

# JavaScript – Programmation avancée

**Intervenant : Rémy PIERRE**



A photograph of a young monkey sitting on the back of an adult monkey. The young monkey is looking directly at the camera with a wide-eyed, curious expression. The adult monkey's head and back are visible in the foreground, with its fur appearing brown and shaggy. The background is a blurred green forest. A semi-transparent dark green banner is overlaid on the left side of the image, containing the text 'Module 5' and 'Les tableaux' in white.

Module 5

# Les tableaux

Déclarer un tableau

Ajouter un élément

Parcourir un tableau

Effacer un élément

Echantillonné un tableau

# Déclarer un tableau de chaîne de caractères

- Un tableau doit être instancier !
- **console.table** est assez pratique pour debugger

```
const tableau =[];  
tableau[0]=42;  
tableau[1]=51;  
tableau.push(78); // tableau[2]=78
```

```
// créer un tableau prérempli  
const tableau2 =[12,23,54];
```

```
console.log(tableau);  
console.table(tableau);
```

**push** est la méthode recommandée  
pour ajouter

# Tableau de chaîne de caractères

- Même utilisation de **push** avec les chaînes de caractères

```
// tableau de chaînes de caractères
const fruits = ['pomme', 'poire', 'cerise'];
fruits.push('kiwi'); // fruits[3]='kiwi';

const couleurs = [];
couleurs.push('rouge');
couleurs.push('vert');
```

# Afficher ou parcourir un tableau

- La boucle for "à papa" !

```
const fruits = ['pomme', 'poire', 'cerise'];  
  
// afficher un tableau : boucle for classique  
for(let i =0; i<fruits.length;i++){  
    console.log(fruits[i]);  
}
```

# Afficher ou parcourir un tableau

- La boucle **for each**
- Très utilisée et simple à écrire.

```
const fruits = ['pomme', 'poire', 'cerise'];  
  
// afficher un tableau : boucle for each  
for (let f of fruits){  
    console.log(f);  
}
```



# Afficher ou parcourir un tableau

- La boucle **map**
- Très utilisée avec React

```
const fruits = ['pomme', 'poire', 'cerise'];  
// le map  
fruits.map (f => {console.log(f)});  
// le map avec indice  
fruits.map ((f,i) => {console.log(fruits[i])});
```





# Afficher ou parcourir un tableau

- La boucle **for avec in**
- Plutôt utiliser pour parcourir les objets.
- Et oui un objet , c'est aussi un tableau en Javascript !!

```
const fruits = ['pomme', 'poire', 'cerise'];  
// indice => string  
// En effet cette boucle sert aussi à parcourir un objet  
for (let indice in fruits){  
    console.log(fruits[parseInt(indice)]);  
}
```

# Afficher ou parcourir un tableau

- Pourquoi la boucle **for** avec **in** let attribut est un **string** ?
- En effet cette boucle sert surtout à parcourir un objet.

```
// JSON
let personne = {prenom:'Brad',nom:'PITT'};
for (let attribut in personne){
    console.log(attribut); // prenom ... nom
}
```

# Effacer un élément

- INTERDIT !
- Le **delete** fait des trous sur un tableau

```
const fruits = ['pomme', 'poire', 'cerise', 'kiwi'];  
// effacer 'cerise' indice 2  
delete fruits[2]; // NE PAS UTILISER
```



- Le **delete** est utilisé sur les objets pour enlever des attributs

```
const personne = {prenom: 'Brad', nom: 'PITT', age: 18};  
delete personne['age']; // Et oui un objet est aussi un tableau !!  
//personne = {prenom: 'Brad', nom: 'PITT'};
```



# Effacer un élément

- La bonne pratique
- Dite **splice** avec un **p**

```
const fruits = ['pomme', 'poire', 'cerise', 'kiwi'];  
fruits.splice(2, 1);
```



splice

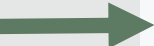
**1** : Nombre d'éléments à effacer

**2** : **indice** à effacer

Ou indice à partir duquel effacer

# Échantillonné un tableau

- Extraire un tableau depuis un autre tableau avec **slice**
- **slice** sans le **p**
- Le premier tableau reste un intacte



```
const fruits = ['pomme', 'poire', 'cerise', 'kiwi'];  
// slice = echantilloné un tableau  
let tableau2 = fruits.slice(2,4); // 2 = indice de depart ; 4 = indice de fin  
// le premier tableau reste intacte  
console.log(fruits); // fruits = ['pomme', 'poire', 'cerise', 'kiwi'];  
console.log(tableau2); // tableau2 = ['cerise', 'kiwi'];
```

slice

**2** : indice de départ

**4** : indice de fin