

JavaScript – Programmation avancée

Intervenant : Rémy PIERRE





Module 8

Les Objets – La Programmation Orientée Objets

Création d'objets "on the fly"

Parcourir un objet

Créer une classe

Les tableaux d'objets

08 Les Objets – La Programmation Orientée Objets

Création d'objets

- **On the fly**, A la volé (sans classe) sur une ligne

```
const personne = {prenom: 'Brad', nom: 'PITT'}
```

- Sur plusieurs lignes :

```
const personne = {};  
personne.prenom = 'Brad';  
personne.nom = 'Brad';
```

- Au Format JSON (pas très pratique!)

```
const personne = {"prenom": "Brad", "nom": "PITT"};
```

Parcourir un objet

- Souvent il est nécessaire de **parcourir** un objet quand on reçoit un JSON
- Avec **for** + **in**

```
for ( let attribut of personne){  
    console.log( attribut); // prenom ..nom  
    // un objet c'est aussi un tableau  
    console.log(personne[attribut]); // Brad ..PITT  
}
```

Créer une classe

- Avec les mots clef **class** et **objet**
- Il est possible de créer une classe

```
class Personne {  
    constructor(prenom, nom) {  
        this.prenom = prenom;  
        this.nom = nom;  
    }  
}  
  
const p1 = new Personne('Brad', 'PITT');  
// on peut aussi  
p1.age = 18;
```


Les tableaux d'objets

- La plus part des données dans les applications sont stockées sous forme de tableau d'objets

```
const appareils =[  
  {name:'TV', status:'allumé'},  
  {name:'Xbox', status:'éteint'},  
  {name:'Machine à café', status:'allumé'}  
];
```

Les tableaux d'objets

- **Déclarer** un tableau d'objet
- La plus part des données dans les applications sont stockées sous forme de tableau d'objets

```
const appareils =[  
  {name:'TV', status:'allumé'},  
  {name:'Xbox', status:'éteint'},  
  {name:'Machine à café', status:'allumé'}  
];
```

```
const appareils =[];
```


Afficher un tableau d'objets

- Pour parcourir la boucle for each (**for** + **of**) est adaptée

```
for (let a of appareils){  
    console.log(a.name + ' ' + a.status)  
}
```

Afficher un tableau d'objets

- La boucle **map** est encore mieux !

```
appareils.map( a => { console.log(a.name + ' ' + a.status)} );
```

- Avec l'indice

```
appareils.map( (a, indice) => {  
    console.log(indice);  
    console.log(a.name + ' ' + a.status);  
});
```

Modifier un objet dans un tableau

- Modifier un objet

```
appareils[2].status = 'éteint'; // Machine à café
```

- Modifier plusieurs objets

```
for (let a of appareils){  
    a.status = 'éteint';  
}
```

```
appareils.map(a => a.status='éteint');
```

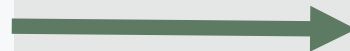
Le if ternaire

- Le if ternaire est très utilisé avec React

```
const age = 18;  
if (age >= 18){  
  console.log('ok');  
}else{  
  console.log('interdit');  
}
```



```
const age = 18;  
// VIVE le if ternaire !!!  
age >= 18  
  ? console.log('ok')  
  : console.log('interdit')
```



- if ternaire avec seulement si la condition est VRAI

```
age >= 18 && console.log('ENTREZ')
```

- Il est possible de l'écrire en une ligne

```
age >= 18 ? console.log('ok') : console.log('interdit')
```