[TỔng hợp 4 phần](https://viblo.asia/p/4-javascript-design-patterns-you-should-know-Qbq5Qr74KD8)

TÓm tắt

Singleton Pattern: tạo nhiều phiên bản của 1 đối tượng.

Module Pattern : tạo phân quyền truy cập như class.

Constructor Pattern: khai báo và truy cập prototype

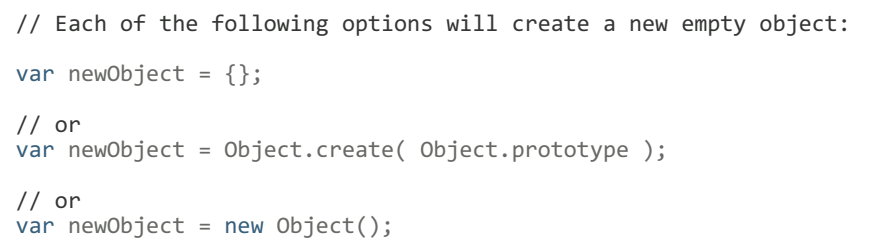
Observer Pattern: Khi 1 đối tượng thay đổi thì các đối tượng(const)

đăng ký theo dõi sẽ được nhận thông báo

Factory Pattern: Giống như interface trong java.

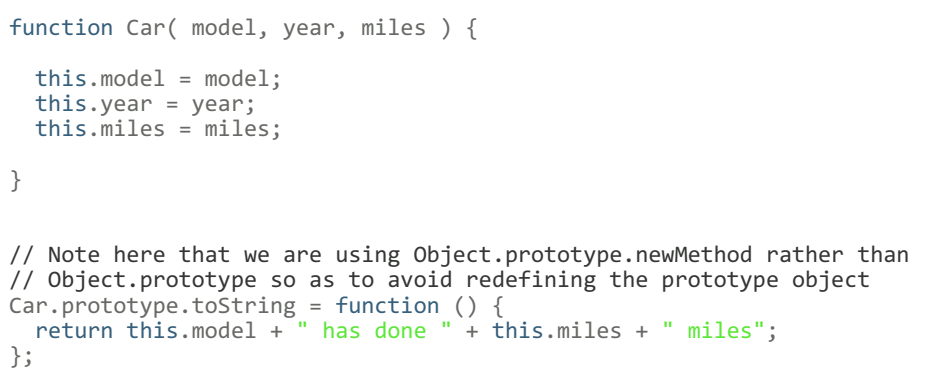
Constructor Pattern

+ Object create: 3 cách



+ Basic Constructors: Ko hỗ trợ class, chỉ hỗ trợ function dạng đặc biệt.

+ Nên dùng .prototype để tránh định nghĩa lại mỗi khi new object



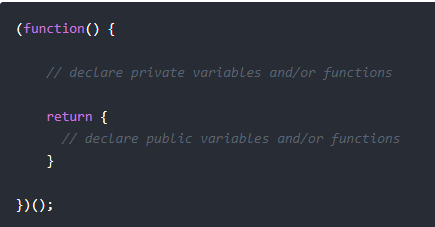
Module Pattern => class trong Java :

Tính đóng gói.

Bảo vệ truy cập các biến trong class.(private, public).

<https://labs.septeni-technology.jp/technote/javascript-module-pattern-mot-cai-nhin-khai-quat/>

Tổng quát

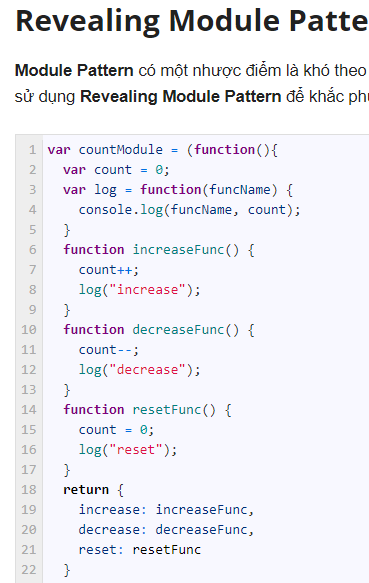


Ví dụ:



