

Les bases de Javascript.

Nous ne ferons probablement qu'un ou deux TD sur javascript, le temps de voir comment cela fonctionne, ce que l'on peut faire avec. La syntaxe ne devrait pas poser de problème, `ActionScript` en reprenant la syntaxe. Pensez toujours que javascript est exécuté côté **client**. De nombreux tutoriels et exemples existent sur internet, voir par exemple <http://www.w3schools.com/JS/default.asp> ou <http://sergep.developpez.com/tutoriels/javascript/introduction/#sommaire>. Si vous ne savez pas comment faire une question, consultez les !

Exercice 1 : premier script. `document.write()`.

Faire une page web comprenant un script javascript (délimité par les tag `<script type="text/javascript">` et `</script>`) permettant d'écrire "Bonjour tout le monde!". Comment faire pour que ce texte soit un titre1 ?

Exercice 2 : variables, fonctions, boucles, tableaux...

Les variables sont déclarées avec le mot clé `var`. Les boucles (`while`, `for`, ...), tests (`if`, `switch`...), etc... ont la même syntaxe que celle à laquelle vous êtes habitués.

1. Déclarez deux variables javascript `a` et `b`, initialisez-les avec deux nombres de votre choix. Faites une fonction en javascript qui calcule la somme de deux nombres. Appelez la avec `a` et `b` en argument. Affichez une ligne qui indique si le résultat de votre somme est positif ou négatif.
2. Déclarez un tableau `mots` (`var mots=new Array()`). Remplissez le avec 3 mots. Ensuite, à l'aide d'une boucle `for`, parcourez ce tableau pour imprimer à l'écran les mots qu'il contient. La longueur du tableau est disponible à l'aide de la méthode `length`...

Exercice 3 : messages d'alerte, de confirmation. `alert()`, `confirm()`.

1. Faites une fonction javascript `alerte` chargée d'envoyer une popup d'alerte (`alert("alerte!")`). Ensuite, dans le corps de votre page, créez un bouton avec comme libellé "Montrer l'alerte" : quand on appuie dessus, la fonction `alerte` est appelée. Pour faire cela, il faut rajouter à votre `input` un attribut `onclick="alerte()"` qui gère ce qui se passe lorsque l'on appuie sur le bouton. En l'occurrence, cela appelle la fonction javascript `alerte()`.
2. Faites le même exercice, mais avec une boîte de confirmation (`confirm()`) au lieu de la boîte d'alerte.
3. Faites afficher un texte qui indique si l'utilisateur a appuyé sur "OK" ou sur "cancel".

Exercice 4 : infos sur le visiteur.

Rappel : accéder à un attribut non privé `attribut1` d'un objet `o` : `o.attribut1`.

Le navigateur du client est reconnu et géré par l'objet `navigator`. L'attribut `appName` de cet objet indique le type de navigateur qu'utilise le client. Faites afficher ce navigateur par un script. En cherchant sur internet les autres attributs de l'objet `navigator`, cherchez à faire imprimer par votre script le plus de renseignements possible sur le navigateur du client.

Exercice 5 : accéder/modifier les différents objets du document. Le DOM.

La page web sur laquelle travaille le script javascript est gérée par un objet `document`. Cet objet possède de nombreux attributs et méthodes qui permettent d'accéder et/ou modifier son contenu, en particulier celui des différents composants de votre page web. Vous avez déjà vu `document.write()` qui permet d'écrire sur la page web. Deux méthodes très utiles sont `getElementById('IdComposant')` et `getElementsByName('NameComposants')`. Attention ! Le second renvoie un **tableau**, plusieurs composants pouvant avoir le même nom. Vous pouvez ensuite utiliser des méthodes pour accéder aux informations du ou des composants obtenus. Par exemple, `document.getElementById('lien1').innerHTML` est une chaîne de caractère contenant le texte compris entre les balises `<a id='lien1'>` et `</a>`.

1. Faites une page web avec un titre1 (balise `<h1>`) et un lien.
2. Faites une fonction javascript qui sera appelée lorsque l'utilisateur clique sur le titre : un message d'alerte s'ouvre alors avec le texte du titre.
3. Faites une fonction javascript qui change le texte du lien lorsque la souris passe dessus. Essayez de faire la même chose en utilisant `document.anchors`, qui est un tableau contenant tous les liens de la page courante. Attention, il faut absolument que votre lien ait un attribut `name`.

Exercice 6 : réagir aux événements côté client.

Javascript permet de prendre en compte ce qui se passe côté client, ce que PHP, qui tourne côté serveur, ne peut pas : on peut ainsi lancer une action correspondant à un click de souris (**onclick**), à un appui sur le clavier, (**onkeypress**), au passage de la souris sur une zone (**onmouseover**) etc...

1. faites une page web avec une petite image **im1** dedans (le logo de l'IUT par exemple).
2. Récupérez une autre petite image **im2** de votre choix. Le but est que, quand la souris survole l'image (**onmouseover**), **im2** s'affiche en lieu et place de **im1**. Quand le pointeur de la souris quitte la zone de l'image (**onmouseout**), il faut afficher de nouveau **im1**. C'est donc l'attribut **src** de votre image qu'il faudra changer au moment où se produisent ces événements...

Pour aller plus loin : bien d'autres possibilités

D'autres exemples très simples sont disponibles ici : http://www.w3schools.com/JS/js_examples.asp.

Vous pourrez ainsi apprendre à :

- Afficher un texte différent selon la partie d'une image que survole le curseur,
- Créer des horloges, des chronomètres (en combinant avec l'exercice 5 → horloge digitale).
- Donner les coordonnées du point sur lequel l'utilisateur a cliqué.
- Dire sur quel bouton l'utilisateur a cliqué.
- Récupérer le texte de tous les liens contenus dans votre document, les changer dynamiquement.
- Changer la taille d'une image.
- Ouvrir une nouvelle fenêtre, recharger une fenêtre, etc (objet **window**)...

Javascript permet aussi, par l'intermédiaire de l'objet **XMLHttpRequest**, de faire de l'AJAX, qui est une technique de programmation en javascript (utilisant aussi des fichiers XML). Vous pouvez par exemple utiliser cette technique pour changer les items d'une liste2 en fonction de l'item sélectionné sur une liste1 et ce, sans recharger toute la page.