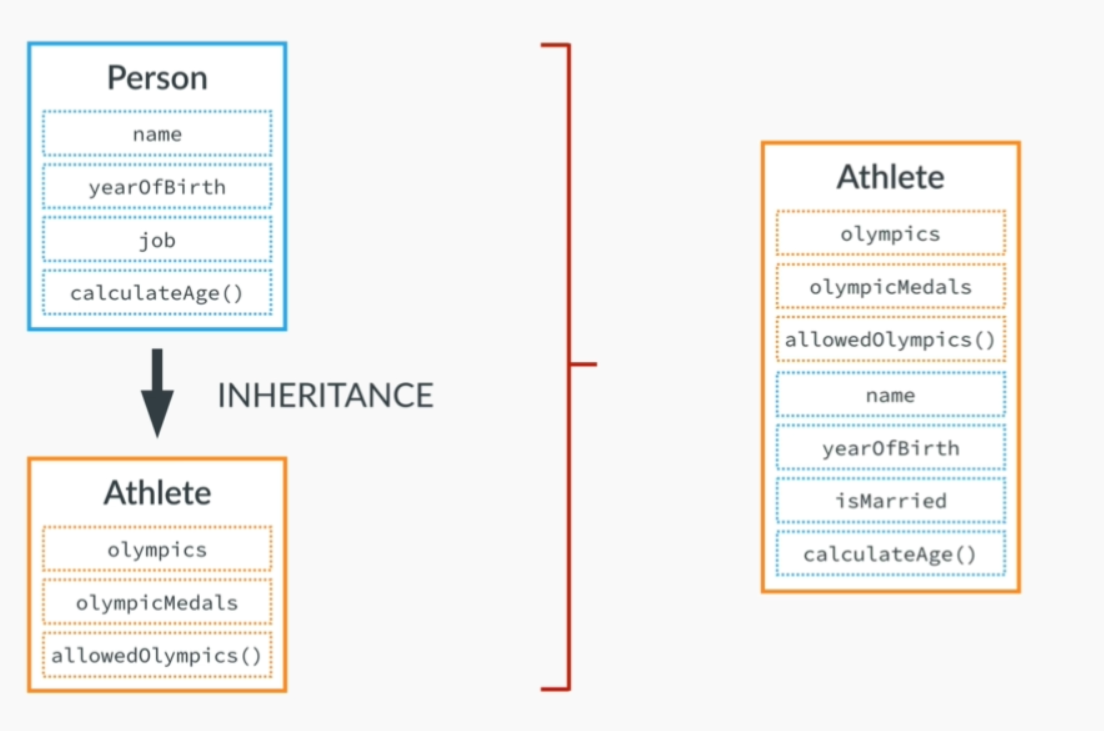
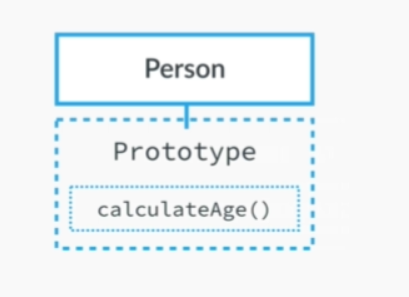
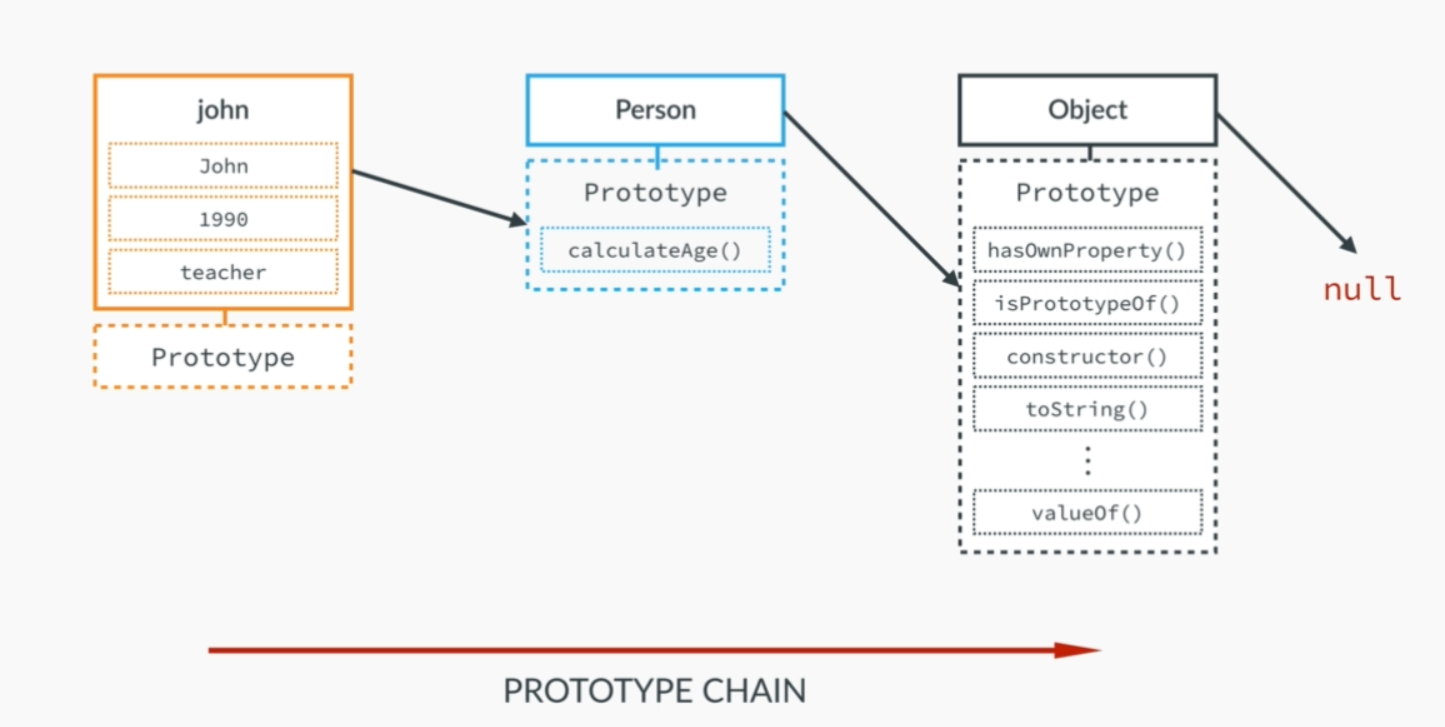


