Tester l'existence et la valeur d'une propriété

Vérifier qu'une propriété existe sur un objet

Il existe plusieurs moyens de vérifier si une propriété existe sur un objet

Utiliser la comparaison à `undefined`

Vous pouvez utiliser l'opérateur de comparaison stricte pour vérifier si la valeur associée à une clé n'est pas `undefined`.

Si c'est le cas, la propriété n'existe pas :

```
const monObj = {};
monObj.a === undefined; // true
```

Vous pouvez le retrouver dans des conditions :

```
if (monObj.a !== undefined) {
…
}
```

Ici on vérifie que la propriété `a` a été définie. Elle peut cependant contenir une valeur `falsy` : par exemple `null`, `false` ou une chaîne de caractère vide.

Nous verrons que nous utiliserons une autre condition si nous voulons simplement savoir si une propriété n'est pas définie ou qu'elle n'a pas de valeur.

Utiliser l'opérateur `in`

Il existe également l'opérateur `in` qui renvoie un booléen suivant qu'une propriété donnée appartient à l'objet donné.

```
const monObj = {a: 1};
"a" in monObj; // true
"b" in monObj; // false
```

Attention ! Si vous définissez la valeur d'une propriété à `undefined`, `in` retournera `true` :

```
const monObj = {a: undefined};
"a" in monObj; // true
```

Comme indiqué dans les chapitres précédents, il est fortement recommandé de ne jamais assigner la valeur `undefined` mais de toujours assigner `null` pour indiquer l'absence de valeur.

Utiliser la méthode `hasOwnProperty`

La méthode `hasOwnProperty()` retourne un booléen indiquant si l'objet a la propriété passée en paramètre et qu'elle n'est pas sur son prototype. Nous verrons plus tard ce qu'est un prototype.

```
const monObj = {a: 1};
monObj.hasOwnProperty("a"); // true
```

Vérifier qu'une propriété a une valeur non `falsy`

Pour vérifier qu'une propriété a une valeur autre que `falsy` vous retrouverez très souvent cela :

```
if (monObj.a) {
  // La propriété a de l'objet existe et a une valeur qui n'est pas falsy.
}
```

Pour rappel les valeurs `falsy` en `JavaScript` sont `false`, `null`, `undefined`, `0`, `NaN`, `''`, `""` et ``` `` ```.

Pour rappel également, une valeur est dite `falsy` car elle est convertie automatiquement en `false`, notamment dans les `if … else`.