

TP1 : Développement WEB : JavaScript

Préambule et Introduction au langage

Activités du Référentiel :

A4.1.1 Proposition d'une solution applicative
A4.1.7 Développement, utilisation ou adaptation de composants logiciels
A5.2.4 Étude d'une technologie, d'un composant, d'un outil ou d'une méthode

Objectifs :

- ✓ Intérêts / Avantages / Utilisation de code JavaScript au sein du développement web
 - ✓ Introduction au langage : syntaxes de base, fonctions, calcul automatique dans un formulaire
-

PARTIE 1 : PREAMBULE : JAVASCRIPT : A QUOI ÇA SERT ?

Qu'est-ce que le JavaScript ?

Le JavaScript est un langage de script incorporé dans un document HTML. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web. Un langage de script est un langage de programmation interprété.
C'est un langage orienté objet.

Langage interprété : dans ce cas, il n'y a pas de compilation. Le code source reste tel quel, et si on veut exécuter ce code, on doit le fournir à un interpréteur (inclus dans le navigateur web) qui se chargera de traduire au fur et à mesure les instructions du programme.

Le JavaScript est un langage dit *client-side*, c'est-à-dire que les scripts sont exécutés par le navigateur chez l'internaute (le **client**).

Dans quels cas l'utiliser ?

Cela permet en général d'améliorer l'ergonomie ou l'interface utilisateur, mais certains scripts sont peu utiles et servent surtout à ajouter un effet esthétique à la page. L'intérêt du JavaScript est d'exécuter un code sans avoir à recharger une nouvelle fois la page.

- Donner de l'animation aux pages ou pour faciliter la navigation
- Donner de l'effet à un texte
- Contrôle de saisies des formulaires
- Gestion de l'heure et la date, du temps passé par les visiteurs sur votre page
- Améliorer la navigation, menu dynamique
- Gestion de la barre de statut, des cookies, des comportements de la souris...
- Animer du texte et des images
- Réagir aux événements de la souris
- Valider des données localement
- Modifier des éléments XHTML
- Réaliser des calculs
- Gérer des menus
- Détecter le navigateur du visiteur
- Créer des témoins (cookie)

Quels intérêts face à l'HTML/CSS ? face au PHP ?

Vis à vis du langage HTML/CSS et PHP, le JavaScript permet de faire des traitements de données et des vérifications sans avoir à recharger la page (Ex: vérifier qu'une adresse email entrée dans un formulaire est valide sans avoir à soumettre le formulaire au serveur et attendre un retour). On peut aussi s'en servir pour des pages dynamiques.

Face au PHP, son avantage est d'être interprété côté client et non côté serveur.

Par rapport au PHP : facile à activer/désactiver grâce au navigateur, plus rapide car n'est pas exécuté sur le serveur mais du côté client. JavaScript peut être désactivé.

Qu'est-ce que le DOM ? en quoi il intervient avec le JavaScript ?

Le **Document Object Model** (ou **DOM**) est un standard du W3C (World Wide Web Consortium) qui décrit une interface indépendante de tout langage de programmation et de toute plate-forme, permettant à des programmes informatiques et à des scripts d'accéder ou de mettre à jour le contenu, la structure ou le style de documents XML et HTML. Le document peut ensuite être traité et les résultats de ces traitements peuvent être réincorporés dans le document tel qu'il sera présenté. Le DOM est une représentation du code document. Lorsque le navigateur reçoit du code HTML, il l'analyse et construit un arbre du document, c'est ce qu'on appelle le DOM. Le DOM fournira les méthodes et les propriétés nécessaires pour lire, modifier, mettre à jour et supprimer des éléments du document examiné.

JavaScript permet d'opérer sur les objets DOM et de les manipuler.

À partir du moment où cet arbre est construit en mémoire, on peut accéder à n'importe quel élément, attribut, et on peut aussi changer leurs valeurs, créer des éléments, des attributs, en cloner d'autres, en supprimer certains, etc.

Et l'AJAX dans tout ça, c'est quoi ?

L'architecture informatique **Ajax** (acronyme d'*Asynchronous JavaScript and XML*) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005.

Ajax combine JavaScript, les CSS, XML, le DOM et le XMLHttpRequest afin d'améliorer maniabilité et confort d'utilisation des Applications Internet Riches:

- DOM et JavaScript permettent de modifier l'information présentée dans le navigateur en respectant sa structure ;
- L'objet XMLHttpRequest sert au dialogue asynchrone avec le serveur Web ;
- XML structure les informations transmises entre serveur Web et navigateur.

L'idée même d'AJAX est de faire communiquer une page Web avec un serveur Web sans occasionner le rechargement de la page. C'est la raison pour laquelle JavaScript est utilisé, car c'est lui qui va se charger d'établir la connexion entre la page Web et le serveur.

PARTIE 2 : PREMIERS PROGRAMMES EN JAVASCRIPT

La syntaxe JavaScript est quasi-identique au C avec quelques variantes.

