



Révisions JavaScript

Tableaux et Objets en JS - JSON

Dans ce TD, vous allez vous familiariser avec la manipulation des tableaux et les Objets en JavaScript.

Exercice 1. Les bases des tableaux JS JavaScript dispose d'un type `Array` ayant les propriétés suivantes :

- `length` pour avoir sa taille
- `join()` ou `toString()` pour afficher un tableau (voir la doc de `join()` sur developer.mozilla.org)
- `join()` fait en gros l'inverse de ce que fait la méthode `split()` de `String` en permettant d'obtenir un tableau à partir d'une chaîne.
- `sort()` et `reverse()` pour trier ou inverser l'ordre d'un tableau

Vous trouverez une documentation sur les tableaux à l'adresse : https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/Array

Ecrire des petites fonctions JS qui partant d'un tableau de chaînes :

```
1 const etudiants = ['Jules', 'agatha', 'asterix', 'Idefix', 'Zazou', 'Robert'];
```

1. l'affichent dans une liste à puce
2. l'affichent trié à l'aide de `sort()`
3. Pour améliorer le tri pour qu'il soit insensible à la casse, donner en paramètre à `sort()` une fonction qui renvoie `+1`, `-1` ou `0` à la manière du `strcmp()` du C ou du `compareTo()` de Java.
4. Inverser ce tri.

— Comment sélectionner les éléments entre les indices 2 et 4 ?

— et l'avant dernier élément ? (`slice()`)

— Que se passe-t-il si on fait :

```
1 etudiants[100] = 'Jojo';
```

dans notre tableau ?

— et si on fait ?

```
1 etudiants.length = 0;
```

— Que font les méthodes `push`, `pop` et `shift` des tableaux ? Testez-les.

Exercice 2. Parcourir et manipuler les tableaux JS On a plusieurs manières de parcourir un tableau en JavaScript :

```
1 const etudiants = ['Jules', 'agatha', 'asterix', 'Idefix', 'Zazou', 'Robert'];
2 // Boucle classique
3 for(let i=0; i < etudiants.length; i++){
4   console.log(etudiants[i].toLowerCase());
5 }
```

ou avec `forEach` :

```
1 etudiants.forEach(function(nom){
2   console.log(nom.toUpperCase());
3 });
4 // pb dans ce dernier cas
5 // pour arrêter la boucle : usage des exceptions
6 // dans la boucle classique : break suffit
```

On peut aussi utiliser `map()` et `filter()` :

```
1 // map
2 etuds = etudiants.map(x => x.toLowerCase());
3 for (let i=0; i<etuds.length; i++){
4   console.log(etuds[i]);
5 }
6 // filter (noms jusqu'à la lettre I)
7 etuds = etudiants.filter(x => (x.toLowerCase() < "k"));
```

- En utilisant `String.prototype`, ajouter à `String` une fonction `reverse()` inversant une chaîne sur place en utilisant `split()` et `join()`.
- Construire avec `map` un tableau contenant tous les noms d'étudiants à l'envers en minuscules.
- Avec `filter`, filtrez les noms de longueur 5

Ou encore `reduce()` :

```
1 const nums = [1,2,3,4,5,6];
2 let prod = nums.reduce((x,y) => x*y ,1);
```

Voir https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/Reduce

- Calculer le minimum, maximum, moyenne d'un tableau de nombres avec `reduce()`
- Comment tester si un tableau de nombres ne comporte que des nombres pairs avec la méthode `every()`.
- Faire de même pour tester l'existence d'un nombre pair dans un tableau avec `some`

Exercice 3. Les classes en EcmaScript 6

Vous trouverez une documentation sur les classes en ES6 à l'adresse : <https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Classes>

Définissons une classe `Personne` de la manière suivante :

```
1 class Personne{
2     constructor(id, prenom, nom){
3         this._id = id;
4         this._nom = nom;
5         this._prenom = prenom;
6     }
7     toString(){
8         return `${this._nom} ${this._prenom}`;
9     }
10    get nom() {
11        return this._nom.toLowerCase();
12    }
13
14    set nom(newNom){
15        if(newNom){
16            this._nom = newNom;
17        }
18    }
19 };
20 let p = new Personne(3, 'John', 'Doe');
21 console.log(p);
22 p.nom='Azerty';
23 console.log(p.nom);
```

1. A quoi servent les méthodes `get` et `set` ?
2. Construire maintenant une classe `Contacts` qui contient une liste de `Personnes` et possède un nom et une date de dernière mise à jour.
3. Affichez la représentation JSON d'une instance de la classe `Contacts`.