



JavaScript, les objets

Alvin Berthelot

Version 1.0.0



Ce(tte) œuvre est mise à disposition selon les termes de la Licence Creative Commons.

Attribution - Partage dans les Mêmes Conditions 3.0 non transposé.



La licence, ses explications ainsi que les moyens de contribution et réappropriation sont détaillés à la fin.

Les objets littéraires

Définition d'un objet littéral

En JavaScript, un objet littéral est une façon concise de définir un objet.

Rappel

Les objets en JavaScript sont des ensembles de propriétés, elles-mêmes définies comme des paires clé/valeur où la clé est une chaîne de caractère et la valeur, toute valeur possible en JavaScript.

```
var cat = {  
  "name": 'Grumpy',  
  "age": 3,  
  "male": true,  
  "saySomething": function() {  
    console.log('Not happy');  
  }  
};
```

Accès aux propriétés (en lecture)

Il existe 2 façons pour accéder à la propriété d'un objet.

```
var cat = {  
  "name": 'Grumpy',  
  "age": 3  
};
```

Avec la notation de type "point" et avec la notation de type "tableau".

```
console.log('My cat ' + cat.name + ' is ' + cat['age'] + ' years old.');
```



La notation "point" est plus lisible mais statique, alors que la notation "tableau" est dynamique.

Accès aux propriétés (en écriture)

Les notations de type "point" et "tableau" permettent également d'assigner des valeurs.

```
cat.name = 'Garfield';  
cat['age'] = 6;
```



L'accès à une propriété inexistante retournera **undefined**.

Mais si la propriété est valorisée alors elle est automatiquement ajoutée à l'objet.

```
console.log(cat.isDomestic); // undefined  
cat.isDomestic = true;  
console.log(cat.isDomestic); // true
```

Référence Vs valeur

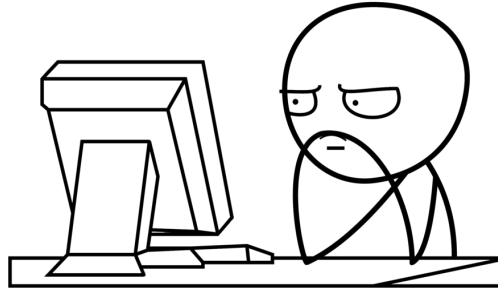
Un objet est toujours retourné par référence et non pas copié/cloné.

Cela signifie qu'une modification d'un état d'un objet dans une fonction sera aussi effective en dehors de la fonction (effets de bord).

```
var sayMeow = function(cat) {  
  ...  
  cat.age ++;  
};
```



Cette façon d'assigner une nouvelle valeur est très dangereuse car elle génère des effets de bord.



Exercices

On va s'appuyer sur la plateforme [NodeSchool.io](https://nodeschool.io), pour cela vous devez [installer le TP "javascripting"](#).

⇒ OBJECTS

⇒ OBJECT PROPERTIES

La notion de prototype

Héritage prototypal

En JavaScript, un objet peut être "dérivé" d'un autre avec la propriété `__proto__`. Ce dernier est alors considéré comme son prototype.

```
// prototype
var animal = {
  "name": 'Grumpy',
  "age": 3,
  "male": true
};

var cat = {};
cat.__proto__ = animal;

console.log('My cat ' + cat.name + ' is ' + cat.age + ' years old.');
```

Chaîne de dérivation

Le lien entre l'objet dérivé et son prototype sert pour l'accès aux propriétés.

Si une propriété demandée n'existe pas dans l'objet dérivé, elle est cherchée dans le prototype.

```
// prototype
var animal = {
  "name": 'Grumpy',
  "age": 3,
  "male": true
};

var cat = {
  saySomething: function() {
    return 'Not happy';
  }
};
cat.__proto__ = animal;

console.log('My cat ' + cat.name + ' says ' + cat.saySomething());
```

Héritage dynamique

La relation est dynamique (mais unilatérale), si à posteriori le prototype reçoit une nouvelle propriété, le dérivé l'expose aussi.

```
// prototype
var animal = {
  "name": 'Grumpy'
};

var cat = {};
cat.__proto__ = animal;

animal.age = 3;

console.log('My cat ' + cat.name + ' is ' + cat.age + ' years old.');
```

