

Que permet JavaScript ?

C'est un langage de programmation incorporé aux balises Html, qui permet d'améliorer la présentation et l'interactivité des pages Web. Ce langage est interprété par le navigateur.

Pourquoi objet ?

Dans une page web, tout élément intégré est considéré comme une entité (un objet) déterminé par des règles. Chaque objet à une spécificité et peut interagir avec d'autres objets
Un paragraphe balisé par les éléments `<p></p>` est un objet par exemple.

Les éléments d'une page web sont hiérarchisés et organisés par un modèle de document d'objets (DOM)

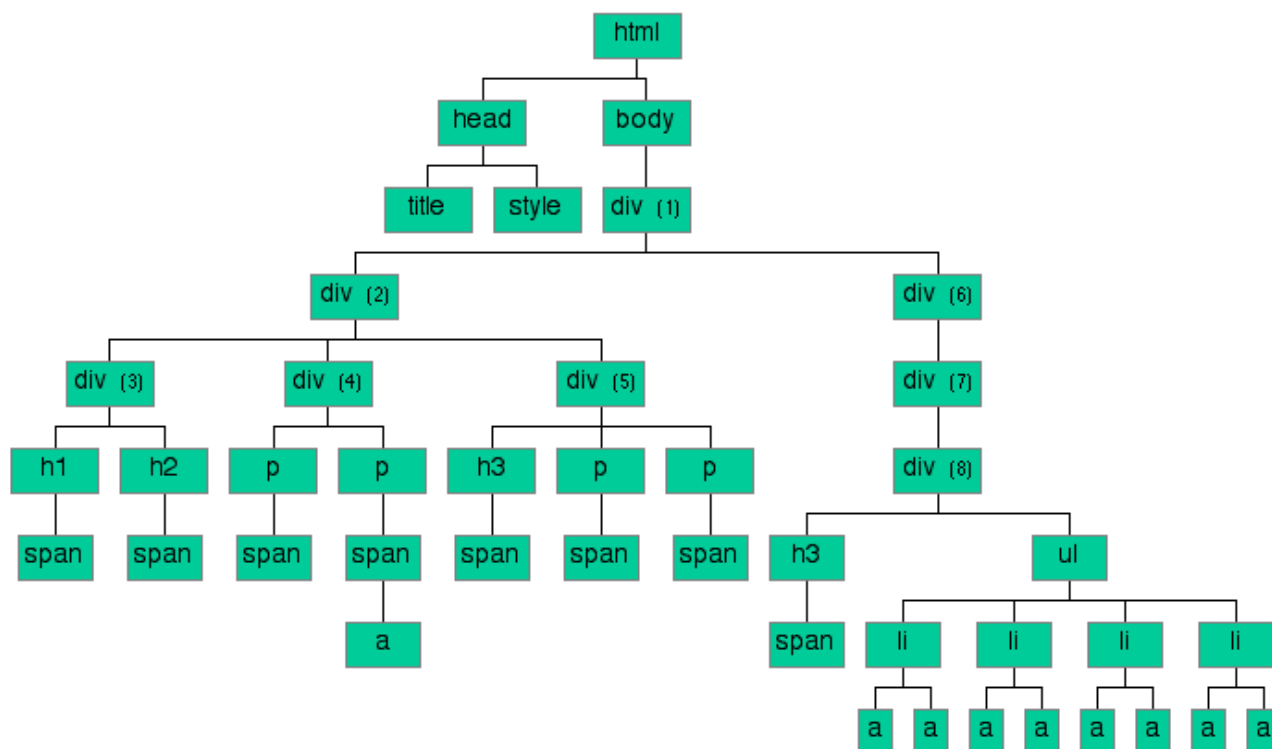


Fig 1 : Exemple d'arbre

Ces éléments sont eux mêmes situés et hiérarchisés en objets imbriqués, dont l'élément de plus haut niveau est la fenêtre de votre navigateur. Ainsi cela donne une hiérarchie comme présentée page suivante.

