

# Le langage Javascript

## Objectif

L'objet de ce TP est d'illustrer le langage Javascript. Dans un premier temps, nous nous attacherons à comprendre les bases de ce langage. L'utilisation avancée (ie l'utilisation des objets complexes concernant le document) sera abordée dans le TP suivant.

Les documents de référence : <http://www.mozilla.org/js/language/>

## 1 Un document HTML dynamique simple

Ecrire un document HTML simple contenant uniquement un titre et un message d'accueil dans le corps du document ("Bienvenue sur ce site").

### 1.1 Du code dans le document

En vous inspirant des exemples du cours, ajouter des instructions javascript qui ajoutent le texte suivant au message d'accueil ("Texte dynamique")

### 1.2 Une boucle

Ecrire le code nécessaire pour afficher la table de multiplication d'un entier  $n$  (une variable dont la valeur est affectée dans le code). Le résultat sera structuré par un tableau.

### 1.3 Un premier pas vers l'interactivité

Les formulaires HTML sera présentée plus tard. Cependant, Javascript propose un moyen simple de demander des données à l'utilisateur, la fonction `prompt()`. (<http://www.w3schools.com/js/default.asp>).

Modifier le programme précédent pour que la valeur de  $n$  soit demandée à l'utilisateur. Modifier le programme précédent pour que la page demande une confirmation avant d'afficher le tableau avec la fonction `confirm()`.

Afficher un message pour prévenir l'utilisateur que la table va être affichée avec la fonction `alert()`.

### 1.4 Les fonctions

Modifier votre page pour qu'elle utilise maintenant une fonction `afficheTable(n)` que vous définirez. Dans le cas où cette fonction est appelée sans paramètre, elle devra afficher la table de 5.

### 1.5 Un programme externe

Modifier la page précédente pour que la fonction soit définie dans un fichier externe.

## 2 Une utilisation simple des formulaires

Dans cette section, nous allons utiliser intuitivement les formulaires pour rendre une page interactive. Nous allons mettre en place pour cela un petit QCM très simple.

Pour le moment nous considérerons qu'un formulaire est un élément html dont le nom est `form` et qui possède un attribut `name` qui identifie le formulaire. Il peut contenir des éléments `input` qui représentent les entrées. Chaque entrée possède un attribut `type` (pour le moment `text`, `radio`, ou `button`).

## 2.1 Ecrire un formulaire simple

Mettre en place un formulaire simple (identifié par le nom "qcm") qui affiche une question et propose trois choix possibles (avec des boutons radio). Vous ajouterez un bouton "valider" et un champ texte "résultat" (ce champ texte sera identifié avec un attribut name="resultat").

## 2.2 Traiter la réponse

Déclarer une variable globale `resultat` initialisée à "pas de réponse". Ecrire une fonction `reponse(r)` qui prend en paramètre le numéro de la réponse qui modifie la variable `resultat` ("bravo !" ou "c'est faux !").

En utilisant les événements `onClick(r)` sur les boutons radio modifier la valeur de résultat en appelant la fonction `reponse`. En cas, de clic sur le bouton "valider" exécuter une fonction `affiche()` qui modifie la valeur du champ résultat avec la valeur de `resultat`. On accède à la valeur d'un champ avec un exprssion de la forme (`document.NomFormulaire.NomChamp.value=resultat`).  
**Utiliser la fonction alert pour faire des tests.**

Ecrire un autre formulaire qui permet saisir votre année de naissance dans un champ et qui affiche votre âge en 2050 quand on clique sur le bouton "calculer".

## 3 Pour aller plus loin (Facultatif lors de la seconde séance)

Un des utilisations très courantes de javascript est la vérification coté client de formulaires.

Dans un nouvelle page, écrire un formulaire qui permet à un utilisateur de saisir ses informations personnelles (nom, prénom, age, email). Ajouter trois boutons valider, tester et effacer (de type reset).

Quand on clique sur le bouton afficher une alerte si un des champ est vide et dire lequel.

Vérifier que l'âge est bien un nombre.

On peut produire un résultat simple en écrivant le résultat (un document html dans un autre fenetre). On procède en deux temps :

1. On ouvre un nouvelle fenêtre : `result=open("", "resultat", "toolbar=no, location=no, directories=no, status=no, menubar=no, scrollbars=1, resizable=1, top=1, left=50, width=500, height=200");`
2. On écrit le code HTML à l'intérieur avec plusieurs : `result.document.write(...)`.