Contructor ask like a blue print

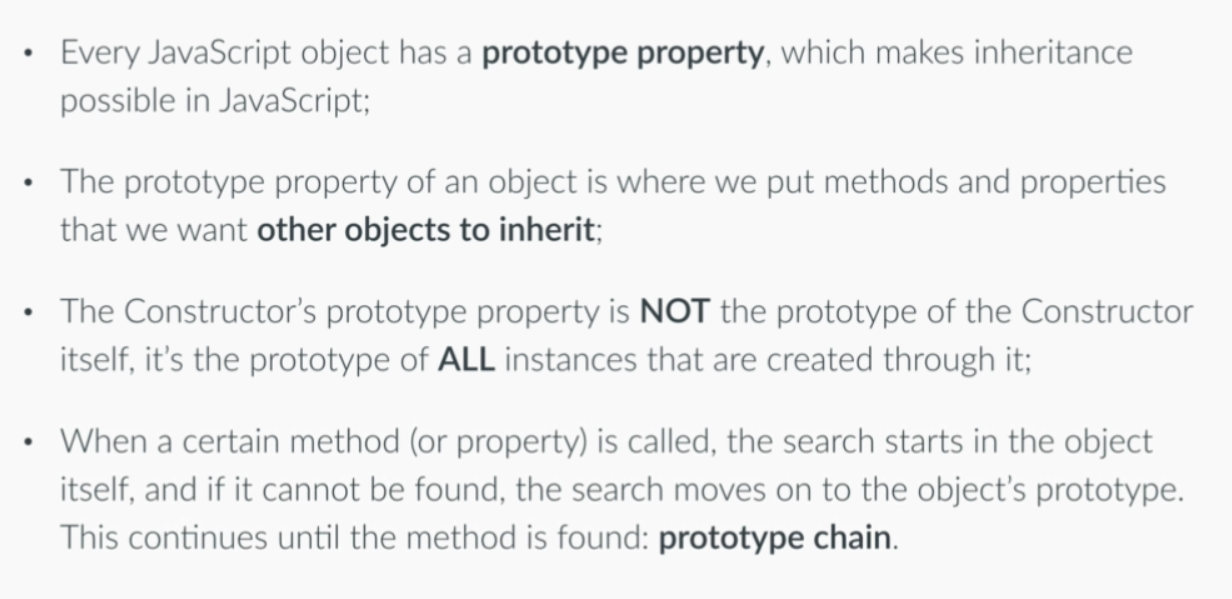




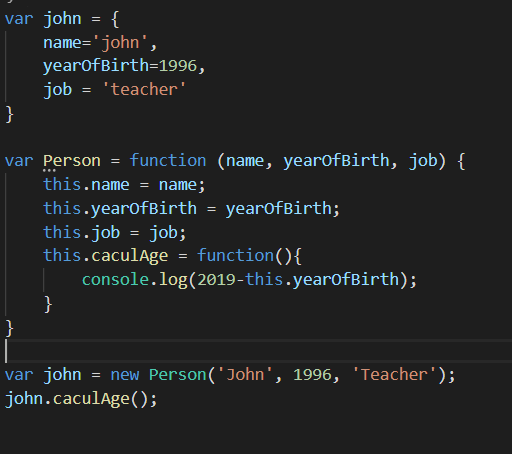
Prototype is what we can do with property or method



The prototype property of an object is where we put method and property that we want other object to inherit

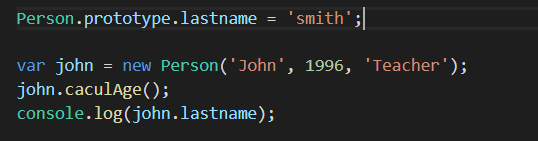


1. Function contructor

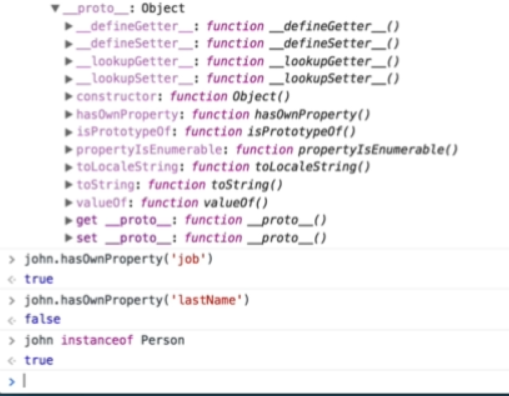


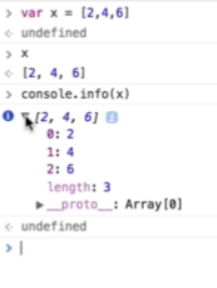