Le premier prototype

Par défaut, les objets ont comme prototype un objet pré-défini par JavaScript `Object.prototype`, c'est également vrai pour les objets littéraux.

```
var cat = {  
  "name": 'Grumpy'  
};  
  
cat.hasOwnProperty('name');
```

Lister les propriétés

Un objet étant une collection de propriétés, elles peuvent être parcourues via l'instruction `for in`.

```
// prototype
var animal = {
  "name": 'Grumpy',
  "age": 3,
  "male": true
};

var cat = {
  saySomething: function() {
    return 'Not happy';
  }
};
cat.__proto__ = animal;

var property;
for (property in cat) {
  console.log(property + ': ' + cat[property]);
}
```

Suppression d'une propriété

Une propriété peut être supprimée avec l'opérateur **delete**.

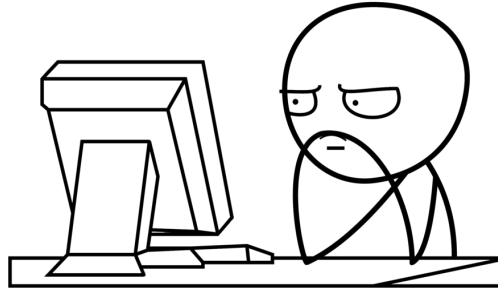
La suppression n'affecte pas le prototype.

```
// prototype
var animal = {
  "name": 'Grumpy',
  "age": 3,
  "male": true
};

var cat = {
  age: 6
};
cat.__proto__ = animal;

console.log(cat.age); // 6

delete cat.age;
console.log(cat.age); // 3
```



Exercices

On va s'appuyer sur la plateforme [NodeSchool.io](https://nodeschool.io), pour cela vous devez [installer le TP "Planet Proto"](#).

- ⇒ Simple objects
- ⇒ Proto
- ⇒ Dynamic lookups
- ⇒ Property assignments
- ⇒ Arrays and Objects

Licence



CC BY-SA 3.0

Ce(tte) œuvre est mise à disposition selon les termes de la Licence Creative Commons. Attribution - Partage dans les Mêmes Conditions 3.0 non transposé.

Copyright © 2017 [Alvin Berthelot](#).

Pour toutes questions, réclamations ou remarques, merci d'envoyer un message à alvin.berthelot@webyousoon.com.

Explications licence CC BY-SA 3.0

Cette licence permet aux autres de remixer, arranger, et adapter votre œuvre, même à des fins commerciales, tant qu'on vous accorde le mérite en citant votre nom et qu'on diffuse les nouvelles créations selon des conditions identiques.

Cette licence est souvent comparée aux licences de logiciels libres, "open source" ou "copyleft".

Toutes les nouvelles œuvres basées sur les vôtres auront la même licence, et toute œuvre dérivée pourra être utilisée même à des fins commerciales.

C'est la licence utilisée par Wikipédia ; elle est recommandée pour des œuvres qui pourraient bénéficier de l'incorporation de contenu depuis Wikipédia et d'autres projets sous licence similaire.

Contribution et réappropriation

Ce fichier PDF est généré avec [Asciidoctor](#) à partir d'un dépôt Git se trouvant sous GitHub.

<https://github.com/alvinberthelot/slides-js>

Cela signifie que vous n'avez pas besoin de vous battre avec un fichier binaire (le PDF) pour **contribuer**, **vous réapproprier le contenu** ou **modifier le thème** de présentation.



Contribution

Vous voulez **contribuer au contenu** car :

- Il y a une erreur (ça arrive à tout le monde), de typographie, de compréhension, ou tout autre chose.
- Vous souhaitez apporter une précision.

Il vous suffit de [contribuer au projet via Git](#) par le moyen d'une "pull request" sur le [dépôt Git](#).



Réappropriation



N'oubliez pas les conditions de la licence.

Vous voulez vous **réapproprier le contenu** car :

- Vous souhaitez donner un style différent.
- Vous souhaitez enlever/ajouter/modifier des sections dans votre contexte.

Il vous suffit de "forker" le [dépôt Git](#) et d'y apporter vos propres modifications, puis de générer par vous même le nouveau PDF.