Modifier votre page pour qu'un clic sur bouton valider vérifie les champs et ouvre un page HTML qui affiche les informations sous forme de tableau.

Peux-t-on vérifier les champs au fur et à mesure de la saisie (regarder les autres types d'évènements) ?

Ecrivez une application javascript qui permet de présenter un QCM sous la forme d'un formulaire HTML puis qui affiche les résultats dans une autre page.

## 4 Les objets du navigateur

Dans cette partie vous allez expérimenter les objets de javascript associés au navigateur :

- navigator
- window
  - document

Nous allons illustrer l'utilisation de ces différents objets. Pour connaître la définition de chacune des classes (propriétés et méthodes), vous vous reportez au site suivant : <http://fr.selfhtml.org/javascript/objets/index.htm>.

Pour les exercices suivants créer un document HTML minimal `testJS.html`.

### 4.1 L'objet navigator

A l'aide de l'objet navigator, affichez dans le document HTML les informations concernant le navigateur utilisé (type de navigateur et sa version, langue, ...).

En utilisant, le parcours d'un tableau afficher la liste des *plugins* disponibles dans un tableau HTML.

## 4.2 L'objet window

En fait, nous l'avons déjà utilisé mais il peut être omis si l'on désigne un objet de la fenêtre courante. Cet objet permet d'accéder à l'adresse à travers laquelle on accède au document (`location`), au contenu du document (`document`), à l'historique de navigation (`history`) et plus encore.

- En utilisant l'objet `location`, afficher le nom du serveur et le chemin pour atteindre le document.
- En utilisant l'objet `history`, ajouter un bouton qui permet de revenir au document précédemment contenu dans la fenêtre quand on clique dessus.

`open()`, `prompt` et `alert()` sont en fait des méthodes de `window`. Différentes méthodes permettent de déplacer la fenêtre (`moveTo()`) ou son contenu (`scrollBy()`).

- *En enregistrant la valeur de retour de la méthode `open` dans une variable. Ouvrir une nouvelle fenêtre (sans barre d'outils) et écrire le message "hello" à l'intérieur.*
- *Ajouter un bouton qui permet de déplacer la fenêtre de 10px vers la droite.*

## 5 L'arbre du document et javascript

### 5.1 L'objet document

L'objet `document` est très important, il permet d'accéder à l'ensemble des éléments qui le compose de plusieurs manières.

#### 5.1.1 Deux éléments spéciaux

Les propriétés `documentElement` et `body` permettent d'accéder respectivement à l'élément `html` et à l'élément `body`.

#### 5.1.2 Accès via les tableaux d'éléments

Javascript range les références vers les éléments dans des tableaux spécifiques (`anchors`, `forms`, `images`, `links`, `stylesheet`, ...) dans l'ordre de leur apparition (La première image `image[0]`, la cinquième `image[4]`). La propriété `length` des tableaux permet de connaître le nombre d'éléments.

#### 5.1.3 Les méthodes de parcours du document

Les trois méthodes `getElementById("ValeurDeId")`, `getElementsByName("ValeurAttrName")` et `getElementsByTagName("nomElement")` permettent d'obtenir un tableau de références vers des éléments en fonction de la valeur de leur attribut `id` ou de leur attribut `name` ou bien du nom de l'élément.

### 5.2 L'objet node

Tous les noeuds du document sont représentés par un objet qui offre en particulier les propriétés suivantes : `id`, `tagName` (nom de l'élément), `nodeName` (valeur de l'attribut `name`), `nodeType` (1=élément, 2=attribut, 3=texte) et `nodeValue` (valeur d'un noeud texte ou d'un attribut). Mais aussi `className` (la valeur de l'attribut `class`) et `title` (la valeur de l'attribut `title`).

- *Ajouter des paragraphes dans le document. Calculer et afficher automatiquement le nombre de paragraphes.*
- *Ajouter un attribut `id='unique'` et un attribut `title='Un P unique'` à l'un des paragraphes. Afficher la valeur de l'attribut `title` de ce paragraphe dans le document.*
- *Mettez plusieurs paragraphes dans la classe 'classeX' et ajouter un titre à chacun. Afficher la liste des titres des paragraphes de la classe 'classeX'.*

### 5.2.1 Le parcours de l'arbre du document

Il est possible d'accéder à d'autres noeuds à partir de n'importe quel noeud de l'arbre du document :

- la méthode `getElementsByTagName("nomElement")` est utilisable aussi sur les noeuds.
- `childNodes[]` (noeuds textes et éléments fils) et `attributes[]` (les attributs d'un élément)
- `parentNode`, `firstChild`, `lastChild`, `nextSibling`, `previousSibling` accès direct à un élément en fonction des relations généalogiques.

