

# Javascript et API Fetch

🕒 Date de création	@10 mars 2023 16:09
🏷 Étiquettes	

Toujours accompagner le code d'une doc !!

## ▼ COURS

- Installer Node JS
- Utiliser VSCode
- Mdn web docs :: bible : <https://developer.mozilla.org/fr/>
- **Commentaires** : // ou /\* \*/
- Kamel Case séparation par des majuscules
- DOM : Document Object Model variable d'object
- Pas de maths avec du JS

### ▼ **RegEx** Pal sert a sélectionner un caractère dans une string

placer une chaine en RegEx /[str]/

g = global, match all instances of the pattern in a string, not just one

i = case-insensitive (so, for example, /a/i will match the string "a" or "A".

### ▼ **Manipulation de variables et de données** :

**Déclaration d'une variable** : var ex : var mavariable "bonjour":

Let = variable qui peut etre réaffecté

Const = variable non modifiable

un itérable est une donnée pouvant être parcourue

map La méthode `map()`

crée un nouveau tableau avec les résultats de l'appel d'une fonction fournie sur chaque élément du tableau appelant.

```
const testArray = [1, 3, 5, 654]
testArray.map( (element) => {
  console.log(element)
} )
```

### ▼ **Structure de contrôle** if, else, elif, while...

en JS utiliser un === pour une égalité !== inégalité car == ne prend pas en compte le type de donnée

While (tant que) / do while —> va execute au moins une fois l'opération

Boucle for in -->

```
for( const key in MyObject ){ // sert a parcourir l'object
  Const element = MyObject[key]
  Console.log(key)
}
```

forEch —> action pour chaque element du tableau

### ▼ **La syntaxe de l'objet en JS** : C'est une struct de données complexe

Ex :

```
//structure :

/*let MyObject = {
  Firstname : 'jhon',
  Lastname : 'Doe',
  age : '25',
}
  object !*/
```

Boucle for in -->

```
for( const key in MyObject ){ // sert a parcourir l'object

  Const element = MyObject[key]

  Console.log(key)

}
```

▼ **LA Fonctions en JS** : éviter le ternaire car compliquer à lire.

déclaration de Fonction != appel de fonction

**Fonction normale ou fléchée. Fléchée INDISPENSABLE EN JS**

Ex :

```
//let sum = (a,b) => a+b // return IMPLICITE
```

```
function sum(a, b) { // parametres

  return a+b;

}

//let sum = (a,b) => a+b // return IMPLICITE
```

**Usage des callbacks : la fonction appelle un autre fonction --> programmation asynchrone**

**Création d'une pile d'appel pour exécution des fonctions**

Ex :

```
setTimeout (

  function(){

    console.log("Hello World !");

  }, 5000)
```

**... en JS sert à étaler les valeurs d'un tableau un objet etc.**

ex:

```
function sum(x, y, z) {
  return x + y + z;
}

const numbers = [1, 2, 3];

console.log(sum(...numbers));
// Expected output: 6
```

**utilisation des équivalents avec ===**

```
if (1===true){
  console.log("equals")
}else{
```

```
    console.log("ne")
  }
```

### ▼ JS et le DOM (document object model) :

```
// la balise afin d'insérer du JS dans le HTML
<script> </script>
```

```
//accès to the document object déclaration de doc
const doc = document;

//accès to body element
const doc = document.body

//select element by class name
const myClassElement = doc.getElementsByClassName("myClass");

//select element by tag name
const myTagElement = doc.getElementsByTagName("p");

// select the first element that matches a CSS selector
const firstMatch= doc.querySelector(".myClass");

// select all elements that matches a CSS selector
const allMatches= doc.querySelectorAll(".myClass");
```

### Manipuler les elements du DOM

```
// change the text content of an element
myElement.textContent= "Newtext"

// change the inner html of an element
myElement.innerHTML = "<p> New paragraphe </p>"

// change the style of an element
myElement.style.color = "red"

//remove element from the dom
myElement.remove()
```

### Répondre aux elements du DOM :

```
// Attach a click even listener to an element
myElement.addEventListener("click", function(){
  console.log("Element clicked!");
});
```

## ▼ API FETCH

L'API Fetch fournit une interface pour la récupération de ressources (e.g., à travers le réseau.)

