Link Ref: <https://javascript.info/>

**Introduction:**

1. Js Engine

Link Ref: <https://codecute.com/javascript/javascript-engine-la-gi-tim-hieu-javascript-engine.html>

**Fundamentals:**

1. Use strict

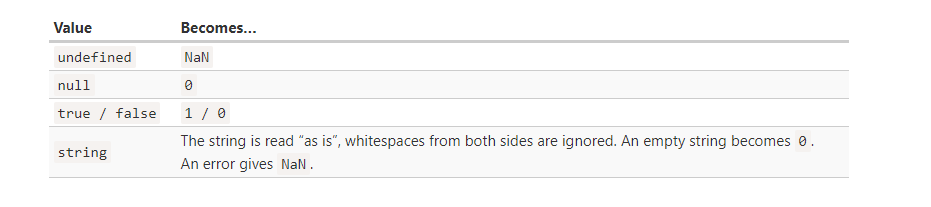
Link Ref: <https://viblo.asia/p/use-strict-la-gi-va-cach-su-dung-trong-javascript-3P0lPz2mKox>

Js lộn xộn nhưng khi dùng use strict => chính xác và nghiêm ngặt 🡺 ko thể viết code bừa bãi

1. Variable: let, var, const

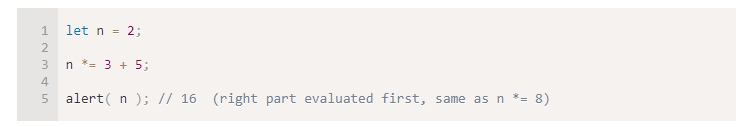
Link Ref: <https://viblo.asia/p/phan-biet-kieu-bien-var-let-va-const-trong-javascript-ORNZqaOnZ0n>

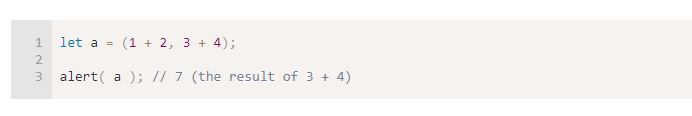
1. Type conversion

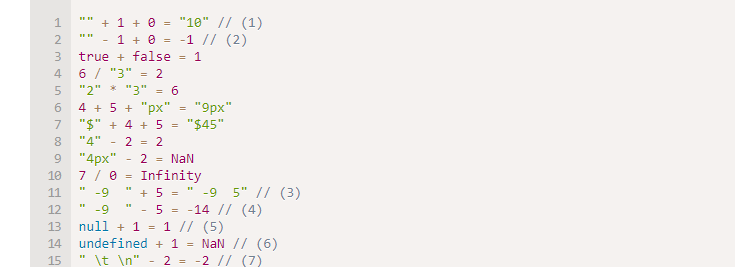


1. Operators

++counter (tăng trước sau đó sử dụng) and counter++(sử dụng trước sau đó tăng)

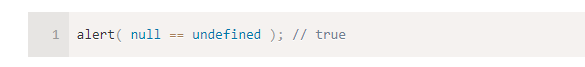






1. Comparisons

+ Comparison diffirent type => convert to number

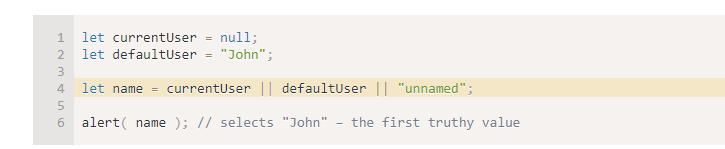


+ Null => 0, underfined => NaN

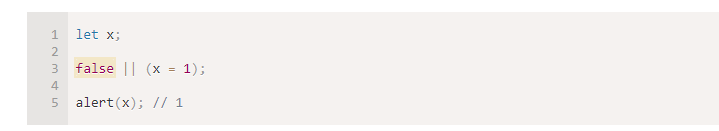
+ Don’t use >=, <=, == with variable may be null/underfined