New : create an empty object then this point to this object( normally this point to global object)



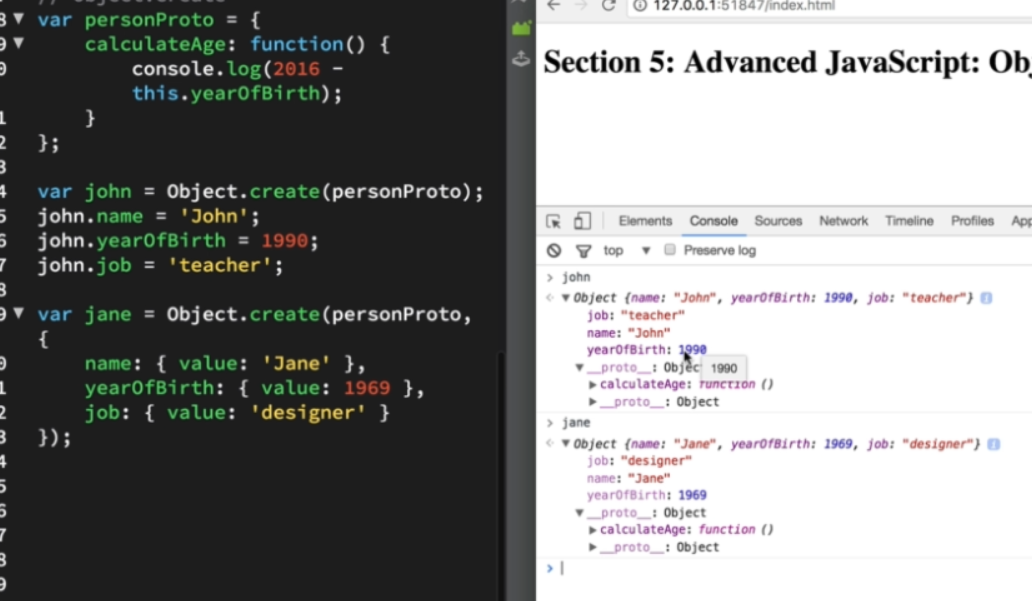


Instead of using method in object, we should use prototype so I can use it to all object without have it in the object





1. Creating objects with object.create



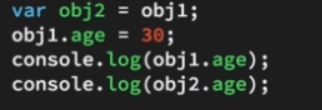
1. Primitive vs object

Variable contaning primitive actually hold the data inside the variable it self.