Il existe des propriétés particulières pour les éléments `table` : `caption` (la légende), `cells[]` (les cellules du tableau), `rows[]` (les lignes si l'élément est `thead`, `tbody` ou `tfoot`, `tBodies[]` (les éléments `tbody`), `tfoot` et `thead` (l'entête et le pied). Pour les `tr` et les `td`, `cellIndex` et `rowIndex` dont la position (colonne, ligne).

- *Ecrire un tableau de 4x4 entiers en HTML d'identifiant 't' et de titre 'un tableau'. Calculer (en javascript) et afficher la somme des entiers de la première colonne. La méthode `parseInt(s)` convertit une chaîne en entier.*

Les méthodes suivantes sont disponibles pour les suppressions :

- pour un `tr` : `deleteCell(cellIndex)`.
- pour un `thead`, `tbody` or `tfoot` : `deleteRow(rowIndex)`.
- pour un `thead` ou un `tfoot` : `deleteTfoot()`, `deleteThead()`.
- *Ajoutez un bouton qui supprime la première colonne. Notez que la somme n'est pas mise à jour.*

### 5.2.2 La création d'attributs et d'éléments

Pour modifier l'arbre du document on procède en deux étapes : On crée un nouvel élément puis on l'ajoute au document :

1. Création : Les trois méthodes suivantes `createAttribute("nom")`, `createElement("nom")` et `createTextNode("valeur")` s'applique sur l'objet `document` et permettent de créer des noeuds dans l'arbre du documents.
2. Ajout (appliqué sur un noeud) : `appendChild(nouveauNoeud)`, `insertBefore(nouveauNoeud, noeudReference)` et `replaceChild(nouveauNoeud, noeudReference)` permet de placer le nouveau noeud (si le noeud existe déjà, il est déplacé).

Il est aussi possible d'accéder au contenu d'un élément (ou de le modifier) avec la propriété `innerHTML`.

Les noeuds peuvent être supprimés avec la méthode `removeChild(nodeReference)`.

- *Modifier la page pour la somme soit mise à jour lors de la suppression de la 1ere colonne*
- *Ajouter un bouton "déplacer unique" qui définit le paragraphe d'id 'unique' comme le premier élément du body.*
- *Créer un bouton qui ajoute une colonne d'entiers aléatoires (entre 1 et 10) au début du tableau.*

## 5.3 La présentation et javascript

Javascript propose un ensemble de méthode et de propriété permettant de préciser la présentation des éléments. Dans le cadre de ce cours, cette approche est déconseillée. A la place nous utiliserons CSS, l'utilisation de javascript pourra alors se faire principalement de deux façon soit en modifiant l'attribut `class` (avec la propriété `className`) des éléments pour changer la règle qui s'applique à l'élément, soit en indiquant via la propriété `style` des propriétés CSS (`node.style.backgroundColor='red'`).

- *Ajouter deux boutons qui permettent de changer la couleur de fond du tableau en rouge ou en bleu en modifiant directement le style.*
- *Ajouter deux règles CSS qui associent aux éléments des classes `tableau1` et `tableau2`, les couleurs de fond. Ajouter deux boutons qui change la classe du tableau en `tableau1` ou `tableau2`*

## 6 Pour aller plus loin

Une page de calcul des moyennes en utilisant HTML, CSS et javascript. Mettez en place une page qui affiche un tableau (vide au début mis à part l'en-tête) qui affiche le nom d'un élève, ses notes en français, mathématiques et histoire et sa moyenne.

1. Votre page devra compter un formulaire qui permet de saisir les informations pour ajouter un nouvel élève et ses notes dans le tableau. La moyenne de l'élève sera calculée automatiquement.
2. Les noms des élèves reçus ( $\text{moyenne} \geq 10$ ) seront écrits en vert et les autres en rouge.
3. Ecrire une fonction permet de mettre en couleur (passée en paramètre) la notes la plus élevée (pour chaque matière et pour la moyenne)
4. Ajouter un bouton "initialiser" pour ajouter un élève construit au hasard (vous définirez un tableau de prénoms).
5. Ajouter un formulaire qui permet de saisir le nom d'un élève et son appréciation qui sera affichée à coté du tableau.
6. En utilisant CSS, proposer une solution pour que seule l'appréciation de l'élève pointé par la souris est affichée.