Fetch fournit une définition générique des objets `Request` et `Response` (et d'autres choses impliquées par les requêtes réseau)

### ▼ Les objects promised

L'objet `Promise`

(pour « promesse ») est utilisé pour réaliser des traitements de façon asynchrone. Une promesse représente une valeur qui peut être disponible maintenant, dans le futur voire jamais.

Promise :

```
const promise = new Promise((resolve, reject) =>{
  setTimeout(() => {
    const randomNumber = Math.floor(Math.random() * 10)
    if (randomNumber % 2 === 0) {
      resolve(randomNumber)
    } else {
      reject(new Error('Le nombre est impair.'))
    }
  })
})
```

```

    }, 1000)

  })
  promise
    .then(result => console.log(result))
    .catch(error => console.error(error.message))

  console.log(promise)

```

## ▼ Exercices

### ▼ Manipulation de tableaux

1. Créez un tableau contenant les nombres de 1 à 10. Écrivez un programme qui affiche chaque . élément du tableau à la console.

```

//manipulation de tableau

const chiffres = [1, 2, 3,4,5,6,7,8,9,10];

for (let i = 0; i<chiffres.length; i++){
  console.log(chiffres[i])
}
//correction
const monTab = [1,2,3,4,5]
monTab.forEach((numero) => {
  console.log(numero)
})

```

2. Créez un tableau contenant les jours de la semaine. Écrivez un programme qui affiche le premier jour de la semaine à la console.

```

const Jours = ["Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi", "Dimanche"]
console.log(Jours[0])

const chiffres2 = [2,4,6,8,10];
let j = 0
for (let i = 0; i<chiffres.length; i++){
  j+=chiffres2[i]
  console.log(j)
}

```

3. Créez un tableau contenant les nombres pairs de 2 à 10. Écrivez un programme qui calcule la somme des nombres du tableau et l'affiche à la console.

```

const Fruits = ["Banane", "Orange", "Pomme", "Mange", "litchi"];
Fruits.push("Kiwi");
console.log(Fruits)

```

4. Créez un tableau contenant les noms de cinq fruits. Écrivez un programme qui ajoute un sixième fruit à la fin du tableau et affiche le tableau complet à la console.

```

const Notes = [2,4,6,8,10];
let k = 0
for (let i = 0; i<Notes.length; i++){
  k+=Notes[i]
}
console.log(k/Notes.length)

```

5. Créez un tableau contenant des noms d'oiseaux. Écrivez un programme qui recherche si un nom spécifique d'oiseau est présent dans le tableau.

```
const Oiseaux = ["Colibri", "Mésange", "Merle", "Perroquet", "Etourneau"];
for (let i = 0; i < Oiseaux.length; i++) {
  if (Oiseaux[i] === "Colibri") {
    console.log("Cet oiseau est Bien dans le tableau à la position", i, "du tableau");
  }
}

//correction
const birdFinder = (tab, birdname) => {
  return tab.includes(birdname);
}
```

6. Créez un tableau contenant des noms de pays. Écrivez un programme qui trie le tableau par ordre alphabétique et l'affiche à la console.

```
const Pays = ["France", "USA", "Kosovo", "Gabon", "Suisse", "Belgique", "Angleterre"];

let PaysTri = Pays.sort();
console.log(PaysTri);
```

7. Créez un tableau contenant les noms des mois de l'année. Écrivez un programme qui affiche le troisième élément du tableau.

```
let Tab = [];
let TabMax = 10;
let limite = TabMax + 1;
for (let i = 0; i < TabMax; i++) {
  Tab.push(Math.floor(Math.random() * limite));
}
console.log(Tab);
```

8. Créez un tableau contenant des nombres aléatoires. Écrivez un programme qui calcule la valeur minimale et la valeur maximale du tableau et les affiche à la console.

```
min=Tab[0]

for (let x = 0; x < TabMax; x++) {

  if (min > Tab[x+1]) {
    min=Tab[x+1]
  }
  x+=1
}
console.log("le minimum du tableau est ",min)

max=Tab[0]

for (let x = 0; x < TabMax; x++) {

  if (max < Tab[x+1]) {
    max=Tab[x+1]
  }
  x+=1
}
console.log("le maximum du tableau est ",max)
```

9. Créez un tableau contenant des phrases. Écrivez un programme qui cherche la phrase la plus longue du tableau et l'affiche à la console.

```
const Phrases = ["je mange", "je lis un livre", "elle regarde", "nous nageons"];

let PhraseL = 0;
for (let x = 0; x < Phrases.length; x++) {

  if (Phrases[x].length < Phrases[x+1].length) {
    PhraseL = Phrases[x+1]
  }
  x+=1
}
```

```

}

console.log("la phrase la plus longue est : ", PhraseL)

```

## ▼ Manipulation de fonctions

1. Écrivez une fonction qui prend deux nombres comme arguments et renvoie leur somme.

```
let sum = (a,b) => a+b
```

2. Écrivez une fonction qui prend un tableau de nombres et renvoie le plus grand nombre du tableau.

```

function maxTab(Tab){
    let maxTab = Math.max(...Tab);

    return maxTab

}
tab = [10,20,30,5,4]
console.log(maxTab(tab))

```

3. Écrivez une fonction qui prend une chaîne de caractères et renvoie une nouvelle chaîne de caractères dont toutes les voyelles ont été supprimées.

```

function supprimerVoyelles(chaine) {
    let nouvelleChaine = "";
    for (let i = 0; i < chaine.length; i++) {
        let caractere = chaine.charAt(i).toLowerCase();
        if (caractere !== "a" && caractere !== "e" && caractere !== "i" && caractere !== "o" && caractere !== "u" && caractere !== "y") {
            nouvelleChaine += chaine.charAt(i);
        }
    }
    return nouvelleChaine;
}
phrase = "je mange "
console.log(supprimerVoyelles(phrase))

//correction
function removeVoyels(str){
    return str.replace(/[aeiouy]/gi,'');
}

```

4. Écrivez une fonction qui prend un tableau de chaînes de caractères et renvoie un tableau de chaînes de caractères triés par ordre alphabétique.

```

function trierSTR(carac){
    trierSTR = carac.sort()
    return carac;
}
const Pays = ["France", "USA", "Kosovo", "Gabon", "Suisse", "Belgique", "Angleterre"]

console.log(trierSTR(Pays))

```

5. Écrivez une fonction qui prend un nombre en argument et renvoie une chaîne de caractères représentant ce nombre en mots. Par exemple, si l'entrée est 42, la fonction doit retourner "quarante-deux".

```

function Nombre (num){
    const ones = [' ', 'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
    const tens = [' ', ' ', 'twenty', 'thirty', 'fourty', 'fifty', 'sixty', 'seventy', 'eighty', 'ninety']
    const teens = ['ten', 'eleven', 'twelve', 'thirteen', 'fourteen', 'fifteen', 'sixteen', 'seventeen', 'eighteen', 'nineteen']
    if (num<10){
        return ones[num];
    }
    else if (num <20){
        return teens [num-10];
    }
    else {
        return tens[Math.floor(num/10)] + '-' + ones[num%10];
    }
}

```

```
}  
console.log(Nombre(42))
```

6. Écrivez une fonction qui prend un tableau d'objets et un nom de propriété et renvoie un tableau des valeurs de cette propriété dans chaque objet.

```
const someObject = {  
  firstname : 'Théophile',  
  lastname: 'Faugeras',  
  age: 19,  
  hasDriverLicence: false,  
  placeOfBirth: 'Nogent-Sur-Marne',  
}  
  
for (const key in someObject) {  
  const element = someObject[key]  
  console.log(element)  
}
```

7. Écrivez une fonction qui prend un tableau de nombres et renvoie un tableau de nombres triés par ordre décroissant.

```
tab2= [10,5,8,67,4,39]  
function sortDescending (tab2){  
  return tab2.sort((a, b)=> b-a)  
}  
console.log(sortDescending(tab2))
```

8. Écrivez une fonction qui prend une chaîne de caractères et renvoie une nouvelle chaîne avec toutes les voyelles en majuscules.
9. Écrivez une fonction qui prend une chaîne de caractères et renvoie le nombre de voyelles dans cette chaîne.
10. Écrivez une fonction qui prend une chaîne de caractères et renvoie une nouvelle chaîne avec toutes les consonnes en majuscules.