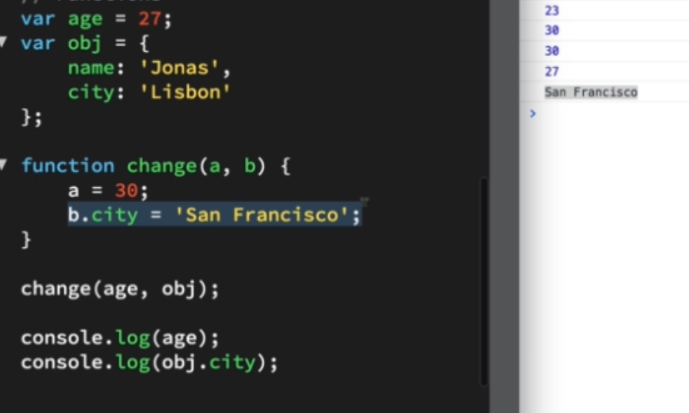
With object variable do not hold the date, it’s point to the address have the data



: obj1 and obj2 point to the same address so it’s the same object. Different to variable



So when we change obj1.age also mean we change obj2.age -> work same with other not primitive

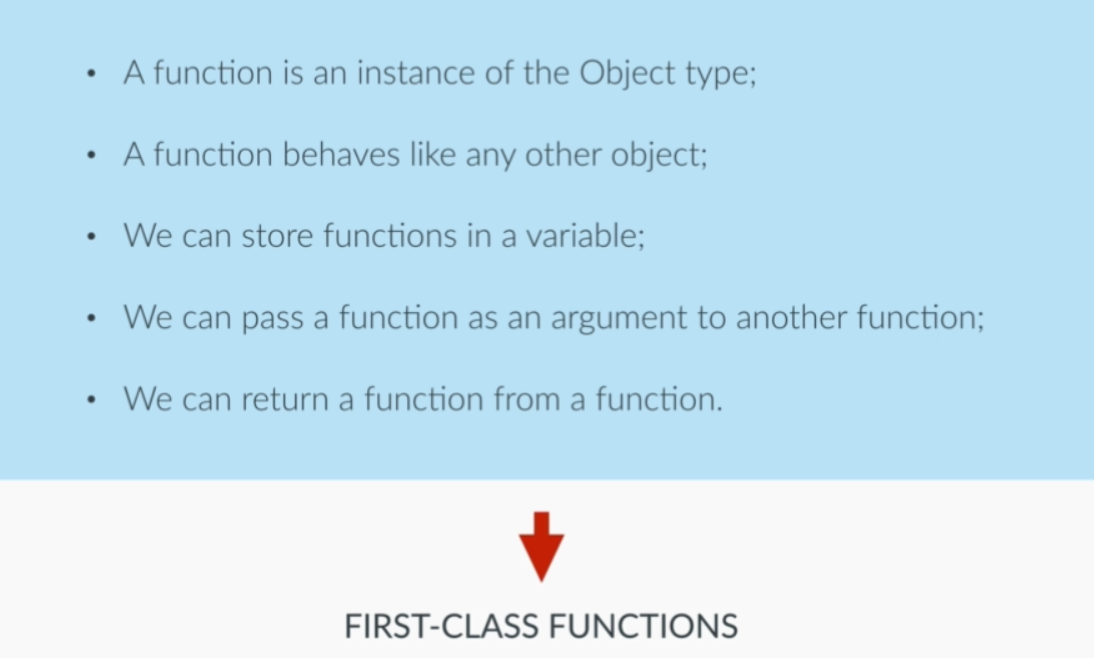


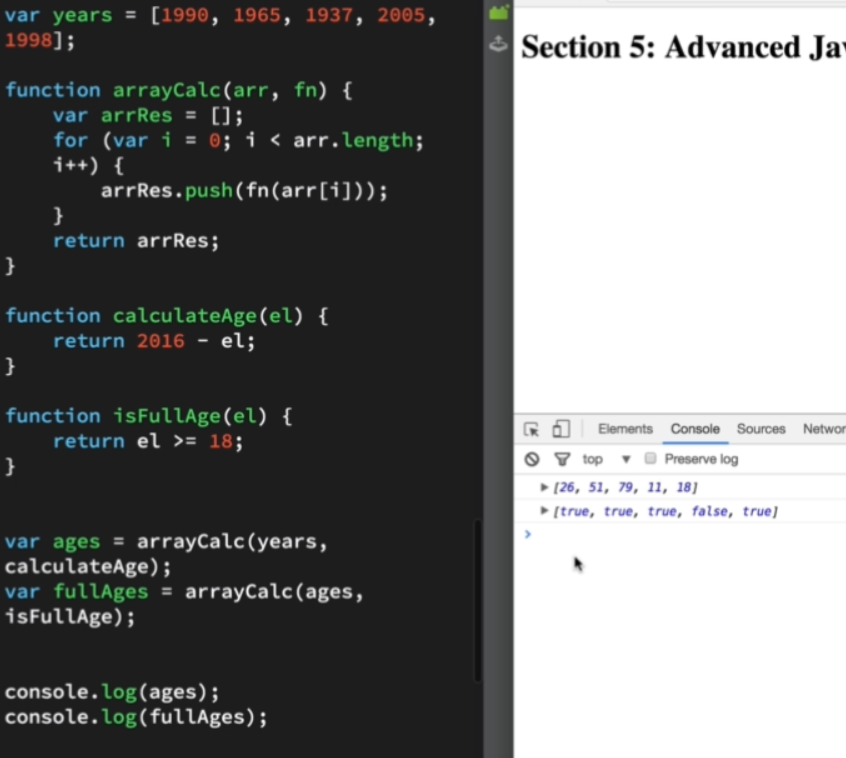
Không in ra a 30 bởi vì thoát hàm a bị giải phóng -> a =27

Tuy nhiên obj.city giữ nguyên

