

JavaScript String Exercises

Basic

1. Viết hàm `capitalize(str)` nhận vào một chuỗi, trả về chuỗi chỉ viết hoa ký tự đầu tiên, tất cả ký tự khác viết thường, loại bỏ khoảng trắng ở 2 đầu nếu có

```
1 capitalize("hello world"); // Hello World
2 capitalize("javascript is fun"); // Javascript is fun
```

2. Viết hàm `title(str)` nhận vào một chuỗi, trả về chuỗi dạng viết hoa chữ cái đầu tiên của từng từ, những chữ cái khác phía sau viết thường, lưu ý loại bỏ phân khoảng trắng ở 2 đầu nếu có.

```
1 title("hello world"); // Hello World
2 title(" ba nGuyễn "); // Ba Nguyễn
```

3. Viết hàm `protectEmail(email)` nhận vào một chuỗi là địa chỉ email, trả về chuỗi email dạng rút gọn, chỉ hiển thị 2 ký tự đầu và thay thế các ký tự khác trước @ dấu ...

```
1 protectEmail("banx9x@gmail.com"); // ba...@gmail.com
2 protectEmail("phoebedo@gmail.com"); // ph...@gmail.com
```

4. Viết hàm `reverse(str)` nhận vào một chuỗi, trả về chuỗi đảo ngược tất cả ký tự

```
1 reverse("hello"); // olleh
2 reverse("béo ú"); // ú oéb
```

5. Viết hàm `countVowel(str)` nhận vào một chuỗi, trả về số ký tự nguyên âm có trong chuỗi, các ký tự nguyên âm là aeiou

```
1 countVowel('hello'); // 3
```

6. Viết hàm `reverseWords(str)` nhận vào một chuỗi, trả về chuỗi đảo ngược thứ tự các từ (không phải toàn bộ ký tự)

```
1 reverseWords('The fox is coming for the chicken'); // chicken the  
   coming is fox The
```

7. Viết hàm `randomHex()` trả về một giá trị màu HEX ngẫu nhiên, mã màu HEX là hệ cơ số 16 (0-9a-f) và có dạng `#00ffaa`

```
1 randomHex(); // #0abd4f => kết quả có thể khác nhau
```

8. Viết hàm `parameterize(str)` nhận vào một chuỗi bất kỳ (chỉ chứa các từ không dấu), trả về chuỗi dạng url dạng viết thường và các từ được nối bằng các dấu -

```
1 parameterize("Hello World"); // hello-world  
2 parameterize(" javascript is very funny "); // javascript-is-very-  
   funny
```

9. Viết hàm `isPalindrome(str)` nhận vào một chuỗi bất kỳ, kiểm tra và trả về kết quả chuỗi gốc và chuỗi đảo ngược có giống nhau hay không, không phân biệt hoa thường và không tính dấu khoảng trắng

```
1 isPalindrome("Race car"); // racecar == racecar => true  
2 isPalindrome("Ba"); // ba != ab => false
```

Intermediate

1. Viết hàm `mostCommonCharacter(str)` tìm và trả về ký tự xuất hiện nhiều lần nhất trong chuỗi, nếu có nhiều ký tự có số lần xuất hiện giống nhau thì trả về ký tự đứng trước theo bảng alphabe

```
1 mostCommonCharacter("Ba Nguyen"); // { n: 2 }  
2 mostCommonCharacter("Hello Worldo"); // { l: 3, o: 3 }
```

2. Viết hàm `sortByName(arr)` nhận vào một mảng các objects có tên học viên, trả về một mảng mới được sắp xếp theo tên (từ cuối cùng trong tên) theo thứ tự alphabe

```
1  const students = [  
2    { name: "Hoàng Thị Ngọc Linh", point: 0 },  
3    { name: "Bùi Lan Nhi", point: 0 },  
4    { name: "Nguyễn Ba", point: 10 }  
5  ];  
6  
7  sortByName(students);  
8  // kết quả  
9  // [  
10 //   { name: "Nguyễn Ba", point: 10 },  
11 //   { name: "Hoàng Thị Ngọc Linh", point: 0},  
12 //   { name: "Bùi Lan Nhi", point: 0}  
13 // ]
```