

## Exercice 1 : .map()

Développer une fonction **MineurMajeur()** qui prend en paramètre une **liste** et puis retourne une nouvelle **liste** avec un champ supplémentaire qui indique si la personne est “**majeur**” si son âge est **supérieur ou égal 18** ou bien “**mineur**” si son âge est **strictement inférieur à 18** .

exemple :

on vous donne :

```
const persons = [
  { name : "MONCEF" , age: 13} ,
  { name : "BAHIJA" , age: 23} ,
  { name : "ALA" , age: 18} ,
];
```

**MineurMajeur(persons)** doit retourner :

```
[
  { name : "MONCEF" , age: 13 , status : "mineur"} ,
  { name : "BAHIJA" , age: 23 , status : "majeur"} ,
  { name : "ALA" , age: 18 , status : "majeur"} ,
]
```

## Exercice 2 : .map() .split() .join()

Développer une fonction **FormatToArray( str1 )** qui prend en paramètre une chaîne de caractère **str1** content une suite de nom&prenom concaténé par le symbole “-” et puis retourne une nouvelle liste d’objet contenant le nom et prénom qui se trouve dans la chaîne .

exemple :

```
const str1 = 'ahmed&ferah-moez&saadi-fadhel&gadihi';
```

**FormatToArray( str1 )** doit retourner :

```
[
  { firstname: 'ahmed', lastname: 'ferah' },
  { firstname: 'moez', lastname: 'saadi' },
  { firstname: 'fadhel', lastname: 'gadihi' },
]
```

### Exercice 3 : .map() .includes()

Développer une fonction **MoodDetect( liste )** qui prend en paramètre une **liste** d'objets contenant le **nom** de l'utilisateur et son **commentaire** par rapport a un post quelconque , la fonction doit détecter d'après chaque commentaire de chaque personne son **état** émotionnelle .

exemple :

```
const users = [
  { name : "sarah" , comment : "j'aime ce poste <3" },
  { name : "monya" , comment : "le poste est null :( " },
  { name : "carlos" , comment : "salut l'algerie" }
]
```

**MoodDetect( users )** doit retourner :

```
[
  { name : "sarah" , etat : "heureux" },
  { name : "monya" , etat : "triste" },
  { name : "carlos" , etat : "neutre" }
]
```

### Exercice 4 : .map() .sort() .filter() .reduce()

Développer une fonction **FailedStudent( liste )** qui prend en paramètre une **liste** d'objets contenant le **nom** de l'étudiant et sa **note** , puis cette fonction doit filtrer **les étudiant sans note** et retourner une nouvelle list **triée par note** avec un nouveau champ "**mention** " indiquant pour chaque étudiant :

- si sa note est supérieur ou égal à **la moyenne** de la classe => **bien**
- si sa note est strictement inférieur à **la moyenne** de la classe => **passable**

INPUT :

```
const etudiants = [
  { name : "sarah" , note : undefined },
  { name : "monya" , note : 7 },
  { name : "carlos" , note : 13 },
  { name : "fadhel" , note : 18 },
]
```

OUTPUT :

```
[
  { name : "fadhel" , note : 18 , mention : "bien" },
  { name : "carlos" , note : 13 , mention : "passable" },
  { name : "monya" , note : 7 , mention : "passable" },
]
```