

2.1) Programmez une fonction régulière nommée `copyObject(object)` qui effectue une copie d'un objet passé en paramètre et que vous pouvez tester de cette façon :

```
5  const user1 = {
6    fullname: "Marc Tremblay",
7    username: "Furry99",
8    password: "pa$$word"
9  }
10
11 const user2 = copyObject(user1)
12
13 user2.fullname = "Francis Roberge"
14 console.log(user1)
15 console.log(user2)
```

```
9
10
11
12
13 'Francis Roberge'
14 {
15   fullname: 'Marc Tremblay',
16   username: 'Furry99',
17   password: 'pa$$word'
18 }
19 {
20   fullname: 'Francis Roberge',
21   username: 'Furry99',
22   password: 'pa$$word'
23 }
```

2.2) Ajoutez une fonction `fullname()` à `user1` utilisant un « template literal » qui retournera. « Je me nomme Marc Tremblay ».

```
1  const user1 = {
2    firstname: "Marc",
3    lastname: "Tremblay"
4  }
```

```
1
2
3
4
```

2.3) Programmez une fonction `getRandomValue(object)` qui retourne la valeur d'une propriété prise au hasard dans la liste d'un objet quelconque.

```
7  const user1 = {
8    fullname: "Marc Tremblay",
9    username: "Furry99",
10   password: "pa$$word"
11 }
12
13 console.log(getRandomValue(user1))
14
```

```
7
8
9
10
11
12
13 'pa$$word'
```

2.4) Programmez une fonction `getValue(key, object)` ayant deux paramètres et retournant la valeur de la clé demandé dans l'objet si elle existe ou « Propriété [key] inexistante! » si non trouvée.

```
9  const user1 = {
10     fullname: "Marc Tremblay",
11     username: "Furry99",
12     password: "pa$$word"
13 }
14
15 console.log(getValue("fullname", user1))
16 console.log(getValue("fullname2", user1))
17
```

```
9
10
11
12
13
14
15 'Marc Tremblay'
16 'Propriété fullname2 inexistante!'
```

2.5) Reprenez le numéro 2.4 mais en ne créant qu'une seule **instruction** dans la fonction.

N.B. Il est possible qu'une seule instruction prenne plus d'une ligne à l'écran au besoin.

2.6) Écrivez une fonction qui supprime complètement les propriétés d'objet dont la valeur contient un caractère passé en paramètre.

```
10 const user1 = {
11     fullname: "Marc Tremblay",
12     username: "Furry99",
13     password: "pa$$word"
14 }
15
16 console.log(deleteFind("a", user1))
```

```
10
11
12
13
14
15
16 { username: 'Furry99' }
```

2.7) Écrivez une fonction qui affiche l'élément de la [suite de Fibonacci](#) à un nombre entré en paramètre.

```
7 console.log(fibonacci(10))
```

```
7 89
```

2.8) Reprenez le numéro précédent mais avec une [fonction récursive](#).

```
7 console.log(fibonacci(10))
```

```
7 89
```