1. Logical operators

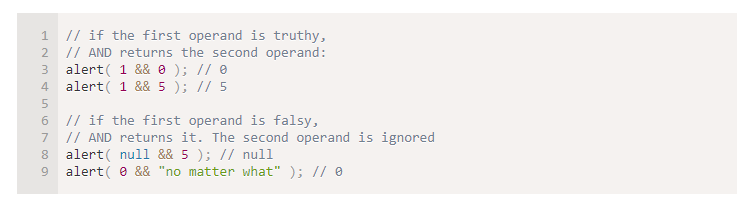
\*\* Trả về giá trị khác null/underfined đầu tiên



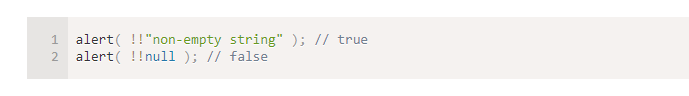
\*\* Short-circuit evaluation



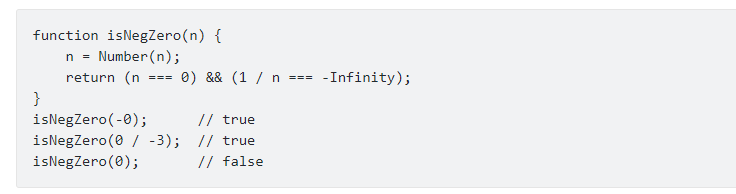
\*\*



\*\* !! : convert to Boolean type



\*\* Phân biệt 0 and -0 sử dụng Infinity



**Code quality:**

1. Comment

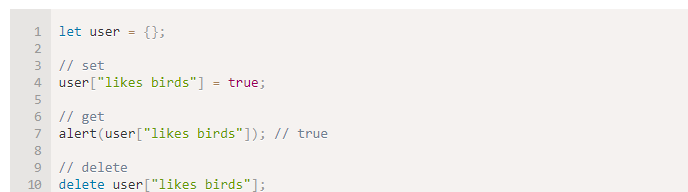
JS DOC: <https://en.wikipedia.org/wiki/JSDoc>

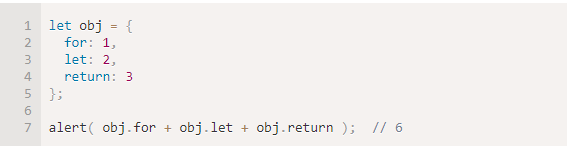
1. Testing with mocha
2. Polyfills

Là một dự phòng của trình duyệt, cho phép chức năng chạy được trên các trifh duyệt cũ hơn.

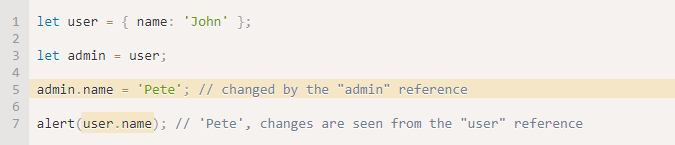
**Object the basic**

1. Objects





++ Khi 1 biến được copy, referen được copy, object ko duplicate



++ Clone object (muốn tạo 1 object mới, mà ko phải là reference)

++ Assign: sao chép thuộc tính riêng có thể đếm được từ 1 hoặc nhiều đối tượng => đối tượng đích

1. Symbols

++ Là duy nhất



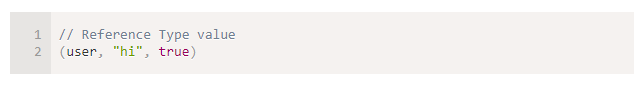
1. Object Method, this

++ keyword “this” ko phụ thuộc vào vị trí gọi mà phụ thuộc vào đối tượng trước khi chạy.

Ref: <https://viblo.asia/p/this-trong-javascript-gAm5ywe8Zdb>

Tham chiếu tới đối tượng gọi hàm this

++ Advance for this.



++ function arrow ko có this, nó được lấy từ bên ngoài

1. Object to primitive conversion

3 case: “string”, “number”, “default”.

3 function:

Symbol.toPrimitive

obj.toString()

obj.valueOf()

* The toString method returns a string "[object Object]".
* The valueOf method returns the object itself.
* Nếu object ko chứa các hàm này, sẽ biến đổi theo mặc định