

Object Javascript

C'est LE type fondamental de javascript :

- types de base: number, string, booléen, null, undefined
- les objets natifs : Date, Array, Function, Regexp, ...
- les objets définis par l'environnement (hôtes): exemple document

Une valeur composées qui regroupe plusieurs valeurs (primitive ou autres objets)

Une collection non ordonnée de propriétés

- chacune a un nom et une valeur
donc map: chaîne -> valeur, hashtable, dictionnaire, tableau associatif ...

Creation d'objets littéraux

```
1  var vide = {}
2
3  var point = {x:0, y:0};
4
5  var point2 = {x:point.x, y:point.y+1};
6
7
8  var livre = {
9      "titre" : "javascript",
10     "sous titre" : "facile !",
11     auteur : {
12         nom : Dupond,
13         prenom : "D" //pas de , après le dernier,
14     }
15 };
16
17 var point3 = {
18     x:0,
19     y:0,
20     affiche: function() {alert (" "+this.x+" , "+this.y+" ")}
21 };
```

Creation avec New

- Pour les objets natifs:

```

1  var a= new Object();
2  var b= new Array();
3  ...
4

```

- Le mot clef `new` est suivi d'une invocation de fonction.
- Cette fonction est appelée **constructeur**
- Pour les objets définis par l'utilisateur

```

1  function Point(x, y) {
2      this.x = x;
3      this.y = y;
4  }
5
6  p1 =new Point(2,3);
7  p2 =new Point(4,3);
8
9
10

```

Accès aux propriétés de l' objet

```

1  point3.x; //x
2
3  point3["x"]; //autre écriture
4
5  //toutes les propriétés:
6  for (var i in point3){
7      alert("nom :"+i+ " , valeur"+ point3[i]);
8  }

```

Objets Dynamiques

- Lire:

```
a= point3.x;
```

- Modifier :

```
point3.x=5;
```

- Ajouter :

```

//ajout a l'object (unique) point3
point3.distance=function (x1, y1) {
    return Math.sqrt((x1 - this.x)*(x1 - this.x) + (y1 - this.y)*(y
}

```

- Supprimer :

```
delete point3.distance;
```

Ajouter à toute les instances produites par un constructeur

prototype

```
1 Point.prototype.distance=function (x1, y1) {  
2     return Math.sqrt((x1 - this.x)*(x1 - this.x))  
3 }  
4  
5 p2.distance(3,2) //renvoie la distance de p2 (4,3) au point  
6 (3,2)
```

Compléments hors programme examen NFA016

On manipule des références

Variable : nom qui fait référence à une valeur en mémoire.

Valeurs :

- primitives : string, number, booleans, ...
- objets

en fait même les valeurs primitive peuvent être vues comme des objets :

```
alert(1["toString"]());
```

portée (scope) des variables

Variables globales

Quelques variables de portée globale pointant vers des valeurs primitives:

```
1 var name = "Dupond D.";  
2 var taille= "180";  
3 var estProgrammeur =true;  
4 var aimeJavaScript = true;  
5 // Teste si deux variables réfèrent la même valeur  
6 estProgrammeur === aimeJavaScript; //renvoie true
```

liées a l'object globale (window) dans un navigateur



portée (scope) des variables

Objet et chaîne de prototype

Objets : collection de référence à d'autre objets et à un **prototype**. La chaine de prototype permet d'accéder aux propriétés qui ne sont pas dans l'object mais dans "ses parents"

```
1 // Créer un point
2 var p = {
3     x:0,
4     y:0,
5     distance:function (x1, y1) {
6         return Math.sqrt((x1 - this.x) * (x1 - this.x) + (y1 -
7             this.y) * (y1 - this.y));
8     }
9 }
10 // Créer un heritier
11 var prouge = Object.create(p);
12
13 // redefinir une propriété some properties locally
14 prouge.x = 2;
15
16 //ajouter une propriété
17 prouge.col = "red";
18
19 // chercher dans la chaine de prototype
20 prouge.distance(2,3);
```