

JAVASCRIPT

Durée Totale du Module : 21H

Auteur :

Jean-François Pech

Date création :

03/03/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

Boucle While / For / ForEach / For ...of / For ...in / map

Comme dans tous les langages de programmation Javascript a un système de boucle, de base cela va nous permettre de répéter des instructions de code selon une condition. Les boucles vont également s'avérer utile par la suite, pour parcourir des itérables comme des tableaux ou des objets.

While

Correspond à répéter une ou plusieurs instructions TANT QUE une condition est vraie.

```
let unIndex = 0;
while (unIndex < 10) {
  console.log("Le Nombre : " + unIndex);
  unIndex++;
};
```

Ci-dessus on a un index initialisé à 0 et TANT QUE cet index est strictement inférieur à 10 ALORS, on va faire un console.log(), puis ne pas oublier d'incrémenter l'index pour pouvoir passer à une itération de boucle suivante.

Le Nombre : 0	app.js:20
Le Nombre : 1	app.js:20
Le Nombre : 2	app.js:20
Le Nombre : 3	app.js:20
Le Nombre : 4	app.js:20
Le Nombre : 5	app.js:20
Le Nombre : 6	app.js:20
Le Nombre : 7	app.js:20
Le Nombre : 8	app.js:20
Le Nombre : 9	app.js:20

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

FOR

Autre manière de créer des boucles, avec `for()`, dans les paramètres on va pouvoir directement initialiser un index, définir une condition et incrémenter l'index dans l'exemple ci-dessous nous allons faire une boucle visant à parcourir chaque case d'un tableau pour l'afficher en console.

```
let listeFilm = ['Ultime Décision', 'Mission Alcatraz', 'Attack Force'];  
/** Boucle for, on définit un index (ici c'est i),  
/** puis on définit une condition qui va définir le nombre de fois que le code dans la  
boucle sera exécutée  
/** puis on définit comment on passe à la prochaine itération de la boucle (ici i++, on  
augmente de 1 le numéro de la case que l'on console.log)  
for(let i=0; i<listeFilm.length; i++){  
    console.log('boucle FOR : ', listeFilm[i]);  
};
```

```
boucle FOR :   Ultime Décision      app.js:14  
boucle FOR :   Mission Alcatraz     app.js:14  
boucle FOR :   Attack Force         app.js:14
```

Auteur :

Jean-François Pech

Date création :

03/03/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

ForEach

Une autre alternative, la fonction `forEach` de JS automatise le parcours d'un tableau ou objet (sans que l'on ait à gérer un système d'indexation (`i++`)).
`forEach` va prendre en paramètre une fonction, cette même fonction pourra avoir un paramètre qui correspondra à chaque case parcourue. (Généralement dans la parenthèse de `forEach` on passe une fonction fléchée).

```
let listeFilm = ['Ultime Décision', 'Mission Alcatraz', 'Attack Force'];
//? La méthode forEach() permet d'exécuter une fonction donnée sur chaque élément du tableau.
// ? On va choisir une variable temporaire pour parcourir les éléments du tableau
listeFilm.forEach(unFilm => console.log('boucle forEach Tableau : ', unFilm));
```

Ici chaque case du tableau sera stockée temporairement dans `unFilm`.

boucle forEach Tableau : Ultime Décision	app.js:27
boucle forEach Tableau : Mission Alcatraz	app.js:27
boucle forEach Tableau : Attack Force	app.js:27

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

For ... of

Encore une alternative pour parcourir des variables tableaux (et autre) c'est la boucle for ... of, qui de la même manière que dans l'exemple précédent dans lequel on va définir une variable temporaire pour parcourir chaque case du tableau :

```
for(let unElementTablo of listeFilm){
  console.log('boucle FOR...OF : ',unElementTablo);
};
```

boucle FOR...OF :	Ultime Décision	app.js:35
boucle FOR...OF :	Mission Alcatraz	app.js:35
boucle FOR...OF :	Attack Force	app.js:35

Auteur :

Jean-François Pech

Date création :

03/03/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

For ... in

Si l'on prend le cas des objets JS propose aussi un équivalent à for of (pour les variables de type Array), les boucles for in qui ont exactement la même utilisation que l'exemple précédent

```
const userData = {
  name: 'John Doe',
  email: 'john.doe@example.com',
  age: 25,
  dob: '08/02/1989',
  active: true
};
```

Il faut définir une variable temporaire qui stockera les clés (propriétés) de l'objet

```
// on définit une variable temporaire pour parcourir le objet :)
for(let cleObjet in userData){
  console.log(`boucle FOR...IN (objet) : clé:${cleObjet} - valeur : $
{userData[cleObjet]} `);
};
```

Ici durant le parcours de l'objet chaque propriété ou clé seront stockées temporairement dans la variable cleObjet.