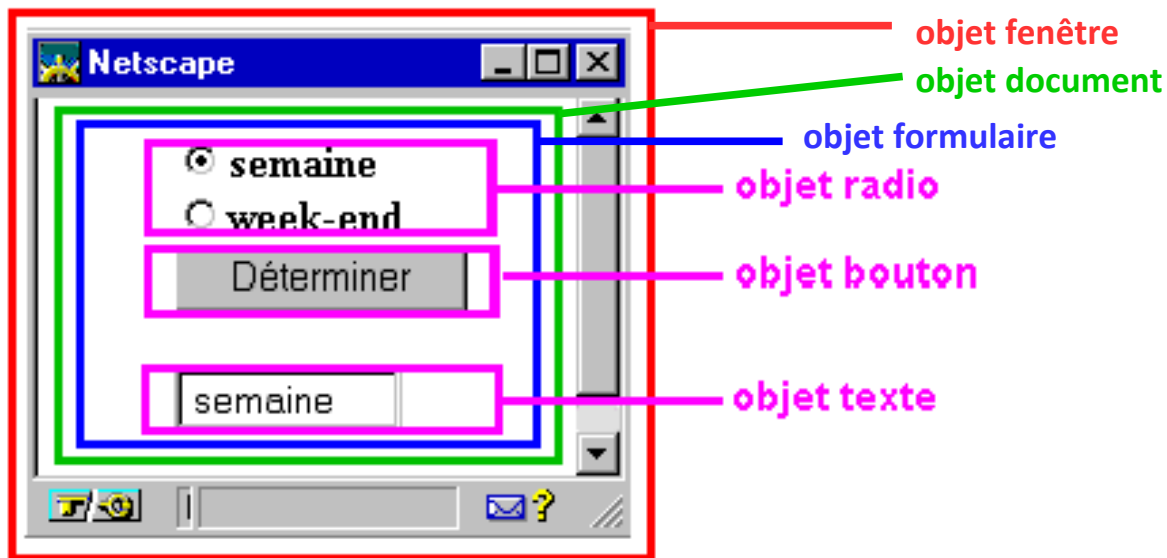


Fig 2 : Hiérarchie des objets

Pour accéder à objet bouton il faudra nommer les éléments supérieurs pour donner son chemin :
fenetre.document.formulaire.bouton

Nous trouvons aussi des objets de niveaux supérieurs :

```

navigator
├── window
│   ├── parent, frames[], self, top
│   ├── location
│   ├── history
│   └── document ----> ["DOM 0"]
│       ├── form[]
│       │   └── elements[] (text, textarea, checkbox,
│       │       password, radio, select, button, submit,
│       │       reset)
│       ├── links[]
│       ├── images[]
│       └── URL
│       └── ...

```

Au démarrage du navigateur, il est immédiatement possible
d'accéder à des informations concernant le navigateur client, les documents HTML, l'écran
d'accéder au seul objet navigator avec les propriétés : nom, version, plugins installés,...
d'accéder à chaque objet window, un par fenêtre et cadre du document HTML, ainsi que ceux créés par
les balises HTML
d'accéder au seul objet screen avec les propriétés largeur, hauteur, ...

Nous allons maintenant, via ses notions basiques à utiliser ce langage

Création du 1^{er} projet javascript

Il faut organiser son projet avec plusieurs fichiers :

- index.html (fichier écrit en html, point d'entrée de tout site web)
- script.js (fichier contenant le programme javascript)



Recopier ces lignes puis **tester** le fonctionnement dans un navigateur web.

```
<!DOCTYPE html>
<html>
<head>
  <title>Premier projet JS</title>
  <meta charset="utf-8"/>
  <script src="script.js" type="text/javascript"></script>
</head>
<body>
  <form name="form1" action="" method="post">
    <input type="button" name="action" id="btnAction" value="Action !"
      onclick="get_mess();" />
  </form>
```

index.html

```
function get_mess(){
  document.getElementById('btnAction').style.color="blue";
}
```

script.js



Indiquer la ligne qui insert le fichier script.js dans le fichier html



Indiquer le paramètre qui modifie la couleur du bouton



Modifier les lignes en italiques puis **tester** le fonctionnement dans un navigateur web.

```
<!DOCTYPE html>
<html>
<head>
  <title>Premier projet JS</title>
  <meta charset="utf-8"/>
  <script src="script.js" type="text/javascript"></script>
</head>
<body>
  <form name="form1" action="" method="post">
    <input type="button" name="action" id="btnAction" value="Action !"
      onclick="get_mess();" />
  </form>
```

index.html

```
function get_mess(couleur='blue'){
  document.getElementById('btnAction').style.color=couleur;
}
```

