

Les bases de Javascript.

Le but de ce Td est de découvrir les bases du javascript ; ce langage nous permettra de réaliser en première année un jeu en ligne, puis, en deuxième année, d'utiliser ajax, de définir des effets d'animation sur nos pages. De nombreux tutoriels et exemples existent sur internet, voir par exemple <http://www.w3schools.com/JS/default.asp> ou <https://grafikart.fr/formations/debuter-javascript>. N'hésitez pas à les consulter, mais méfiez-vous : il y a souvent plusieurs façons d'écrire les instructions, en javascript, pour obtenir un résultat donné ; j'attendrai en TP et en examen que vous utilisiez les syntaxes vues en cours..

Si quelque chose ne fonctionne pas, pensez à vérifier la console d'erreur en appuyant sur F12 sous chrome ou firefox.

Exercice 1 : premier script. `document.write()`, `alert()`, `console.log()`

1. 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!" à l'aide de l'instruction `document.write()` (qui crée un nouveau document). Comment faire pour que ce texte soit un titre ?
2. Rajoutez maintenant l'instruction `alert('Hello tout le monde')` dans votre code javascript. Qu'est ce que cela fait ? A quoi sert la fonction `alert()` ?
3. Ensuite, écrivez un texte ("bonjour depuis la console" par exemple) dans la console (grâce à l'instruction `console.log()`) : sous firefox ou sous chrome, vous y accédez en appuyant sur F12 (ou click droit -> inspecter/examiner l'élément). C'est également que les messages d'erreur s'afficheront : si ce que vous vouliez faire ne fonctionne pas, le premier réflexe à avoir est de consulter la console !

Exercice 2 : messages d'alerte, de confirmation. `alert()`, `confirm()`.

Pour cette exercice, je vous expliquerai d'abord ce qu'est une fonction, et vous aurez également besoin de connaître le if.

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 3 : variables, fonctions, boucles, tableaux... Initiation à l'algorithmique.

Les variables sont déclarées avec le mot clé `var` ou `let`. Les boucles (`while`, `for`, ...), tests (`if`, `switch`...), etc... vous seront présentés, vous essaierez de répondre aux questions à partir de mes explications. On reviendra sur ces instructions dans le TP jeu !

1. (variables) Déclarez deux variables javascript a et b, initialisez-les avec deux nombres de votre choix.
2. (affectation) Augmentez la valeur de a de 2. Multipliez la valeur de b par 2.
3. (concaténation) créez une autre variable, affectez-lui comme valeur la somme de a et b. Affichez une phrase disant "la somme de a et b est " suivi de la valeur de cette autre variable. Quand cela fonctionne, faites de même, mais en remplaçant a et b de la phrase précédente par leurs valeurs.
4. (fonctions) 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.
5. (tableaux, boucles) 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 4 : 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 caractères contenant le texte compris entre les balises `` et `++`.

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 5 : 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...

Exercice 6 : `setTimeout()` et `setInterval()`.

Ces deux fonctions vont être essentielles pour les animations : elles permettent respectivement de lancer une action au bout d'un certain temps (`setTimeOut()`), et de lancer une action indéfiniment, à intervalle de temps régulier (`setInterval()`). Attention, mettre un `setTimeOut()` dans une boucle ne donnera souvent pas un comportement similaire à un `setInterval()`. En effet, les timer inclus dans la boucle for pourraient être lancé après que la boucle soit finie...

On travaille ici sur une page comportant un titre (`<h1>`), une div contenant un paragraphe et une image.

1. Faites en sorte que la propriété `margin-top` (en javascript on accède à cette propriété d'un élément en faisant : `element.style.marginTop` où `element` est à remplacer par un élément de la page) de votre div change quand vous cliquez sur le titre : elle passe à 300px.
2. **`setTimeout()`** : Faites ensuite de sorte qu'à chaque clic sur le titre, la propriété `marginTop` de la div passe à 300px, comme à la question précédente, à la différence que maintenant au bout de 2s elle doit revenir à sa position initiale.
3. **`setInterval()`** : et maintenant, faites en sorte que toutes les 2s la propriété `marginTop` de la div change (d'abord en la faisant varier entre deux positions, puis en faisant descendre de plus en plus la div). Vous commencerez votre fonction en définissant la propriété `marginTop` initiale, puis il faudra la faire augmenter de 5 px en 5px dans le `setInterval()`. Vous pourriez avoir besoin de la fonction `parseInt()` qui permet de convertir une chaîne de caractères en entier.
4. Sur le même principe, cherchez maintenant à ce que la div se déplace tout autour de l'image, indéfiniment. On jouera cette fois sur les propriétés `top`, `left`, `right`, `bottom` de la div (définie en position absolute donc..).