Chapitre 6

Les objets de JavaScript

Dans la Figure 5.3 du chapitre précédant, nous avons vu tous les objets prédéfinis de JavaScript . Dans ce chapitre, nous allons voir les propriétés et les méthodes importantes de chacun de ces objets.

6.1 L'objet Array

6.1.1 Tableaux

Array veut dire tableau en français. Un tableau est une suite finie de variables auxquels on peut accéder par indice (au lieu de leur accéder par leur nom). L'indexation des tableaux débute à 0. Cette notion de tableau existe dans tous les langages de programmation.

En JavaScript l'objet Array est utilisé pour créer des tableaux. La syntaxe est

1) Si on connaît le contenu du tableau

```
var monTableau = new Array(E0, E1, ..., En);
```

de façon équivalente, on peut aussi déclarer un tableau avec la syntaxe suivante :

```
var monTableau=[E0, E1, ..., En];
```

on a monTableau[0]=E0, monTableau[1]=E1, etc

```
<script type="text/javascript">
T=new Array('a','b',3,5);
document.write(T[0],T[2]) //retourne a 3
T[0]='A'; // écrase l'ancienne valeur et la remplace par A
document.write(T[0]) //retourne A
</script>
```

2) on connaît le nombre des éléments du tableau.

```
Tableau = new Array(nombre);// ou  Tableau=[nombre]
```

pour compléter ce tableau on fait des affectations Tableau[0]=V0, Tableau[1]=V1, ...

```
<script type="text/javascript">
T=new Array(3);
```

```
T[0]='pomme'; T[1]='banane';T[2]='poire';
U=[3];
U[0]='fraise'; U[1]='orange'; U[2]='mangue';
document.write(T[0],T[2]) //retourne pomme poire
document.write(U[0],U[2]) //retourne fraise mangue
</script>
```

3) Si on ne connaît pas le nombre des éléments du tableau à l'avance.

```
Tableau = new Array(); ou Tableau=[]
```

pour compléter ce tableau on fait des affectations : Tableau[0]=E0, Tableau[1]=E1, ...

```
<script type="text/javascript">
T=new Array();
T[0]='pomme'; T[2]='poire';
U=[];
U[0]='fraise'; U[1]='orange'; U[2]='mangue';
document.write(T[0],T[1]) //retourne pomme undefined
document.write(U[0],U[2]) //retourne fraise mangue
</script>
```

On peut parcourir les tableaux en utilisant une boucle. La propriété length retourne la longueur de tableau.

```
<script language="javascript">
var T = new Array();
T[0] = "orange";
T[1] = "pomme";
T[2] = "kiwi";
for (i=0;i<T.length;i++){document.write(T[i] + "<br />");}
</script>
```

6.1.2 Tableaux associatifs

Les tableaux associatifs sont des tableaux indexés par des chaînes de caractères et non pas par des indices numériques.

```
var T = new Array(); T['Clé1'] = 'Val1'; T['Clé2'] = 'Val2 '; ...
```

ou en utilisant les parenthèses

```
var T = {"clé1" :"val1" , "clé2" :"val2" , ... };
```

```
<script type="text/javascript">
var T = new Array();
T['Clé1'] = 'Val1';
T['Clé2'] = 'Val2 ';
```

```
for( x in T){document.write(x + ' : ' + T[x] + ' '+'<br/>')};

var Notes = {'Ali':[12,13,10,16,13], 'Med':[15,10,8,12,14],
      'Saloua':[10,12,15,4,18]};
document.write(Notes['Ali'],'<br>'); // retourne 12,13,10,16,13
document.write(Notes['Ali'][3]); // retourne 16
</script>
```

6.2 Tableaux prédéfinis de JavaScript

6.2.1 Objet prédéfini images[]

images : est un tableau contenant toutes les images de document HTML. Il a comme propriétés src, width, height, id, ...

document.images[i] indique la (i+1)eme image du document en cours.

document.images.length : retourne le nombre d'image dans le document HTML en cours.

document.images[i].src : contient le le chemin et nom de fichier de la (i+1)eme image dans le document HTML en cours.

Au lieu d'utiliser les indices pour le tableau images on peut utiliser l'attribut name de la balise <img> et utiliser document.images['nomImage'].

On peut aussi utiliser l'attribut id pour la balise <img> et la méthode getElementById() de document : document.getElementById('nomId') pour indiquer l'image ayant l'Id='nomId'.

```
<img src='img2.jpeg' id="abc">
<script type="text/javascript">
p=document.getElementById('abc').src
document.write(p); //retourne img2.jpeg
</script>
```

Propriété de l'objet images[]

alt contient le texte alternatif de l'image.

border Contient la valeur de border.

complete Contient un indicateur de fin de chargement de l'image. Vaut true si l'image est complètement chargée et false sinon.

fileSize Contient la taille en octets de l'image (ne fonctionne pas avec tous les navigateurs).

height contient la valeur définie par le paramètre height de la balise <img> en son absence indique la hauteur réelle de l'image.

id contient la valeur définie par le paramètre id de la balise <img>. Cet identifiant s'utilise avec la méthode getElementById() de document. Attention de ne pas confondre id avec name.

name contient la valeur définie par le paramètre name de la balise <img>.

src La propriété src contient le chemin et le nom de fichier de l'image.

width contient la valeur définie par le paramètre width de la balise <img> en son absence indique la largeur réelle de l'image.

