

Objets et Javascript

Jean-Marc Lecarpentier

Adapté du cours de Frédéric Jurie

Notion d'objets

- peuvent être comparés aux objets de la vie de tous les jours.
- entité indépendante, ayant des propriétés et un type.
- ex : une voiture possède des propriétés, couleur, sa forme, son matériau
- un objet JavaScript disposera de propriétés définissant ses caractéristiques.

Objets et propriétés

- propriété : peut être vue comme une variable associée à l'objet.
- Les propriétés d'un objet sont des variables classiques à l'exception du lien qui les rattache à l'objet.
- permettent de définir les caractéristiques d'un objet
- possible d'accéder aux propriétés d'un objet en utilisant le nom de l'objet, un point et le nom de la propriété.

```
nomDeMonObjet.nomDeLaPropriete
```

Exemple

```
let maVoiture = {  
  "fabricant": "Ford",  
  "modele": "Mustang";  
  "annee": 1969  
}
```

Exemple

```
maVoiture["fabricant"] = "Ford";  
maVoiture["modele"] = "Mustang";  
maVoiture["annee"] = 1969;
```

Objets et méthodes

- Une méthode est une fonction propre à un objet
- et qui « agit » sur l'objet lui-même

- Exemple :

```
let a = "une chaine de caractères";
```

```
// propriété length
```

```
console.log(a.length);
```

```
// méthode substr
```

```
console.log(a.substr(14, 10));
```

Les objets natifs de JavaScript

- Tout est un objet en Javascript
- Programmation Objet : créer ses propres objets (Python, Java en L2)
- Objet existants du langage : objets natifs
- Manipulation des données plus facile

Objet String

- Création :
 - `let string1 = new String("Hello");`
 - `let string1bis = "Hello";`
 - `let string2 = new String(123);`
 - `let string3 = new String(123.456);`
- Propriété length :
 - `maChaine.length`

Objet String (suite)

- Recherche de sous chaines : méthode
 - `indexOf()`, `lastIndexOf()`; retourne la position de la première/ dernière sous-chaîne. 1er arg : chaîne à recherche. 2nd (optionnel) : caractère de début de la recherche.
- copie d'une partie de la chaîne :
 - `myString.substr(0, 4);`
- sélection d'un caractère :
 - `let theLastChar = myString.charAt(myString.length - 1);`
- changement de case :
 - `myString.toLowerCase()`
 - `myString.toUpperCase()`

Objet Array

- déclaration :

```
let tab = new Array( 'item1', 'item2', 'item3' );  
// ou bien  
let tab = [ 'item1', 'item2', 'item3' ];
```

- longueur : `tab.length`
- jonction de tableaux

```
let names = new Array( "Paul", "Jeremy", "Nick" );  
let ages = new Array( 31, 30, 31 );  
let concatArray = names.concat( ages );
```

- copie d'une partie de tableau :

```
let names = new  
Array( "Paul", "Sarah", "Jeremy", "Adam", "Bob" );  
let slicedArray = names.slice( 1, 3 );
```

Objet Array (suite)

- conversion en chaine de caractères :

```
let myShopping = new  
Array("Eggs", "Milk", "Potatoes", "Cereal", "Banana");  
let myShoppingList = myShopping.join("<br />");
```

- Tri :

```
let names = new Array("Paul", "Sarah", "Jeremy", "Adam", "Bob");  
names.sort();
```

- inversion de l'ordre de la liste :

```
let myShopping = ["Eggs", "Milk", "Potatoes", "Cereal", "Banana"];  
myShopping.reverse();
```

Objet Math

- `Math.abs(myNumber)`
- `Math.max()`
`Math.min()` // (deux OU PLUS arguments)
- `Math.ceil()`
`Math.floor()`
`Math.round()`
- `Math.random()`
- `Math.pow(10,decimalPlaces)`

Saisie de données et nombres

- Saisie avec `prompt()` ou input de formulaires
⇒ données de type `String`
- Fonction `parseInt()` : essayer d'obtenir un entier à partir d'une chaîne
- Exemples : // renvoient tous 15
`parseInt('15 * 3', 10);`
`parseInt('15e2', 10);`
`parseInt('15px', 10);`

Not a Number

- NaN : Not a Number
- Valeur pour représenter un nombre qui n'est pas un nombre
- Exemple : `parseInt('abc')` renvoie NaN
- Fonction `isNaN()` pour tester si on n'a pas un nombre
- Ex. `if (isNaN(parseInt(valeur)))` pour tester si on a réussi à extraire un entier de la variable `valeur`

Objet Date

- création
- `let theDate1 = new Date();`
`let theDate2 = new Date(949278000000);`
`let theDate3 = new Date("31 January 2010");`
`let theDate4 = new Date(2010,0,31,15,35,20,20);`
- `getDate()`, `getDay()`, `getMonth()`, `getFullYear()`
- `getTime()` : timestamp Unix en millisecondes (nombre de ms depuis le 1/1/1970)
- `toString()`
- `setDate()`, `setMonth()`, `setFullYear()`

Objet window

- désigne l'onglet de la page dans lequel s'exécute le JS
- objet global : c'est l'objet utilisé par défaut si il est omis
Exemple : `alert('toto')` est un raccourci de `window.alert('toto')`
- Méthodes boites de dialogue : `alert()`, `prompt()`, `confirm()`
- Temporisation : `setTimeout()`, `setInterval()`, `clearTimeout()`, `clearInterval()`
- Méthodes manipulation de l'onglet : `scrollBy()`, `scrollTo()`
- Méthode `open()` pour ouvrir un nouvel onglet
- Méthodes `resizeBy()`, `resizeTo()`, `moveBy()`, `moveTo()`

Objet window (suite)

- contient lui même 5 sous-objets :
- location : contient l'URL active
- history : contient l'historique des URL visitées
- document : contient le contenu de la page web
- navigator : contient des propriétés sur le navigateur
- screen : contient des propriétés sur l'écran

Objet window.location

- Il contient des informations sur l'URL en cours et possède en particulier:
- les propriétés : href, hostname, port, protocol
- la méthode replace() qui permet de modifier l'URL actuellement visitée

Objet window.history

- Il contient l'historique de la fenêtre (c'est-à-dire l'ensemble des URLs visitées).
- Il possède les propriétés et méthodes suivantes (entre autres) :
- la propriété `length` qui représente la taille de l'historique (nombre d'URL visitées).
- la méthode `back()` va en arrière d'une page et `forward()` va en avant d'une page
- `go()` permet un déplacement relatif (p. ex. `go(-1)` revient en arrière d'une page)

En bref

- Toute donnée est un objet
- Possède propriétés et méthodes
- Méthode : action sur l'objet
- Objets natifs : manipulation des données de types String, Number, Date, Array
- Objet window : le contexte d'exécution du JS
- Penser à chercher dans les documentations pour voir si une méthode existe déjà ou non