Một ví dụ khác cho việc quản lý code khi dung module pattern:

+ Dễ nhìn hơn



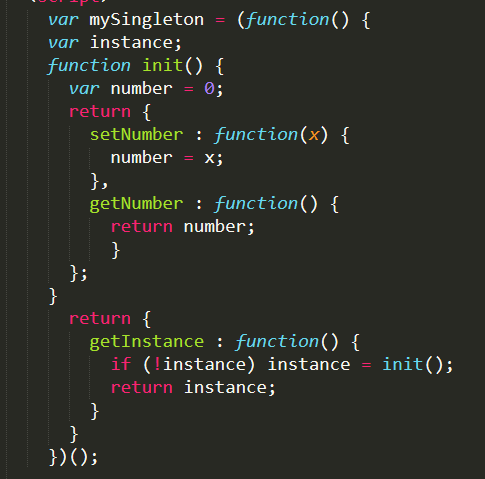
2 cách khai báo prototype trong JS:



Singleton Pattern

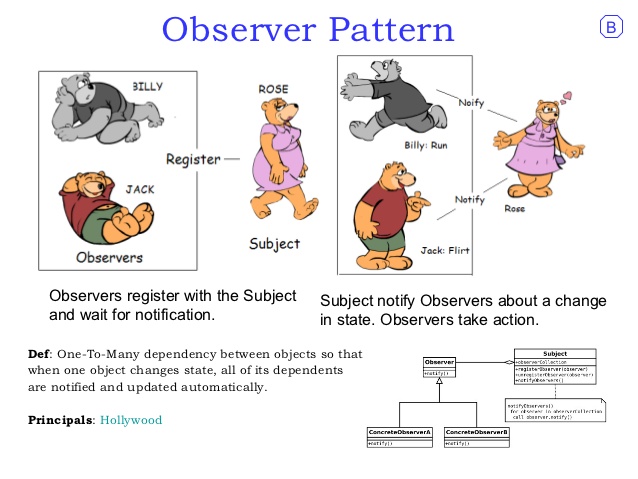
<https://viblo.asia/p/javascript-design-pattern-singleton-pattern-gDVK28GelLj>

Một Singleton chỉ cho phép cho một khởi tạo đơn, nhưng nhiều phiên bản của cùng một đối tượng. Singleton hạn chế clients tạo ra nhiều đối tượng, sau khi đối tượng đầu tiên được tạo ra, nó sẽ trả về các phiên bản của chính nó.



Observer Pattern

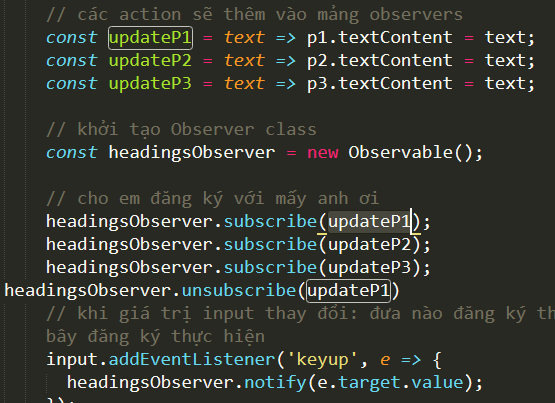
Khi 1 đối tượng thay đổi thì các đối tượng đăng ký theo dõi sẽ được nhận thông báo



<https://luubinhan.github.io/blog/2018-04-17-giai-thich-observer-pattern-trong-javascript/>

<https://xuanthulab.net/observer-pattern-mo-hinh-lap-trinh-chu-the-quan-sat.html>





Factory Pattern

<https://viblo.asia/p/factory-method-creational-patterns-maGK7WLOKj2>

Giống như interface trong java.

Không phải dung biến this và new như contructor khi l=tạo class.

VD1:





VD2:

<https://medium.com/front-end-hacking/understand-the-factory-design-pattern-in-plain-javascript-20b348c832bd>