Rappel : pour accéder aux propriétés d'un objet on la notation en tableau associatif
 unObjet[quelque chose]

```
boucle FOR...IN (objet) : clé:name - valeur : John Doe
boucle FOR...IN (objet) : clé:email - valeur : john.doe@example.com
boucle FOR...IN (objet) : clé:age - valeur : 25
boucle FOR...IN (objet) : clé:dob - valeur : 08/02/1989
boucle FOR...IN (objet) : clé:active - valeur : true
```

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

Depuis sa version ES8 JS propose des fonctions utilisables sur les Objets qui vont pouvoir transformer leurs clés et ou leurs valeurs sous forme de tableau (pour utiliser les f° `forEach` ou `map` par exemple).

```
// De fait, une fois les objets convertis en tableau on peut ruser et utiliser forEach
par exemple :
valuesUser.forEach((lesValeurs)=>{
    console.log(`FOREACH avec objet converti en tableau chaque valeurs : ${lesValeurs}`);
});
convertedDataUser.forEach(([key, value])=>{
    console.log(`FOREACH avec objet converti en tableau : ${key}: ${value}`);
});
```

Exercice : boucles

```
// TODO :JS map phase 1  
// TODO : côté template html rajouter plein de <p></p>  
// TODO :On va récupérer TOUS les <p> de notre page dans une  
variable lesTxt via getElementsByTagName  
// TODO :On va faire un console log de lesTxt
```

```
//TODO JS map Phase 2  
//TODO Avec la methode Array.from(), dans une nouvelle variable  
textesTab on va transformer notre htmlCollection en array  
//TODO On console log la variables textesTab  
//* On transforme le HTMLCollection(qui contient tous nos <p>) en  
Array classique
```

```
//TODO JS Map Phase 3 (on peut travailler sur un Array)  
//TODO Sur textesTab on va utiliser la f° map(),  
//TODO dans map(), on va lui passer en param une fonction fléchée  
qui elle a en paramètre une variable temporaire  
(nom de la variable au choix)  
//TODO cette fonction fléchée elle va modifier le innerHTML ou  
innerText de la variable temporaire
```

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

Solution

```
const lesTxt = document.getElementsByTagName("p");  
console.log(lesTxt);
```

```
/* On transforme le HTMLCollection (qui contient tous nos <p>) en  
Array classique  
const textesTab = Array.from(lesTxt);  
console.log(textesTab);
```

```
textesTab.map(uneCase => uneCase.innerHTML = "LOL JE SUIS  
HACKERMAN" );
```

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

i http://127.0.0.1:5500

Les système de boucles

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Totam et rem accusantium dicta voluptatum quam suscipit, molestiae dolor, nihil asperiores corrupti, quisquam ducimus! Sint, ipsa. Voluptates adipisci facilis veniam impedit?

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Totam et rem accusantium dicta voluptatum quam suscipit, molestiae dolor, nihil asperiores corrupti, quisquam ducimus! Sint, ipsa. Voluptates adipisci facilis veniam impedit?

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Totam et rem accusantium dicta voluptatum quam suscipit, molestiae dolor, nihil asperiores corrupti, quisquam ducimus! Sint, ipsa. Voluptates adipisci facilis veniam impedit?

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Totam et rem accusantium dicta voluptatum quam suscipit, molestiae dolor, nihil asperiores corrupti, quisquam ducimus! Sint, ipsa. Voluptates adipisci facilis veniam impedit?

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Totam et rem accusantium dicta voluptatum quam suscipit, molestiae dolor, nihil asperiores corrupti, quisquam ducimus! Sint, ipsa. Voluptates adipisci facilis veniam impedit?

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

i http://127.0.0.1:5500

Les système de boucles

LOL JE SUIS HACKERMAN

LOL JE SUIS HACKERMAN

LOL JE SUIS HACKERMAN

LOL JE SUIS HACKERMAN

LOL JE SUIS HACKERMAN

LOL JE SUIS HACKERMAN

LOL JE SUIS HACKERMAN

LOL JE SUIS HACKERMAN

LOL JE SUIS HACKERMAN

LOL JE SUIS HACKERMAN

<https://github.com/jefff404/cours-js/tree/10-boucle>

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.