6.2.2 L'objet prédéfini links[]

links : est un tableau contenant tous les liens du document HTML en cours. Il a comme propriétés href, target, id, ...

document.links[i] indique le (i+1)eme lien du document en cours.

document.links.length : retourne le nombre de liens dans le document HTML en cours.

document.links[i].href : contient l'URL du (i+1)eme lien dans le document HTML en cours.

Au lieu d'utiliser les indices pour le tableau links on peut utiliser l'attribut name de la balise <a> ... et utiliser document.links['nomLien']

On peut aussi utiliser l'attribut id pour la balise <a> ... et la méthode getElementById() de document : document.getElementById('nomId') pour indiquer le lien ayant l'Id='nomId'.

```
<a href="http://www.google.com" id='abc'> Google </a> <br>
<a href="http://www.yahoo.com" target="blank"> Yahoo </a> <br>
<script type="text/javascript">
document.write('URL du 1er lien est : ',document.links[0].href,'<br>');
document.write('La cible du 2e lien : ',document.links[1].target,'<br>');
document.links[0].href='http://www.youtube.com'; // modifie URL
document.getElementById('abc').textContent='Youtube';
</script>
```

6.2.3 L'objet prédéfinis forms[]

document.forms[i] indique le (i+1)eme formulaire du document en cours.

Au lieu d'utiliser les indices pour le tableau forms on peut utiliser l'attribut name de <form> ...</form> et utiliser document.forms['nomFormulaire'] pour indiquer le formulaire nommé 'nomFormulaire'. On peut aussi utiliser l'attribut id="monId" dans la balise <form> et document.getElementById('monId') pour repérer le formulaire dont l'Id="monId".

document.forms.length : retourne le nombre de formulaire dans le document HTML.

voir exemple ci-dessous.

Propriétés de l'objet forms[]

action Contient l'action définie pour un formulaire document.forms[X].action ; (action utile pour PHP).

elements Tableau de tous les élément d'un formulaire.

encoding Contient le type de " ENCTYPE" des données du formulaire. document.forms[X].encoding.

length Nombre de formulaires que contient le document. document.forms.length

method Contient la "méthode de transmission des données" (get/post) du formulaire

document.forms[X].method ; X est un indice ou nom de formulaire donné par l'attribut name de la balise <form>.

name Contient le nom du formulaire document.forms[X].name
target Fenêtre cible du formulaire. document.forms[X].target

Méthodes de l'objet forms[]

reset() Réinitialise un formulaire : document.forms[X].reset()
submit() Soumet un formulaire : document.forms[X].submit()

6.2.4 L'objet prédéfinis elements[]

elements : est un tableau contenant tous les éléments d'un formulaire. Il s'agit de tous les types dans <input>, <select>, <textarea>, etc

Au lieu d'utiliser les indices pour le tableau elements on peut utiliser l'attribut name de <input> et utiliser document.forms[i].elements['nomElement']. On peut aussi utiliser l'attribut id avec document.getElementById() pour repérer l'élément dont l'id indiqué.

document.forms["nomForm"].elements['nomElement'].value; valeur de l'élément nommé 'nomElement' du formulaire nommé "nomForm".

On peu mixer la désignation par nom et par indice.

```
<form name='ff'>
<input type='text' name="cc">+
<input type='text' id="abc">=
<input type='text' onClick="
a=parseFloat(document.forms['ff'].elements[0].value);
b=parseFloat(document.getElementById('abc').value);
document.forms[0].elements[2].value=a+b;">
</form>
```

6.2.5 Localisation de balise

En plus des attributs name et id, JavaScript permet de localiser des balises d'après leurs noms ou la valeur de leur attribut class. Nous allons voir :

getElementsByTagName

getElementsByTagName(nomBalise) : renvoie tous les éléments du document dont le nom de balise est nomBalise. (remarquer le s dans getElementsByTagName).

```
<p> Paragraphe Paragraphe Paragraphe </p>
<p> texte texte texte </p>
<script type="text/javascript">
var c=document.getElementsByTagName("p");
document.write(c[1].innerHTML); // retourne : texte texte texte
</script>
```

getElementsByClassName

getElementsByClassName(maClasse) renvoie tous les éléments du document ayant un attribut HTML class dont la valeur est maClasse.

```
<div class='abc'> Paragraphe Paragraphe </div>
<div class='xyz'> texte texte texte </div>
<div class='abc'> phrase phrase phrase </div>
<script type="text/javascript">
var c=document.getElementsByClassName("abc");
document.write(c[1].innerHTML); // retourne : phrase phrase phrase
</script>
```