**Basic exercises**

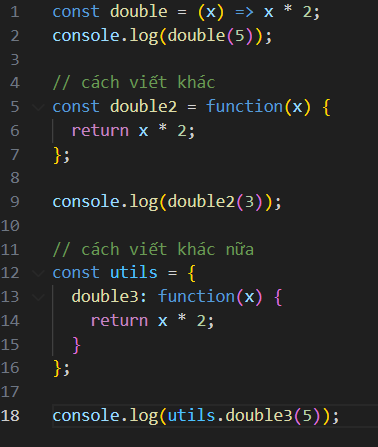
1. **Arrow function cơ bản – double & isEven**

**Mục tiêu:** Nắm cú pháp arrow function, return tường minh/ngầm định.  
**Yêu cầu:**

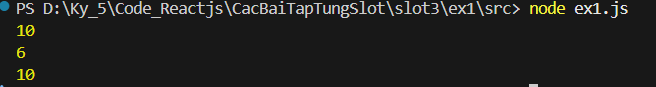
* Viết double(n) trả về n \* 2.
* Viết isEven(n) trả về true/false.
* In kết quả: double(7), isEven(10), isEven(7).  
  **Ràng buộc:** Dùng **arrow function** cho cả 2 hàm.

Ví dụ:

Code:



**Kết quả:**



## Rest parameter – tính tổng và trung bình

**Mục tiêu:** Nhận số lượng tham số bất kỳ bằng ...args.  
**Yêu cầu:**

* Viết sum(...nums) trả về tổng các số hợp lệ (bỏ NaN, string không số).
* Viết avg(...nums) trả về trung bình (2 chữ số thập phân), nếu rỗng trả 0.
* In: sum(1,2,3), sum(1,'x',4), avg(1,2,3,4), avg().  
  **Ràng buộc:** Dùng **rest parameter** + reduce.

const companies = [

{ name: "Company One", category: "Finance", start: 1981, end: 2004 },

{ name: "Company Two", category: "Retail", start: 1992, end: 2008 },

{ name: "Company Three", category: "Auto", start: 1999, end: 2007 },

{ name: "Company Four", category: "Retail", start: 1989, end: 2010 },

{ name: "Company Five", category: "Technology", start: 2009, end: 2014 },

{ name: "Company Six", category: "Finance", start: 1987, end: 2010 },

{ name: "Company Seven", category: "Auto", start: 1986, end: 1996 },

{ name: "Company Eight", category: "Technology", start: 2011, end: 2016 },

{ name: "Company Nine", category: "Retail", start: 1981, end: 1989 }

];

const ages = [33, 12, 20, 16, 5, 54, 21, 44, 61, 13, 15, 45, 25, 64, 32];

const person = {

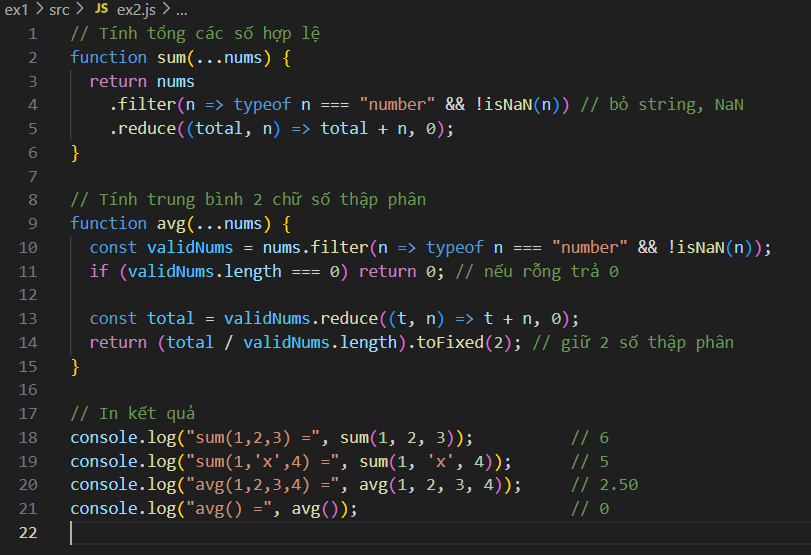
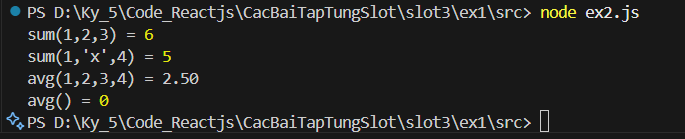
name: "Costas",

address: {

street: "Lalaland 12"