* Biến primitive sẽ thay đổi theo hàm, phép so sánh bở vì nó lưu chính giá trị đó -> kết thúc hàm, giá trị được trả về cho biến global hoặc được giải phóng như age ở trên
* Biến object thay đổi theo object, obj trỏ tới địa chỉ, khi thay đổi giá trị tại địa chỉ đó thì thay đổi luôn -> như obj.city ở trên

1. Function





Một hàm là input của một hàm khác

Tham số hàm isFullAge or hàm caculate được lấy trong thân hàm arraycalc, do đó không cần truyền tham số vào

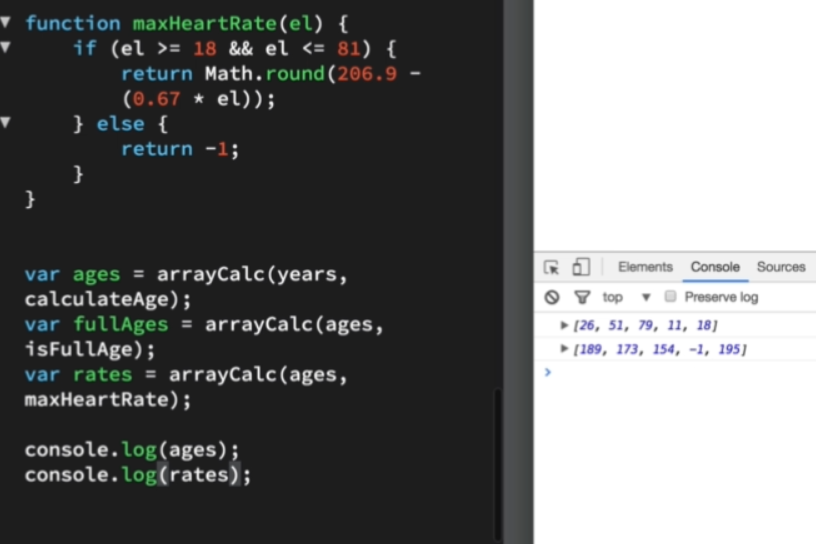
* This is function as a argument ( đối số)

