



# Web Front End cho người mới

## Phần 8 – Javascript cơ bản: Vòng lặp

## Lặp qua các ký tự của 1 chuỗi

```
const string = 'Techmaster';

for (let i = 0; i < string.length; i++) {
  console.log(string[i])
}
```

## Lặp qua các thuộc tính của Object bằng for ... in

```
const student = {  
  firstName: 'Tèo',  
  lastName: 'Nguyễn',  
  age: 20  
}  
  
for (const key in student) {  
  console.log(student[key])  
}
```

## Lặp qua các thuộc tính của Array và String bằng for ... of

```
const array = ['a', 'b', 'c', 'd'];  
  
for (const item of array) {  
    console.log(item)  
}
```

```
const string = 'Techmaster';  
  
for (const character of string) {  
    console.log(character)  
}
```



## Phương thức map() của Array

- Dùng để tạo ra một mảng mới bằng cách gọi hàm truyền vào (callback function) với mỗi phần tử của mảng.
- Phương thức này không thay đổi giá trị của mảng ban đầu.

```
const numbers = [2, 3, 4, 5];
```

```
const doubles = numbers.map(number => number * 2);
```



## Phương thức filter() của Array

- Dùng để tạo ra một mảng mới với những phần tử thỏa mãn điều kiện của hàm truyền vào (callback function).
- Phương thức này không thay đổi giá trị của mảng ban đầu.

```
const words = ['apple', 'microsoft', 'samsung', 'intel'];  
  
const longWords = words.filter(word => word.length > 5);
```



## Phương thức reduce() của Array

- Dùng để tạo ra một kết quả duy nhất bằng cách thực hiện hàm truyền vào (callback function) trên mỗi phần tử trong mảng.
- Phương thức này không thay đổi giá trị của mảng ban đầu.

```
const array = [2, 3, 4, 5];  
  
const total = array.reduce((sum, value) => sum + value);
```

## map, filter, and reduce explained with emoji 🤔

```
map([🐮, 🍌, 🐔, 🌽], cook)  
=> [🍔, 🍟, 🍗, 🍿]
```

```
filter([🍔, 🍟, 🍗, 🍿], isVegetarian)  
=> [🍟, 🍿]
```

```
reduce([🍔, 🍟, 🍗, 🍿], eat)  
=> 🤮
```

Minh họa cho map, filter và reduce



---

# Bài tập thực hành



## Bài tập thực hành

- Viết hàm truyền vào một mảng các số và tính ra tổng các số đó.
- Cho một mảng các số nguyên bất kỳ, dùng phương thức **map()** tạo ra một mảng mới với các số là số dư tương ứng khi chia các số trong mảng cũ cho **2**.
- Dùng phương thức **filter()** lọc ra các số chẵn trong một mảng nhiều số bất kỳ.
- Cho một mảng các chuỗi là tên các học viên. Viết hàm lọc ra những học viên có tên bắt đầu bằng chữ **H** (không phân biệt hoa thường).

# Bài tập về nhà

---



## Bài tập về nhà

- Ôn lại lý thuyết phần **Vòng lặp**.
- Học và làm bài tập phần **Điều kiện, rẽ nhánh phần 1**.