}

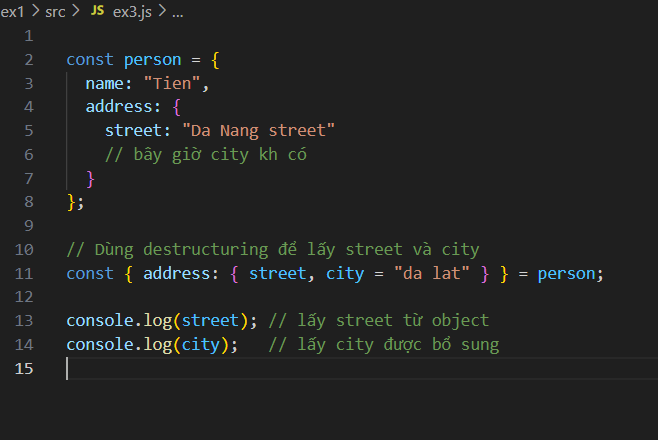
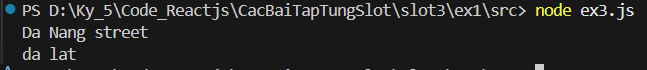
};

Ví dụ :   
Code   
  
  
  
  
Kết quả   


## Destructuring object lồng nhau – lấy địa chỉ

**Mục tiêu:** Lấy thuộc tính lồng nhau + giá trị mặc định.  
**Yêu cầu:**

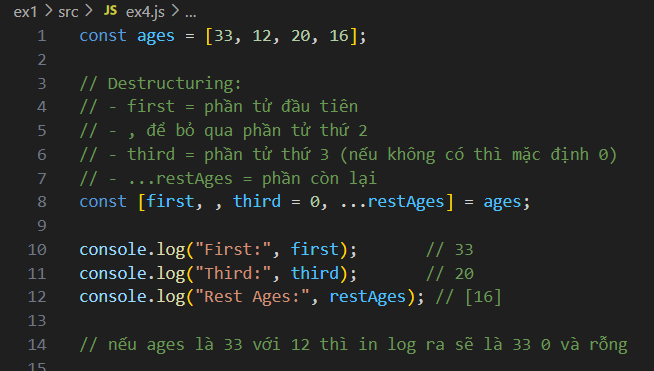
* Cho person như dưới. Dùng **destructuring** để lấy street, city (mặc định "Unknown City" nếu không có).
* In: street, city.  
  **Ràng buộc:** Không truy cập kiểu person.address.street trực tiếp.  
  **Kỳ vọng console:**

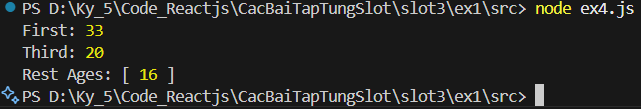
**code** **kết quả :**

## Destructuring array + skip + default

**Mục tiêu:** Bỏ qua phần tử, đặt mặc định.  
**Yêu cầu:**

* Với const ages = [33, 12, 20, 16];
* Dùng **destructuring** để lấy first, bỏ qua phần tử thứ 2, lấy third (mặc định 0 nếu không tồn tại), và restAges cho phần còn lại.
* In: first, third, restAges.