Khi sử dụng hàm theo cách này hàm arrayCalc có hai đối số là

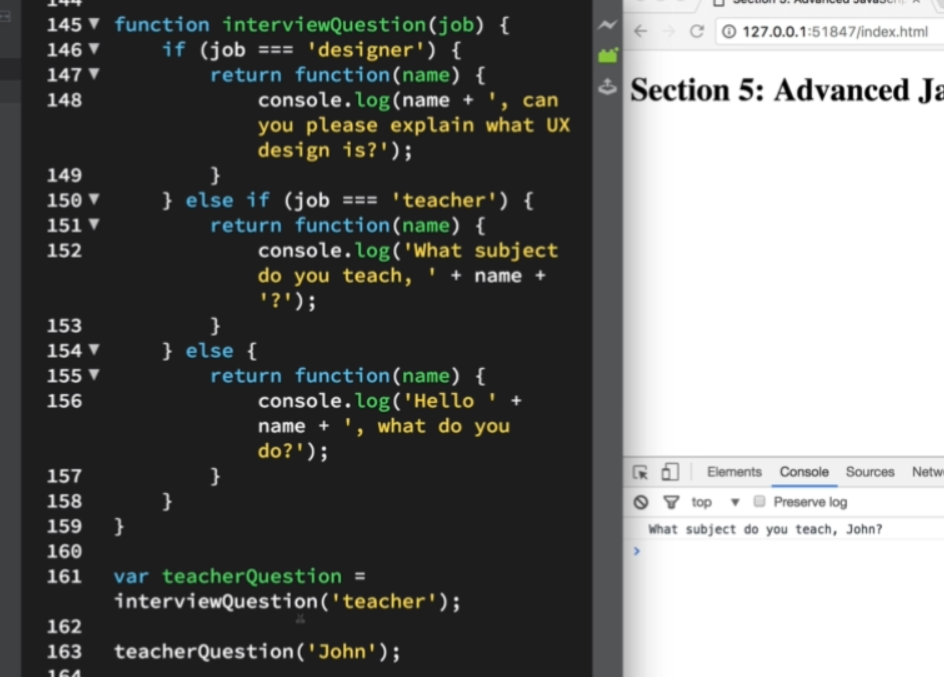
Mảng Đối tượng

Hàm đối số

Thành phần đối tượng ở đây cũng chính là đối số truyền vào cho hàm đối số



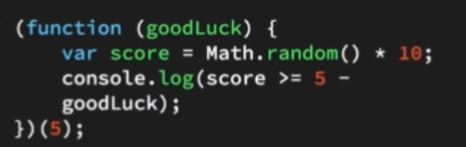
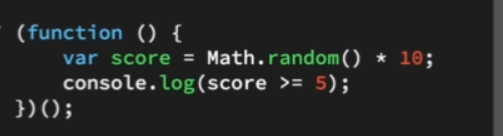
1. Funtion returning funtion



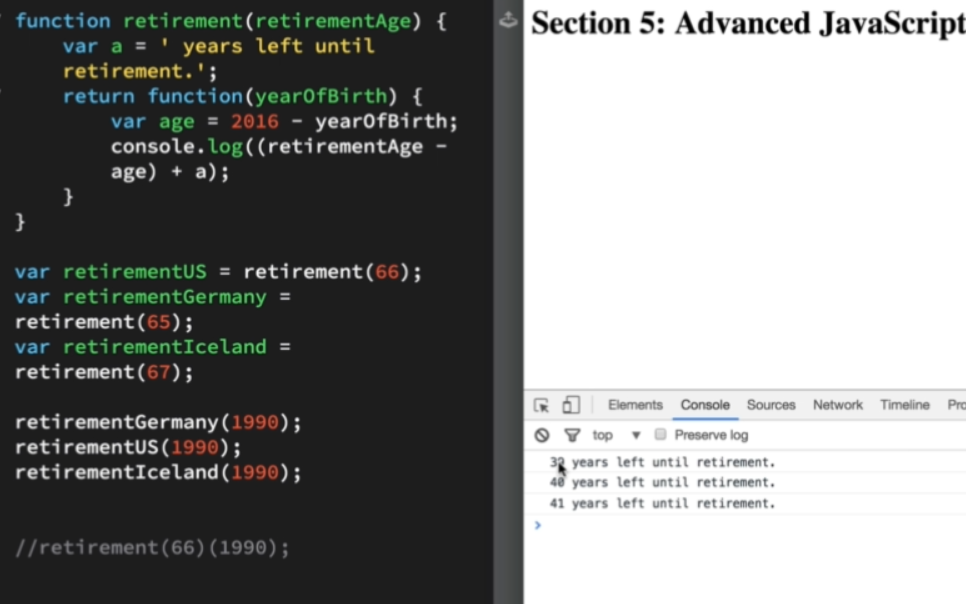
1. Iife

IIFE là viết tắt của**Immediately Invoked Function Expression**, có nghĩa là khởi tạo một function và thực thi nó ngay lập tức.

 IIFE như là một các hộp đóng gói code của chính nó. Do đó, những biến trong hộp này là private, bên ngoài(global) không thể truy xuất hay thay đổi được. Và nếu vô tình bạn đặt tên biến giống với global thì cũng không bị ảnh hưởng bên ngoài