script.js



Indiquer ce qui a été modifié ainsi que le comportement résultant



Modifier la ligne en italique puis **tester** le fonctionnement dans un navigateur web.

```
<!DOCTYPE html>
<html>
<head>
  <title>Premier projet JS</title>
  <meta charset="utf-8"/>
  <script src="script.js" type="text/javascript"></script>
</head>
<body>
  <form name="form1" action="" method="post">
    <input type="button" name="action" id="btnAction" value="Action !"
      onclick="get_mess('red');" />
  </form>
```

index.html

```
function get_mess(couleur='blue'){
  document.getElementById('btnAction').style.color=couleur;
}
```

script.js



Indiquer ce qui a été modifié ainsi que le comportement résultant



Changer la valeur passée en paramètre pour **'green'** et **tester** de nouveau. Le fonctionnement est-il celui attendu ?



Ajouter les lignes en italiques puis **tester** le fonctionnement dans un navigateur web.

```
<!DOCTYPE html>
<html>
<head>
  <title>Premier projet JS</title>
  <meta charset="utf-8"/>
  <script src="script.js" type="text/javascript"></script>
</head>
<body>
  <form name="form1" action="" method="post">
    <input type="text" value="Bonjour" name="champTxt" id="champTxt1"/>
    <input type="button" name="action" id="btnAction" value="Action !"
      onclick="get_mess('red');"/>
  </form>
</body>
```

index.html

```
function get_mess(couleur='blue'){
  document.getElementById('btnAction').style.color=couleur;
  document.getElementById('champTxt1').value="Click sur le bouton : "
+ document.getElementsByTagName('input')[1].name;
}
```

script.js



Justifier les 2 façons utilisées pour sélectionner des objets dans la page html



Justifier l'utilisation de [1] devant getElementsByTagName('input')



Justifier l'utilisation du signe + à ...Bouton : " + document.getElem...

Les types de variables

JavaScript est un langage typé dynamiquement : on peut y mettre du texte en premier lieu puis l'effacer et y mettre un nombre quel qu'il soit, et ce, sans contraintes.

Les trois types principaux en JavaScript :

- Le type numérique (number) : représente un nombre, que ce soit un entier, un négatif, un nombre scientifique, etc. La valeur peut être spécifiée en valeur entière, en notation décimale (exp : 5.5), scientifique (ex : 2718E-3), ou hexadécimale (exp : 0x3fa).
- Le type chaînes de caractères (string) : représente n'importe quel texte. On peut l'assigner de deux façons différentes : entourées de guillemets " " ou d'apostrophes ' ' (à privilégier). Dans ce dernier cas, pour insérer une apostrophe au sein du texte, il faut la faire précéder de l'anti-slash : \'
- Le type booléen (boolean) : représente un type ayant deux valeurs : vrai (true) ou faux (false). Par extension, toute variable non nulle d'un type quelconque est équivalente à true.



Nous allons maintenant afficher un texte dans la page html à chaque clique sur le bouton *Action* !



Ajouter les lignes en italiques puis **tester** le fonctionnement dans un navigateur web.

```
<!DOCTYPE html>
<html>
<head>
  <title>Premier projet JS</title>
  <meta charset="utf-8"/>
  <script src="script.js" type="text/javascript"></script>
</head>
<body>
  <form name="form1" action="" method="post">
    <p id="parTxt1"/></p>
    <input type="text" value="Bonjour" name="champTxt" id="champTxt1"/>
    <input type="button" name="write" value="Write !" onclick="write_mess();"/>
    <input type="button" name="action" id="btnAction" value="Action !"
      onclick="get_mess('red');"/>
  </form>
</body>
</html>
```

index.html

```
var cntLine = 0;

function get_mess(couleur='blue'){
  document.getElementById('btnAction').style.color=couleur;
  document.getElementById('champTxt1').value="Click sur le bouton : "
+ document.getElementsByTagName('input')[1].name;
}

function write_mess(){
  cntLine++;
  var itemTxt = document.createTextNode("Ligne " + cntLine + " ecrire<br/>");
  document.getElementById('parTxt1').appendChild(itemTxt);
}
```

script.js



Justifier pourquoi le nom du bouton affiché dans le champ texte n'est pas *Action* ! lorsque l'on clique dessus.



