**TÌM HIỂU VỀ ES6**

**Nhóm 3:**

**Gmail:** ltdd.nhom3@gmail.com

**Pass:** ltdd\_nhom3\_2018

**Github:**

* + Tên: Nhom3LTDD
  + Link Project: https://github.com/Nhom3LTDD/LTDD3\_Nhom3.git

1. So sánh sự khác nhau giữa Let, Var trong ES6: (Vũ Thái Ngân – Nguyễn Quang Hiểu)

* Giống nhau:
  + Var, let và const đều là từ khoá – keyword.
  + Var, let và const cùng được dùng để khai báo biến trong JavaScript.
* Khác nhau:
  + Let tạo ra một biến chỉ có thể truy cập được trong block bao quanh nó, khác với Var - tạo ra một biến có phạm vi truy cập xuyên suốt function chứa nó. Được giới hạn bởi dấu {}
  + Trong khi đó, var có phạm vi sử dụng lớn hơn, phạm vi của biến số sử dụng **var** là phạm vi hàm số hoặc bên ngoài [hàm số](https://completejavascript.com/tim-hieu-function-javascript/), toàn cục.
  + Const dùng để khai báo hằng số, giá trị không thể thay đổi trong suốt quá trình chạy chương trình
* Ví dụ minh họa:

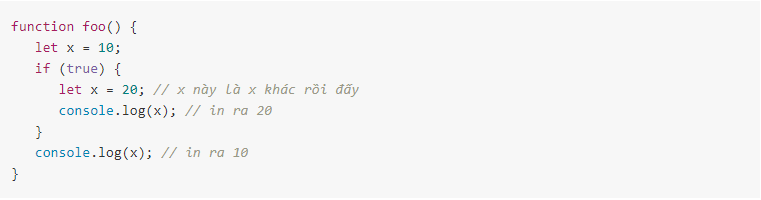
+ *Ví dụ cho var:*

Từ ví dụ minh họa sử dụng cho biến var cho thấy, chúng ta có thể truy cập biến x trong if và cả ngoài if với giá trị in ra không thay đổi đều là 20. Giá trị trong x đã được gán luôn cho bên ngoài.



+ *Ví dụ cho let:*

Ta thấy, khác với var, trong biến if x được gán giá trị = 20, nhưng sau khi trong khỏi điều kiện, biến x được trả lại giá trị như ban đầu, x = 10.

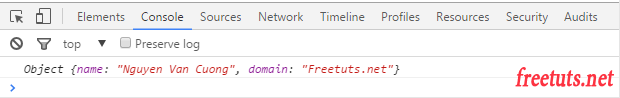


*+ Ví dụ cho const*

Đầu tiên, ta khởi tạo cho const



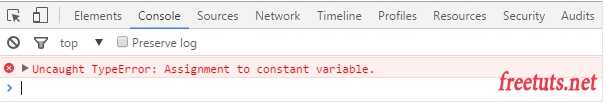
Chúng ta được kết quả:



Như chúng ta thấy, kết quả được hiển thị với tên và domain như đã được khai báo, chúng ta tiếp tục test với Const



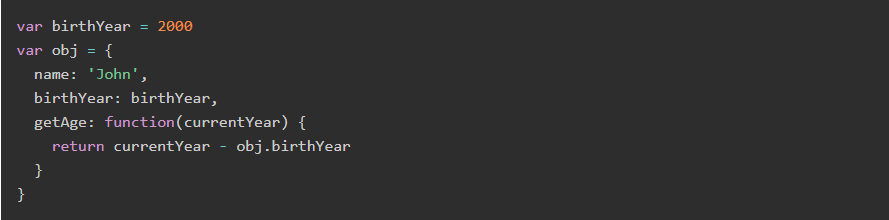
Thay đổi giá trị của const thì màn hình sẽ báo lỗi:



Như vậy có thể thấy, giá trị của const không thể thay đổi. Nếu khai báo biến const trong phạm vi của cặp {} thì nó chỉ tồn tại trong phạm vi đó mà thôi. *( Nguồn của “const” được tham khảo từ freetuts.net)*