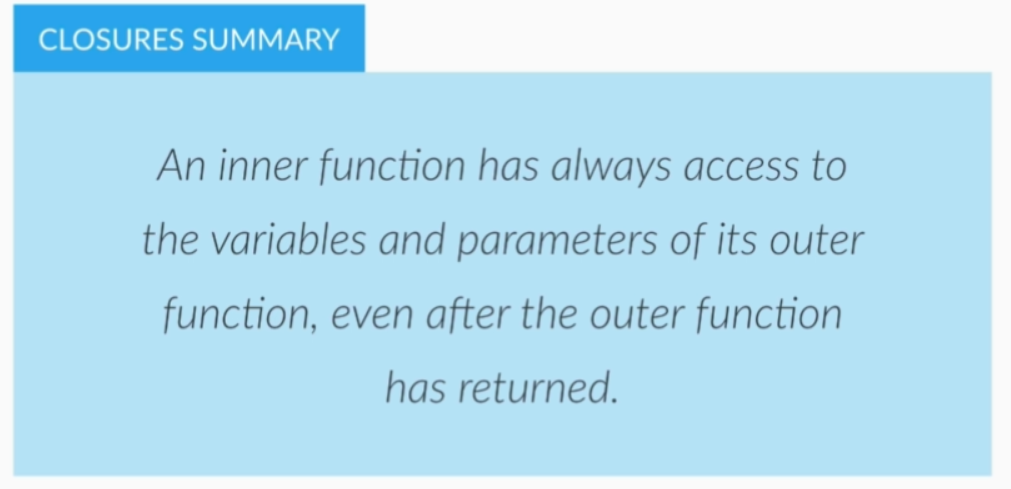


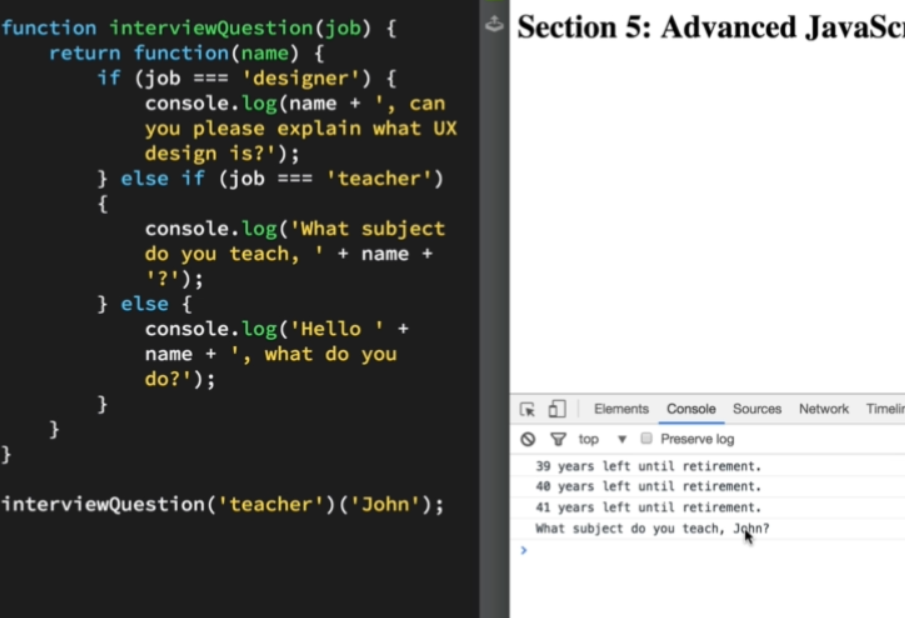
1. Closure



Gọi hàm 1 xong -> trả về hàm hai -> hàm hay chạy -> truyền biến vào

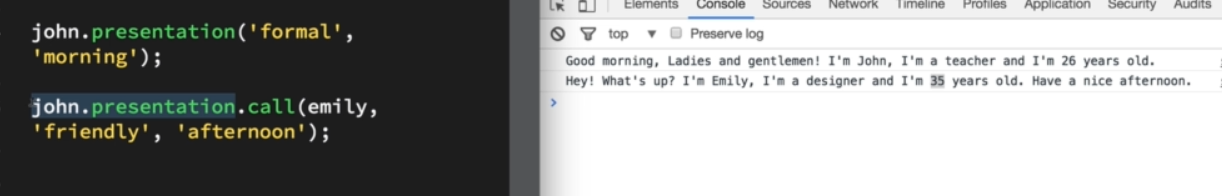
Closure là hàm trong vẫn giữ lại và xài biến của hàm ngoài, ngay cả khi hàm ngoài đã return( đã kết thúc)