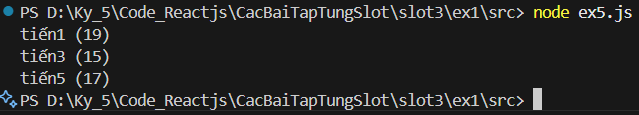
Code :   


Kết quả :   


## Map + filter – danh sách teen

**Mục tiêu:** Kết hợp filter và map với arrow function.  
**Yêu cầu:**

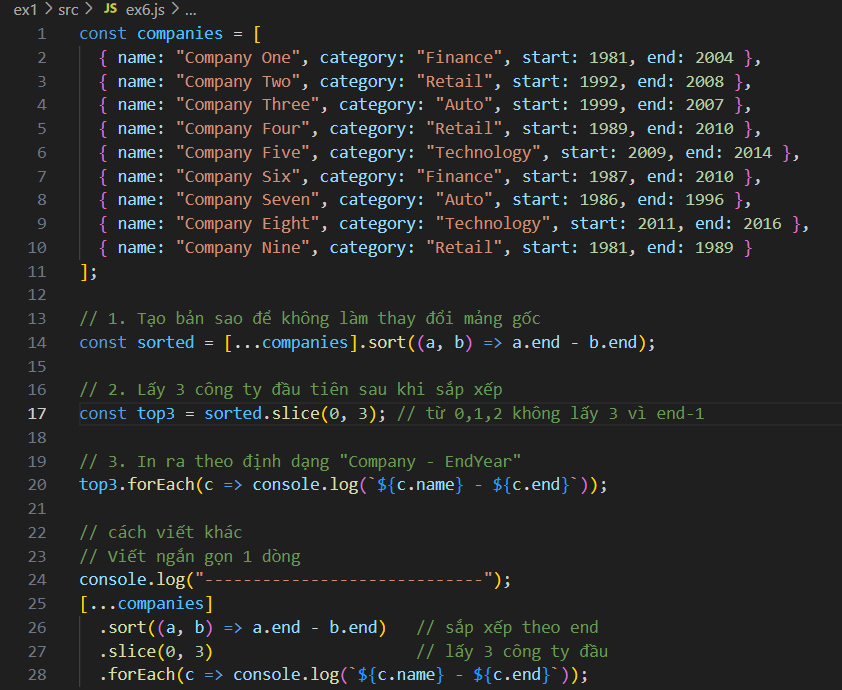
* Cho mảng people (name, age).
* Lọc những người tuổi 13–19 (bao gồm 13 và 19).
* Map sang chuỗi "Ann (19)".
* In ra từng dòng.

Code :   
kết quả : 

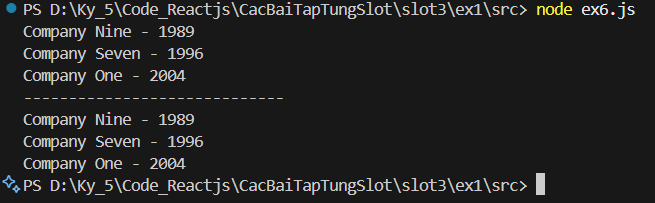
## Sort + slice – doanh nghiệp theo năm kết thúc

**Mục tiêu:** Thao tác sort bất biến (không mutate), cắt mảng.  
**Yêu cầu:**

* Cho companies (name, category, start, end).
* Tạo bản sao đã sắp xếp theo end tăng dần.
* In **3 công ty đầu** theo định dạng "Company - EndYear".

Code :   


Kết quả :