Modifier alors l'élément nécessaire pour afficher la bonne information



Expliquer l'instruction `cntLine++`



Expliquer l'instruction `document.createTextNode`



Expliquer l'instruction `document.getElementById('parTxt1').appendChild(itemTxt);`



Donner une justification sur l'affichage de la balise `
` dans le texte qui s'affiche.



Ajouter les lignes en italiques puis **tester** le fonctionnement dans un navigateur web.

```
var cntLine = 0;

function get_mess(couleur='blue'){
    document.getElementById('btnAction').style.color=couleur;
    document.getElementById('champTxt1').value="Click sur le bouton : "
+ document.getElementsByTagName('input')[1].name;
}

function write_mess(){
    cntLine++;
    var itemBr = document.createElement("br");
    var itemTxt = document.createTextNode("Ligne " + cntLine + " ecrire");
    document.getElementById('parTxt1').appendChild(itemTxt);
    document.getElementById('parTxt1').appendChild(itemBr);
}
```

script.js



Expliquer l'apport de la modification



A vous de **jouer** maintenant !



Utiliser l'instruction suivante :

https://www.w3schools.com/jsref/met_node_removechild.asp



D'autres instructions se trouvent sur cette page :

https://www.w3schools.com/jsref/dom_obj_all.asp

http://www.w3schools.com/jsref/dom_obj_document.asp



Ajouter une fonction en JavaScript ainsi qu'un bouton pour effacer le paragraphe ayant pour identifiant id="parTxt1"

Projet 2 : jouons avec l'objet date



Le jeu d'instructions se trouvent sur cette page : http://www.w3schools.com/jsref/jsref_obj_date.asp



Recopier les lignes dans des nouveaux fichiers puis **tester** le fonctionnement dans un navigateur web.

```
<!DOCTYPE html>
<html>
<head>
  <title>Deuxieme projet JS</title>
  <meta charset="utf-8"/>
  <script src="script2.js" type="text/javascript"></script>
</head>
<body>
  <form name="form2" action="" method="post">
    <input type="text" value="" name="champTxt" id="champTxt1"/>
    <input type="button" name="datetime" value="Go !" onclick="get_datetime();" />
  </form>
</body>
</html>
```

index2.html

```
var date_time = new Date();

function get_datetime(){
  var hrs = date_time.getHours();
  var min = date_time.getMinutes();
  var sec = date_time.getSeconds();
  var jrs = date_time.getDate();
  var mth = date_time.getMonth()+1;

  document.getElementById('champTxt1').value="Date: "+jrs+"/"+mth+" Heures: "
+hrs+": "+min+": "+sec;
}
```

script2.js



Expliquer le comportement du fonctionnement



Apporter les modifications nécessaires pour actualiser la date lors du click sur le bouton.



A vous de **jouer** maintenant !



Ajouter une fonction qui appelle votre fonction et qui permet de mettre à jour l'heure dans le champs toutes les secondes de manière automatique.



Le jeu d'instructions se trouvent sur cette page :

http://www.w3schools.com/jsref/obj_window.asp

rubrique : Window Object Methods



Apporter les modifications à la date pour afficher le jour littéral à l'aide d'un tableau. **Faire vérifier**

Projet 3 : stockage d'informations avec les Cookies



Lire le document sur le traitement des Cookies en JavaScript situé à :

https://www.w3schools.com/js/js_cookies.asp



Réaliser une page avec un formulaire qui prend le nom, prénom et age. **Ajouter** un bouton.

```
<!DOCTYPE html>
<html>
<head>
  <title>Troisieme projet JS</title>
  <meta charset="utf-8"/>
  <script src="script3.js" type="text/javascript"></script>
</head>
<body>
  <form name="form3" action="" method="post">
    <input type="text" value="" name="prenom" id="prenom"/>
    <input type="text" value="" name="nom" id="nom"/>
    <input type="number" value="" name="age" id="age"/>
    <input type="button" name="record" value="Enregistrer" onclick="enrCookie();" />
  </form>
</body>
</html>
```



Ajouter ensuite un évènement onclick sur le bouton qui permettra, lors du clique sur le celui-ci, d'enregistrer les informations du formulaire dans un cookie.



Créer ensuite une autre page qui permettra de récupérer les informations contenu dans le cookie et de les afficher.