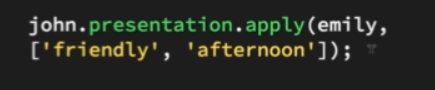




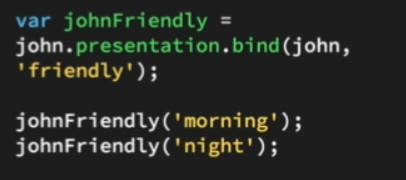
Có thể sử dụng job trong function

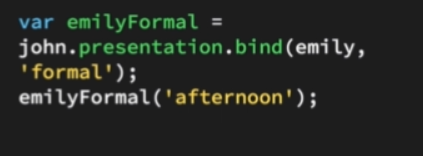
1. Call method



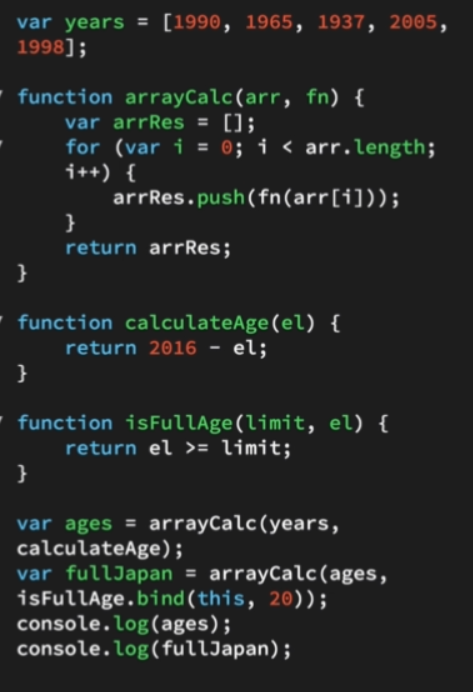


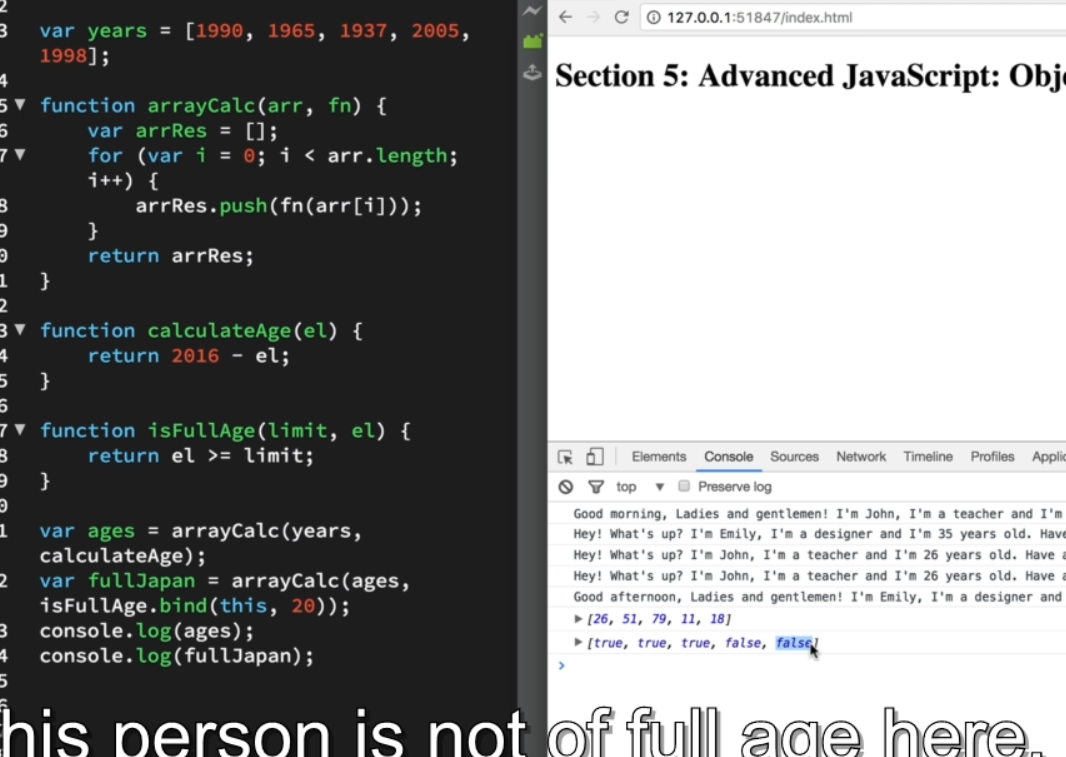
Apply giống call nhưng đối số là mảng





Bind tạo ra một hàm mới từ 1 hàm khác, với một hoặc một vài biến đã được cài sãn





Hàm isfullage gồm 2 thuộc tính, nhưng để truyền vào thì chỉ có 1 thuộc tính -> dùng bind truyền thuộc tính limit trước khi truyền vào

parseInt() : chuyển chuỗi về số