1. Kiểu dữ liệu Object: (Phạm Trọng Duyễn)

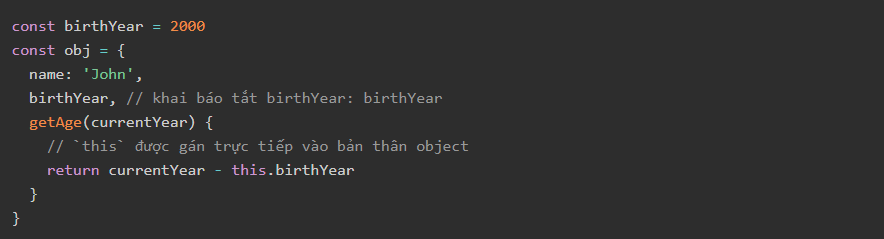
*Khai báo Object đơn giản:*



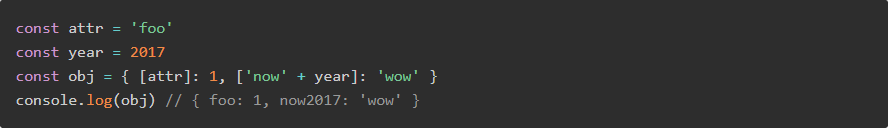
Trong ES6, Object được thêm tên gọi là Object literals (tạm dịch là Object chân phương)

*Object literals* (tạm dịch là Object chân phương) cho phép bạn khai báo tắt thuộc tính của object với biến cùng tên, và khai báo phương thức cho object.

Ví dụ:

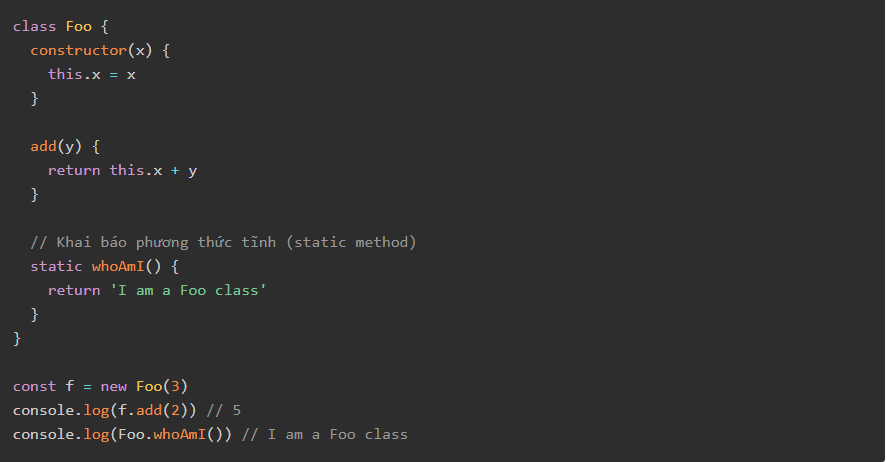


Ngoài ra từ ES6 bạn cũng có thể khai báo thuộc tính cho object một cách linh động bằng cách sử dụng cú pháp []



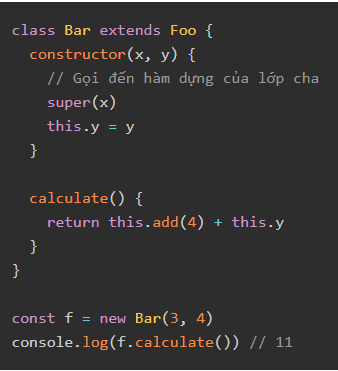
1. Khai báo class trong ES6 ( Hoàng Văn Giang)

*Cú pháp khai báo một CLASS trong ES6* được cải tiến hơn so với ES5, chúng ta không cần sử dụng function hay prototype, cú pháp ES6 được sử dụng như:



Chúng ta cũng có thể kế thừa từ lớp khác bằng từ khóa extends.

Ví dụ minh họa kế thừa:



Chúng ta có thể dùng hàm mũi tên khi khai báo phương thức trong lớp. Điều này giúp đảm bảo this luôn trỏ đến đối tượng hiện tại. Cú pháp này đặc biệt thông dụng trong các ứng dụng React, tuy nhiên **vẫn chưa được hỗ trợ mặc định** bởi các trình duyệt, nên bạn phải dùng Babel để chuyển đổi mã nguồn.

Ví dụ:

