

Node JS

Développement d'applications Web

Intervenant : Jean-Frédéric VINCENT





Module 2

JavaScript next Génération

Définition de variables : let , const et var

Les fonctions fléchées

L'opérateur ... (3 petits points) Spread et Rest

Les fonctions asynchrones

Les promesses

Les littéraux de gabarit

Définition de variables : let & const

- **let** est le nouveau var
- **const**

une variable que l'on instancie une fois
ne reçoit pas de nouvelle valeur

- **var**

var n'est plus utilisé

```
const pi = 3.14;  
  
const personne = {prenom:'Brad',nom:'PITT'};  
  
const tableau=['pomme','poire','cerise'];  
  
for (let i=0;i<4;i++){  
    console.log(i);  
}  
console.log(i);
```

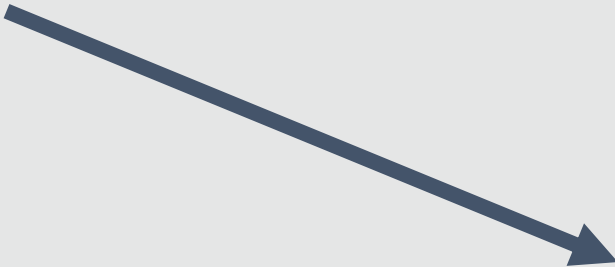
Les fonctions fléchées

- **Arrow functions**

Utilisation de const

abstraction :

- des parenthèses
- du return



```
function afficheMonNom(nom) {  
    console.log(nom);  
}  
  
const afficheNom2 = (nom) => {  
    console.log(nom)  
}  
  
const multi = (num) => {  
    return num*2;  
}  
  
const multi2 = num => num*2;
```

L'opérateur Spread ...

Le ...

Permet de récupérer toutes les valeurs

- d'un autre tableau
- d'un autre objet

```
const personne = {prenom: 'Brad', nom: 'PITT'};
const client = {...personne, age: 18};
console.log(client);

const tableau = [1, 2, 3];
const tableau2 = [...tableau, 4, 5];
console.log(tableau2);
```

```
[object Object] {
  age: 18,
  nom: "PITT",
  prenom: "Brad"
}

[1, 2, 3, 4, 5]
```

L'opérateur Rest ...

Le ...

Permet de récupérer des arguments

```
const trouveKiwi=(...args)=>{  
  return args.filter(el => el === 'kiwi');  
}  
console.log(trouveKiwi('pomme', 'kiwi'));
```

```
['kiwi']
```

Les fonctions asynchrones

- Une fonction synchrone **async** peut contenir une expression **await** qui interrompt l'exécution de la fonction asynchrone et attend la résolution de la promesse passée Promise. La fonction asynchrone reprend ensuite puis renvoie la valeur de résolution.
- Le mot-clé **await** est uniquement valide au sein des fonctions **asynchrones**.

```
const go async ()=>{  
  const response = await fetch("https://swapi.dev/api/people/1");  
  // on récupère la partie JSON de la response  
  const data = await response.json();  
  console.log(data);  
}  
go();
```

Dès que ma fonction contient **await**, elle doit être prefixer par **async**

JavaScript next Génération

Les promesses

- Une Promise est un objet qui va prendre un certain temps pour se résoudre ou rejeter.

```
const repondApresDeuxSeconde= () => {  
  return new Promise((resolve) => {  
    setTimeout(() => {  
      resolve('ok');  
    }, 2000);  
  });  
}  
  
const go= async()=> {  
  const result = await repondApresDeuxSeconde();  
  console.log(result); // affiche : ok  
}  
  
go();
```

Les littéraux de gabarits

- Les littéraux de gabarits **Template literals** sont une alternative intéressante à la concaténation
 - Le "back quote" : ALT GR + 7 (2 fois)

```
const age =18;
console.log (`Votre age ${age}`);
// avec un if ternaire
console.log (`Votre age ${age} statut :${age >= 18? 'ok':'interdit'}`);
```

JavaScript next Génération

Démonstration