

**Langage obligatoire :**JavaScript (aucun autre langage autorisé)

## Consignes

- Vous devez rendre chaque étape dans un fichier portant son nom, et l'extension '.js'.
  - Vous serez corrigés par une moulinette, pensez à respecter les consignes.
  - De plus, vous devez gérer **toutes les erreurs possibles** (mauvais paramètre ou mauvais type y compris)
  - Vous n'avez le droit qu'au JavaScript.
  - Votre code doit être **propre** et épuré.
  - Ne laissez pas du code inutile et optimisez un maximum
  - Vous n'avez pas le droit d'utiliser de bibliothèque externe
  - Copier du code d'internet ou d'un autre étudiant est strictement interdit et résultera d'un -42.
- 
- `console.log()`
  - `Math.*`
  - `Object.hasOwnProperty()`

## Introduction

Tous les exemples proposés sont exécutés à partir de la console du navigateur, d'où les '>' délimitant les instructions

## Etape 1

Ecrire une fonction avec le prototype suivant :

```
function myEach(list, func) {};
```

Cette fonction itère sur chaque élément d'une liste (*list*) et les passe en paramètre à la méthode *func*.

Exemple :

```
> var arr = [1, 2, 3];  
  
> var test = function(value){  
    alert(value);  
}  
  
> myEach(arr, test);
```

Donnera :

=> Effectue un alert sur chaque élément

Autre exemple :

```
> var obj = {"pol": 2, "lol": 4, "tol": 6};  
  
> var test = function(value){  
    console.log(value);  
};
```

```
}  
> myEach(obj, test);
```

Donnera :

```
> 2  
4  
6
```

## Etape 2

Ecrire une fonction avec le prototype suivant :

```
function myReject (arr, func) {};
```

Cette fonction prend un array et une fonction en paramètres.

Elle va appeler la fonction 'func' en lui passant chaque élément du tableau en paramètre. La fonction retourne un array contenant tout les éléments pour lequel le retour de *func* est false.

Exemple :

```
> var arr = [1, 2, 3, 4, 5, 6];  
  
> var isPair = function (num) {  
    return num % 2 === 0;  
};  
  
> myReject(arr, isPair);
```

Donnera :

```
> [1, 3, 5]
```

## Etape 3

Ecrire une fonction avec le prototype suivant :

```
function myMap (arr, func) {};
```

Cette fonction prend un array, une méthode en paramètres.

Elle va appeler la fonction 'func' pour chaque élément du tableau en le passant en paramètre, et va push le retour de chaque appel de la fonction dans un nouveau tableau.

La fonction retourne ce tableau si il n'y a pas d'erreurs, sinon elle retourne null.

Exemple :

```
> var arr = [0, 1, 2, 3, 4];  
  
> var add_two = function (num) {  
    return num + 2;  
};  
  
> myMap(arr, add_two);
```

Donnera :

```
> [2, 3, 4, 5, 6]
```

Autre exemple :

```
> var arr = [3, 42, 72, -1, 8/8];  
  
> var sqrt = function (num) {  
    if (num < 0){  
        console.log('Error: Argument must be greater or equal to zero')  
        return null;  
    }  
    return Math.sqrt(num);  
};  
  
> myMap(arr, sqrt);
```

Donnera :

```
> Error: Argument must be greater or equal to zero  
> null
```

## Etape 4

Ecrire une fonction avec le prototype suivant :

```
function myContain (arr, value, [indexStart]) {};
```

Cette fonction prend un array, une valeur et un paramètre optionnel qui permet de commencer la recherche à un index choisi.

La fonction retourne true si l'array *arr* contient la variable *value*, et false si ce n'est pas le cas.

Exemple :

```
> var arr = [1, 2, 3, 4, 5, 6];  
  
> myContain(arr, 2);
```

Donnera :

```
> true
```

Autre exemple :

```
> var arr = [1, 2, 3, 4, 5, 6];  
  
> myContain(arr, 42);
```

Donnera :

```
> false
```

Autre exemple :

```
> var arr = [1, 2, 3, 4, 5, 6];
```

```
> myContain(arr, 4, 2);
```

Donnera :

```
> true
```

## Etape 5

Ecrire une fonction avec le prototype suivant :

```
function myFind (list, func) {};
```

Cette fonction prend une liste et un prédicat (valeur ou méthode) en paramètres.

La fonction passe sur chaque élément de la liste et retourne le premier élément qui passe le test avec le prédicat ou *undefined* si les éléments ne passent pas le test.

La fonction retourne l'élément dès qu'elle match, il ne faut donc pas parourir toute la liste.

Exemple :

```
> var myList = [1, 2, 3, 4, 5, 6];
```

```
> var isPair = function(num){  
    return num % 2 == 0;  
}
```

```
> myFind(myList, isPair);
```

Donnera :

```
> 2
```

Autre exemple :

```
> var arr = [1, 2, 3, 4, 5, 6];
```

```
> myFind(myList, 42);
```

Donnera :

```
> undefined
```

Autre exemple :

```
> var list = {"lol":1, "pol":3, "poule":5};
```

```
> var test = function(test){ return test == 3; };
```

```
> myFind(list, test);
```

Donnera :

```
> 3
```