Vous pouvez explorer ce langage en suivant les liens :

- // trois grands classiques des cours en ligne : openclassrooms, developpez.com et commentcamarche
<http://fr.openclassrooms.com/informatique/cours/tout-sur-le-javascript>
<http://javascript.developpez.com/tutoriels/introduction/>
<http://www.commentcamarche.net/contents/javascript-2229965047>
- N'hésitez pas à utiliser d'autres sites plus privés

Quelques informations utiles :

- Dans la syntaxe de base, on retrouve le **if else**, **while**, **do while**, **for**..., les commentaires // ou /* ... */
- Pour déclarer une variable, on ne la prétype pas mais on utilise le mot clé var : **var nomVariable** suffit.
- La méthode **alert()** permet d'afficher dans une boîte toute simple composée d'une fenêtre et d'un bouton OK le texte qu'on lui fournit en paramètre : `alert(nom_de_la_variable);`
`alert('Chaîne de caractères');`
- La méthode **confirm()** est similaire à la méthode alert(), si ce n'est qu'elle permet un choix entre "OK" et "Annuler". Elle renvoie true (bouton OK) ou false (bouton Annuler) :
`var continuer = confirm("Veux-tu quitter ?");`

- La méthode **write()** de l'objet *document* permet de modifier de façon dynamique le contenu d'une page HTML, en écrivant
soit du texte directement (ou une variable) :

```
document.write("bonjour");
var Chaine='bonjour';
document.write('je vous passe le ' + Chaine);
```


soit du contenu HTML :

```
document.write('<font color="#FF0000">Bonjour</font>');
```
- La méthode **prompt()** fournit un moyen simple de récupérer une information provenant de l'utilisateur, on parle alors de boîte d'invite :

```
var saisie = prompt('Posez ici votre question','chaîne par défaut');
```
- Pour **inclure du code javascript** dans du HTML, il suffit de baliser le code avec les balises :

```
<script language="javascript">
Code...
</script>
```

A vous de jouer !

Pour cette série d'exercices, nous ne nous attarderons pas ici à la beauté de la présentation (donc pas de CSS) juste une manipulation du code javascript. Nos pages auront donc globalement la structure suivante pour tester :

```
<html><head><title>Exercice...</title></head>
  <body>
    <script language="javascript">

      // ... code javascript ...

    </script>
  </body>
</html>
```

EXERCICE 1 : du tout simple !

Ecrire un programme qui récupère un nombre entier entre 1 et 10 et recommence tant que la saisie n'est pas correcte. Une fois la saisie correcte, affiche une boîte de dialogue pour rappeler le nb saisi et préciser qu'il est correct.

EXERCICE 2 : encore bien simple !

Compléter votre exercice précédent pour afficher en HTML (dans une table par exemple) la table de multiplication de 1 à 10 du nombre précédemment saisi, du style (exemple avec nb = 2) :

Table de multiplication

x	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20

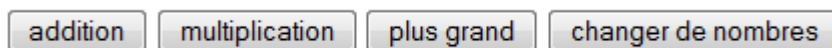
EXERCICE 3 : et avec des fonctions !

1. Mais que fait ce programme ?

```
<script language="javascript">
function modif(c)
{
  document.bgColor=c;
}
</script> </head><body>
<form>
<input type="button" value="mettre en rouge" OnClick="modif('red')">
<input type="button" value="mettre en bleu" OnClick="modif('blue')">
<input type="button" value="mettre en vert" OnClick="modif('green')">
</form></body></html>
```

2. En vous inspirant de ce programme, écrire un programme qui récupère deux nombres et qui propose par le biais de boutons d'en faire la somme ou la multiplication et d'afficher le plus grand. Un autre bouton permettra de changer les valeurs des nombres.

Les événements générés par les boutons et la saisie des deux nombres seront codés par le biais de fonctions.



EXERCICE 4 : mise à jour automatique de champs dans un formulaire

Ci-joint un petit formulaire qui, à partir d'un prix d'achat et d'une quantité, on souhaite que le prix de vente se calcule automatiquement (ce calcul doit se mettre à jour dès qu'on change soit le prix d'achat, soit la quantité grâce à l'évènement `onblur`). **Ecrire la fonction `calculvente` qui met à jour ce calcul.**

Pour récupérer une information du formulaire, on peut utiliser la syntaxe :

```
var nom_var = document.getElementById("nom de l'id du formulaire").elements["name du champs"].value;
```

Pour convertir un champ texte en réel : fonction **`parseFloat`**

```
var nom_var=parseFloat(var_chaine);
```

Pour tester si une valeur est numérique : fonction **`isNaN`** (**`isNaN`** vérifie en fait que ce qui est entré n'est pas un nb)

`isNaN` (valeur) retourne vrai si c'est un nombre, faux sinon

```
<!DOCTYPE html>
<html lang="fr">
<html>
  <head>
    <meta charset="utf-8" />
  </head>
  <form method="POST" id="formulaire">
    <p>Prix <input type="text" name="prix" size="20" onblur="calculvente()"></p>
    <p>Quantité <input type="text" name="qte" value="1" size="20" onblur="calculvente()"></p>
    <p>Vente <input type="text" name="vente" size="20"></p>
    <p><input type="submit" value="Envoyer" name="B1"></p>
  </form>
```

EXERCICE 5 : le grand classique : jeu du plus ou moins

Simuler le jeu de hasard du plus ou moins qui permet de trouver un nb compris entre 1 et 100 (avec des indications « plus » ou « moins » à chaque proposition jusqu'à trouver le bon nombre), nombre choisi aléatoirement par l'ordinateur. Vous utiliserez au maximum des fonctions.

L'utilisateur pourra rejouer autant de fois qu'il le souhaite, on conservera son meilleur score (son nombre de coups minimum pour trouver un nombre).

Pour retourner un nb aléatoire en javascript, vous pouvez utiliser la fonction :

```
function nb_aleatoire(min, max)
{
    var nb = min + (max-min+1)*Math.random();
    return Math.floor(nb);
}
```