Les objets et JSON

En JavaScript, on utilise beaucoup d'objets prédéfinis, comme les objets du DOM.

Mais, on peut aussi créer de nouveaux objets.

Il n'y a pas les notions telles que les classes, l'encapsulation et le polymorphisme, qu'on retrouve habituellement dans les langages orientés objet (Java, C#, C++,…). Cependant, on peut définir des fonctions « constructeur » pour créer nos objets.

Par exemple :

Avec la fonction suivante :

```
function etudiant(n, nom, age, moyenne)
{
this.numero = n;
this.nom = nom;
this.age = age;
this.moyenne = moyenne;
}
```

On peut créer des objets *etudiant* :

```
var Etud1 = new etudiant(345,"Pierre",25,89.6);
```

Tous les attributs sont publics et on peut y accéder avec l'opérateur *point* (.) :

```
alert("Bonjour " + Etud1.nom + " votre moyenne est : " +
      Etud1.moyenne);
Etud1.moyenne = Etud1.moyenne - 10 ;
alert("Vous avez été penalisés de 10pts. Votre moyenne est : "
      + Etud1.moyenne);
```

On peut aussi accéder aux attributs d'un objet à l'aide des crochets :

```
Etud1['moyenne']
```

Est équivalent à :

```
Etud1.moyenne
```

On peut ajouter des fonctions membres à nos objets *etudiant* :

```
function etudiant(numero, nom, age, moyenne)
{
this.numero = numero;
this.nom = nom;
this.age = age;
this.moyenne = moyenne;
this.ajouterUnBonus = function(bonus){
       this.moyenne += bonus;
       }
}
```

Qu'on peut utiliser comme ceci :

```
Etud1.ajouterUnBonus(5);
```

La notation objet

JSON (JavaScript Object Notation) est un format de stockage et d'échange de données. Défini initialement dans JavaScript, il est indépendant de tout langage et est entrain de concurrencer XML (eXensible Markup Language). Dans beaucoup de situations, JSON se présente comme une alternative à XML avec l'avantage d'être plus « léger ». En particulier, on peut très facilement convertir une chaîne de caractères JSON en objet JavaScript.

Voici un exemple d'un objet JavaScript :

```
var Etud1 = {numero : 345, nom : "Pierre", age : 25, moyenne :
             89.6 };
```

Pour ajouter une function membre :

```
var Etud1 = {
  numero : 345,
  nom : "Pierre",
  age : 25,
  moyenne : 89.6,
  adresse : {numRue : 1260, nomRue : "av. Bois-de-boulogne"},
  ajouterUnBonus: function(b) {
                              this.moyenne +=b;
                              }
};
```

On peut aussi créer des tableaux suivant cette notation :

```
var notes = [89.2,69,48,75];
```

Et, bien évidemment :

- on peut créer des tableaux d'objets;
- un objet peut avoir un attribut qui est lui-même un objet ou un tableau.

Les chaînes JSON :

Voici un exemple d'une chaîne de caractères JSON décrivant un objet « compte bancaire » :

```
{"numero":"B22","titulaire":"Annie","solde":350}
```

Et voici une chaîne JSON décrivant un tableau d'objets « cours »:

```
[{
 "num": "420-343-RO",
 "nom": "Mise en oeuvre",
 "duree": 45
}, {
 "num": "420-275-RO",
```

```
 "nom": "BD 1",
 "duree": 75
}, {
 "num": "420-035-RO",
 "nom": "Syst. 1",
 "duree": 75
}, {
 "num": "9956",
 "nom": "Test des espaces",
 "duree": 44
}, {
 "num": "INF-015",
 "nom": "Ajax et jQuery",
 "duree": 30
}]
```

On peut convertir une chaîne JSON en objet/tableau JavaScript. Une solution consiste à utiliser la fonction ***eval()***. Par exemple, si *cpt1* est une variable qui contient une chaîne JSON décrivant un objet compte bancaire, alors on peut récupérer un objet JavaScript représentant cet objet :

```
var objCpt1 = eval("("+cpt1+")");
```

De même, on peut récupérer un tableau d'objets JavaScript à partir d'une chaîne JSON contenue dans une variable *tabCours* comme ceci :

```
tabObjCours = eval("("+tabCours+")");
```

On peut aussi utiliser la fonction ***JSON.parse()***.

Pour plus d'informations sur JSON : http://www.w3schools.com/json/default.asp

Validateur JSON : http://jsonlint.com/