L'objet littéral

En JavaScript, un objet est une entité à part entière qui possède des propriétés et un type. Si on effectue cette comparaison avec une tasse par exemple, on pourra dire qu'une tasse est un objet avec des propriétés. Ces propriétés pourront être la couleur, la forme, le poids, le matériau qui la constitue, etc. De la même façon, un objet JavaScript possède des propriétés, chacune définissant une caractéristique.

Un objet est un ensemble de propriétés et une propriété est une **association** entre un **nom** (aussi appelé clé) et une **valeur**. La valeur d'une propriété peut être une fonction, auquel cas la propriété peut être appelée « méthode ».

```
let pierre = {
    nom : ['Pierre', 'Giraud'],
    age : 29,
    mail : 'pierre.giraud@afpa.fr',
    bonjour : function(){
        alert('Bonjour, je suis ' + this.nom[0] +
        'j\'ai ' + this.age + ' ans');
    }

};
```

1 Nom, age et email sont des **propriétés** de l'objet « pierre ».

2 Bonjour est une **méthode** de l'objet « pierre ».

Remarque ☺ :

On parle ici d'objet « littéral » car nous avons défini chacune de ses propriétés et de ses méthodes lors de la création, c'est-à-dire littéralement.

Création d'un objet :

Pour créer un objet littéral, on utilise une syntaxe utilisant une paire d'accolades { … } qui indique au JavaScript que nous créons un objet.

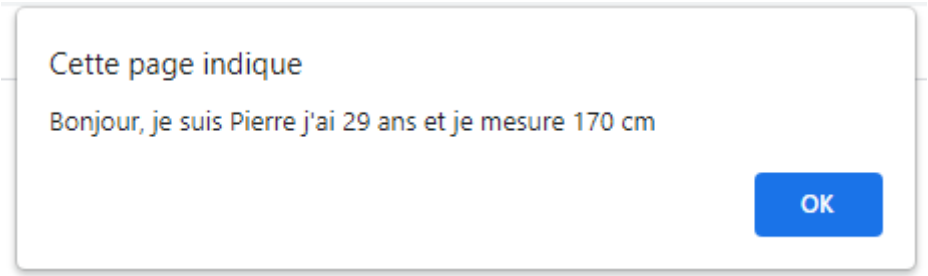
Les accesseurs de propriétés :

Permettent de fournir un accès aux propriétés d'un objet en utilisant une notation avec un **point** ou une notation avec des **crochets.**

```
alert(pierre.age);
alert(pierre["age"]);
```

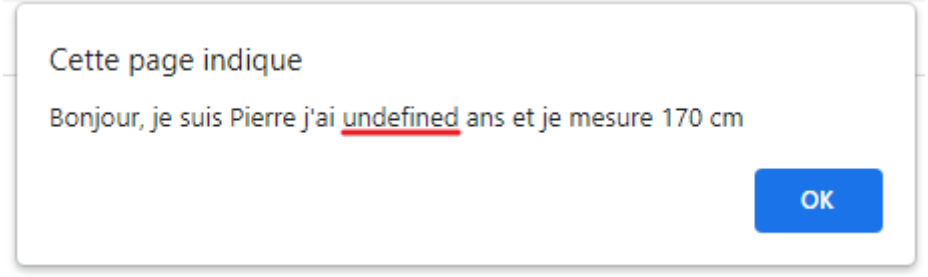
Ajouter une propriété :

```
12  pierre.taille = 170;
13  //or
14  pierre["taille"] = 170;
15
16  pierre.pres = function(){
17      alert('Bonjour, je suis ' + this.nom[0] +
18          ' j\'ai ' + this.age + ' ans et je mesure '
19          + this.taille + ' cm');}
20
21  pierre.pres();
```



Supprimer une propriété :

```
18  delete pierre.age;
19  // or
20  delete pierre['age'];
21  // or
22  let del = "age";
23  delete pierre[del];
24
25  pierre.pres = function(){
26      alert('Bonjour, je suis ' + this.nom[0] +
27          ' j\'ai ' + this.age + ' ans et je mesure '
28          + this.taille + ' cm');}
29
30  pierre.pres();
```



L'opérateur in :

Renvoie **true** si une propriété donnée appartient à l'objet donné .

```
let objet = {
    a : 1,
    b : 2,
    c : 3,
};
console.log('b' in objet); //true

delete objet.b; // supprimer la prop b

if('b' in objet === false){
    objet.d = 4; // ajouter la prop d
}
```

For ... in :

L'instruction for-in permet d'itérer sur les propriétés énumérables d'un objet qui ne sont pas des symboles. Pour chaque propriété obtenue, on exécute une instruction (ou plusieurs grâce à un bloc d'instructions).

```
for (let property in objet){
    console.log(property + "=" + objet[property] );

}
```