặc tả ECMA (phiên bản 5). Đặc tả này mô tả toàn bộ các object khi chuyển sang boolean sẽ là true. Tuy nhiên, document.all là một ngoại lệ.

**Cụ thể như sau:**

1. document.all chuyển sang boolean sẽ là false
2. document.all khi là toán hạng của toán tử so sánh == hoặc != sẽ là undefined
3. Khi typeof document.all sẽ trả về "undefined"

ECMA là đặc tả chi tiết kỹ thuật mà các ngôn ngữ theo đặc tả này phải tuân theo. Javascript là một ngôn ngữ tuân thủ đặc tả kỹ thuật ECMA.