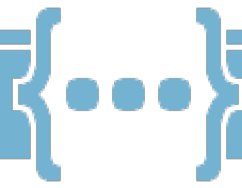


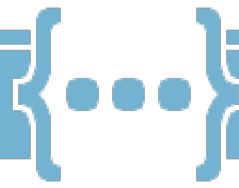
Cours 10

Splice et modulo

Intro. à la programmation - Aut. 2021



- ❖ Splice (Tableaux)
 - ◆ Retirer des éléments
 - ◆ Ajouter des éléments
- ❖ Rappel : modulo



❖ Fonction `splice()`

- ◆ Permet de retirer / ajouter des éléments dans un **tableau**
 - Et pas juste à la fin comme `push()` et `pop()` !
- ◆ Syntaxe pour **retirer** des éléments

`monTableau.splice(index, nbRetirer)`

Premier élément à retirer

Combien d'élément on retire au total ?

◆ Exemple :

```
let couleurs = ["Bleu", "Rouge", "Jaune", "Vert", "Orange", "Violet"];
couleurs.splice(0, 2);
// couleurs vaut maintenant ["Jaune", "Vert", "Orange", "Violet"]
```

- L'élément à l'**index 0** est le premier à être retiré et est inclus dans le nombre total d'éléments à retirer.



❖ Fonction `splice()`

◆ Retirer des éléments

◆ Autre Exemple :

```
let couleurs = ["Bleu", "Rouge", "Jaune", "Vert", "Orange", "Violet"];
couleurs.splice(2, 1);
// couleurs vaut maintenant ["Bleu", "Rouge", "Vert", "Orange", "Violet"]
```

◆ "Jaune" était l'élément à l'index 2 et un seul élément devait être retiré, au total.



❖ Fonction `splice()`

- ◆ Ajouter des éléments
- ◆ Syntaxe pour ajouter des éléments

`monTableau.splice(index, nbRetirer, valeur1, valeur2, ...)`

Premier élément décalé

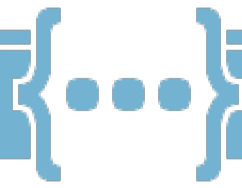
Combien d'élément on retire au total ?
Ceci doit être 0 !

Valeurs à ajouter dans le tableau.

◆ Exemple :

```
let couleurs = ["Bleu", "Rouge", "Jaune"];  
couleurs.splice(1, 0, "Rose", "Turquoise");  
// couleurs vaut maintenant ["Bleu", "Rose", "Turquoise", "Rouge", "Jaune"]
```

- L'élément à l'`index` spécifié ("**Rouge**") et les suivants sont donc **décalés vers la droite** et les nouveaux éléments sont insérés juste avant.



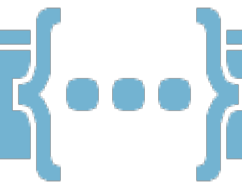
❖ Fonction **splice()**

◆ Ajouter des éléments

◆ Autre Exemple :

```
let couleurs = ["Bleu", "Rouge", "Jaune"];  
couleurs.splice(0, 0, "Magenta");  
// couleurs vaut maintenant ["Magenta", "Bleu", "Rouge", "Jaune"]
```

◆ "Bleu" était l'élément à l'index 0 et on a donc ajouté l'élément "Magenta" juste avant.



❖ Fonction `splice()`

- ◆ Ajouter ET retirer des éléments
- ◆ Syntaxe pour ajouter des éléments

```
monTableau.splice(index, nbRetirer, valeur1, valeur2, ...)
```

Premier élément à retirer

Combien d'élément on retire au total ?

Valeurs à ajouter dans le tableau.

◆ Exemple :

```
let couleurs = ["Bleu", "Rouge", "Jaune", "Vert"];  
couleurs.splice(1, 2, "Beige", "Lilas");  
// couleurs vaut maintenant ["Bleu", "Beige", "Lilas", "Vert"]
```

- Les éléments **"Rouge"** et **"Jaune"** ont été retirés car ce sont les **2** premiers éléments à partir de l'**index 1**. On a ensuite mis **"Beige"** et **"Lilas"** à la place.



❖ Fonction `splice()`

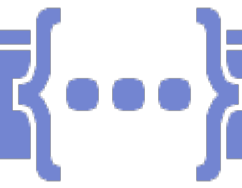
◆ Récupérer les éléments retirés du tableau dans une variable.

- Un peu comme avec `pop()`, `splice()` permet de stocker les **éléments** qui viennent d'être **retirés** du tableau dans une variable.
- La variable contiendra un nouveau tableau avec le ou les **éléments retirés**.
- Bien entendu, nous ne sommes pas obligés de stocker ces **éléments retirés**.

◆ Exemple :

```
let couleurs = ["Bleu", "Rouge", "Jaune", "Vert"];  
let couleurs2 = couleurs.splice(1, 2);  
// couleurs vaut maintenant ["Bleu", "Vert"]  
// couleurs2 vaut ["Rouge", "Jaune"]
```

- Les éléments **"Rouge"** et **"Jaune"** ont été retirés car ce sont les **2** premiers éléments à partir de l'**index 1**. On les a donc stockés dans la variable **couleurs2** à l'intérieur d'un nouveau tableau.



❖ Opérateur modulo

- ◆ Permet d'obtenir le « restant de la division ».

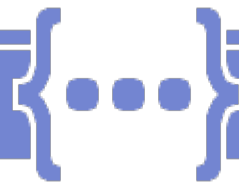
- ◆ Exemple :

```
let nombre = 14 % 6;  
// nombre vaut 2.  
let nombre2 = 7 % 4;  
// nombre2 vaut 3.
```

- ◆ $14 \% 6$ -> On tente de soustraire 6 à 14 le plus de fois possible. Dès que le résultat est inférieur à 6, on a notre restant.

- 1) $14 - 6$ donne 8
- 2) $8 - 6$ donne 2
- 3) 2 est inférieur à 6. C'est donc le restant.

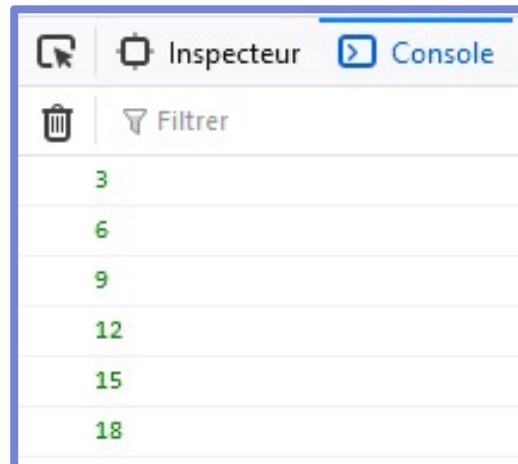
- ◆ Ainsi, si on a $x \% 6$, le restant de x sera forcément un nombre entre 0 et 5.



❖ Opérateur modulo

- ◆ Exemple : On imprime dans la console du navigateur tous les nombres de 1 à 20 qui sont des multiples de 3. (3, 6, 9, 12, 15 et 18)

```
for(let index = 1; index < 21; index++){  
    if(index % 3 == 0){  
        console.log(index);  
    }  
}
```



Expression	Valeur
1 % 3	1
2 % 3	2
3 % 3	0
4 % 3	1
5 % 3	2
6 % 3	0
7 % 3	1