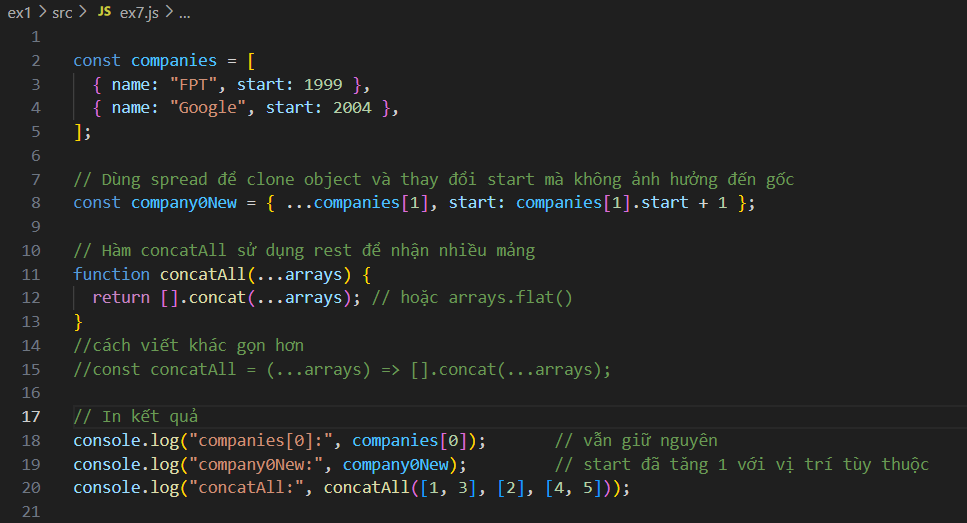


## Spread vs. rest – bất biến & gộp mảng

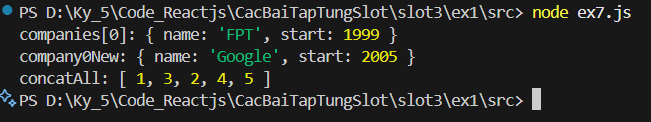
**Mục tiêu:** Phân biệt rest (định nghĩa hàm) và spread (sao chép/gộp).  
**Yêu cầu:**

* Từ companies[0], tạo company0New với start += 1 **mà không làm đổi companies[0]**.
* Viết hàm concatAll(...arrays) trả về mảng gộp của mọi mảng truyền vào.
* In: companies[0] và company0New; kết quả concatAll([1,2],[3],[4,5]).

Code



Kết quả :

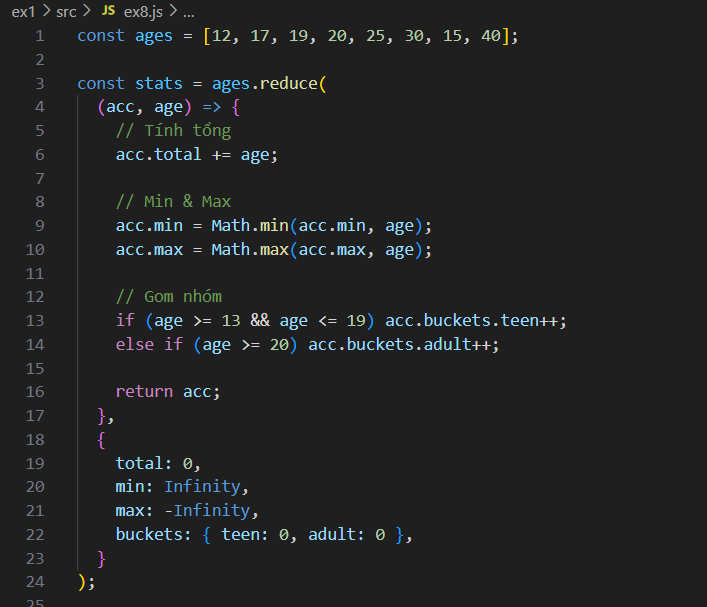
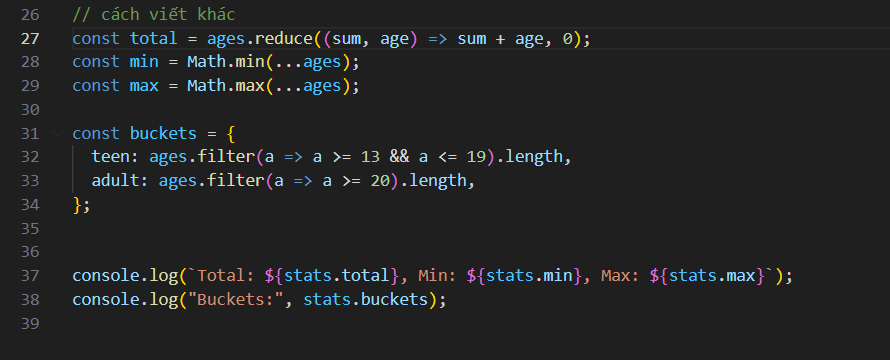


## Reduce nâng cao – thống kê tuổi

**Mục tiêu:** Tính tổng, min, max, và đếm theo nhóm bằng reduce.  
**Yêu cầu:**

* Với ages (mảng số), tính:
  + total, min, max
  + buckets: { teen: count(13–19), adult: count(>=20) }
* In dạng:
  + Total: X, Min: Y, Max: Z
  + Buckets: { teen: a, adult: b }

Code :

   
kết quả :

