Les propriétés des objets

Déclarer des propriétés sur un objet

Les objets `JavaScript` peuvent avoir une ou plusieurs propriétés.

Vous pouvez voir une propriété comme une variable attachée à l'objet.

Il est possible de déclarer des propriétés avec **le point `.`** ou **en utilisant des crochets**.

En utilisant le point `.`

Nous allons déclarer un objet littéral et définir deux propriétés sur cet objet :

```
const monObjet = {};
monObjet.propriete1 = 42;
monObjet.propriete2 = "Koala";

console.log(monObjet); // {propriete1: 42, propriete2: "Koala"}
```

En utilisant les crochets `[ ]`

Nous allons maintenant déclarer un objet littéral et définir trois propriétés sur cet objet mais en utilisant les crochets :

```
const monObjet = {};
monObjet["prop1"] = 42;
monObjet['prop2'] = "Koala";
monObjet['Une autre propriété'] = 14664441n;

console.log(monObjet); // {prop1: 42, prop2: "Koala", Une autre propriété: 14664441n}
```

Notez que nous pouvons utiliser des guillemets simples ou doubles.

Il existe deux cas où **la définition de propriété avec les crochets est obligatoire : en utilisant le contenu d'une variable comme clé et lorsque la clé contient des espaces**.

Prenons un exemple en utilisant une variable comme nom de propriété :

```
let unePropVariable = "Koala";
const monObjet = {};
monObjet[unePropVariable] = 42;
console.log(monObjet); // {Koala: 42}
```

Dans ce cas, nous parlons de **nom de propriété calculée** car elle est définie dynamiquement lors de l'exécution.

Directement lors de la déclaration

Il est également possible de définir les propriétés initiales d'un objet lors de sa déclaration comme nous l'avions fait avec :

```
const employe1 = {
  age: 42,
  salaire: 32000,
  fonction: 'acheteur'
};
```

Accèder aux propriétés d'un objet

De la même manière, vous pouvez accéder aux propriétés en utilisant **un point ou les crochets**.

En utilisant un point :

```
employe1.age; // 42
```

En utilisant les crochets, ce qui est obligatoire pour les propriétés calculées ou sur plusieurs mots :

```
employe1['salaire']; // 32000
```

Modifier un objet

Vous pouvez modifier sans problème les propriétés d'un objet :

```
const monObjet = {};
monObjet["prop1"] = 42;
monObjet.prop1 = 43;

console.log(monObjet); // {prop1: 43}
```

Il faut cependant se rappeler que **les objets sont passés par référence**. Il faut donc faire très attention de ne pas modifier involontairement un objet référencé dans plusieurs variables :

```
const monObjet = {a: 42};
const monObjet2 = monObjet;
```

```
monObjet2.a = 0;

console.log(monObjet); // {a: 0}
console.log(monObjet2); // {a: 0}
```

Les méthodes

Les méthodes d'un objet sont des propriétés dont la valeur sont des fonctions. Nous allons les voir que très brièvement car nous verrons les fonctions plus tard.

```
const monObjet = {
  direBonjour: function() {
    console.log("Salut !");
  }
};
monObjet.direBonjour(); // Salut !
```

Il existe également une notation raccourcie pour la déclaration de méthodes :

```
const monObjet = {
  direBonjour() {
    console.log("Salut !");
  }
};
monObjet.direBonjour(); // Salut !
```