



Travaux pratiques développement web

HTML5 et JavaScript : Les éléments de base et Formulaire

Objectifs :

Le but de cette séance de travaux pratiques est de nous familiariser avec JavaScript en tant que langage de programmation à un niveau élémentaire, sans, pour l'instant, nous préoccuper trop de l'interface avec HTML et du modèle orienté objet des documents

Consignes :

- Vous devez rendre à la fin de ce TP un compte rendu avec des imprimés écran montrant toutes les étapes détaillées et validées.

Rappel

Pour qu'un navigateur puisse différencier le code en JavaScript du reste du document HTML, on l'encapsule dans une balise spécifique :

```
<script language="Javascript">  
.....  
</script>
```

Voici un premier exemple de code écrit en Javascript :

```
<html>  
  <head>  
    <title> Voici une page contenant du Javascript</title>  
  </head>  
  <body>  
    <script language="Javascript">  
      alert("Voici un message d alerte!");  
    </script>  
  </body>  
</html>
```

Dans cet exemple, nous appelons la méthode **alert** dans le corps du programme. Modifions l'exemple, pour qu'elle soit appelée dans une fonction :

```

<html>
  <head>
    <title> Voici une page contenant du Javascript</title>
    <script type="text/javascript">
      function myalert() {
        alert("OK, ca marche") ;
        alert("un deuxieme pour la route") ;
      }
    </script>
  </head>
  <body>
    <script language="Javascript">
      myalert();
    </script>
  </body>
</html>

```

Nous constatons ainsi, que du code Javascript peut s'insérer dans l'entête (en général il s'agit des initialisations des variables et des définitions des fonctions) autant que dans le corps de la page (où il s'agira la plupart du temps d'écriture directe sur la page et d'appel de fonctions). Enfin, comme dans la plupart des langages de programmation, nous pouvons utiliser des variables, et passer des arguments aux fonctions. En exemple, notre code encore une fois modifié.

```

<html>
  <head>
    <title> Voici une page contenant du Javascript</title>
    <script type="text/javascript">
      function myalert(text){
        alert(text) ;
        alert("OK..." );}
    </script>
  </head>
  <body>
    <script language="Javascript">
      mytext = "c'est mon texte";
      myalert(mytext);
    </script>
  </body>
</html>

```



Il faut retenir :

- Que le mot clé **var** devant la déclaration est facultatif, mais est fortement recommandé
- Que le nom d'une variable peut commencer soit par une lettre soit par **_**.
- Que l'affectation, ou initialisation se fait avec l'opérateur **=** (à ne pas confondre avec l'opérateur de comparaison **==**).
- Que *Javascript comme le C ou Java*, différencie les minuscules des majuscules.
- Que Javascript utilise un typage faible pour ses variables. De plus il détermine automatique en fonction de l'initialisation le type de ses variables : chaîne de caractères, nombre ou indéterminé (**"undefined"**).

Exercice 1 : Etape par étape

Réalisez les pages ci-dessus.

Etape 1 : En vous aidant de vos connaissances en C et en java, réalisez une page qui affiche 10 boites d'alerte pour les 10 premiers entiers (1,2 ...) dans lequel on affiche si l'entier est pair ou impair.

Etape 2 : Nous avons la possibilité d'écrire directement du code HTML dans la page en utilisant la fonction (prédéfinie) **document.write**, fonction prenant une chaîne de caractère en argument. Par exemple :

```
<html>
  <body>
    <script>
      document.write("Salut les<br>IAC </br> !!!");
    </script>
  </body>
</html>
```

→ **A faire :** Affichez dans une table HTML dix valeurs aléatoires entre 0 et 1 (obtenue grâce à la fonction **Math.random()**) en utilisant **document.write**.

Etape 3 - Portée des variables : Si vous déclarez une variable dans une fonction, elle n'est utilisable que dans cette fonction, cependant, si vous déclarez une variable à l'extérieure, celle-ci est définie et utilisable dans toutes les parties Javascript de votre document.

→ **A faire:** Pour vous en convaincre, déclarer une variable **mon_message**, et initialiser la à votre convenance, dans une première partie javascript (dans l'entête de votre document par exemple), puis afficher son contenu dans le corps du programme.

Etape 4 : Nous utilisons les tableaux en Javascript comme en C. La déclaration et l'utilisation sont les mêmes :

```
<html>
  <body>
    <script>
      value=Array;
      value[0]="1";
      value[1]="1";
      value[2]="2";
      value[3]="3";
      value[4]="5";
      value[5]="8";
      value[6]="13";
    </script>
  </body>
</html>
```

A faire : Faites un tableau (HTML) représentant les différentes combinaisons de couleurs en R,V,B avec les valeurs 00, 33, 66, 99, CC et FF.

Etape 5 : La fonction **prompt** permet de récupérer une valeur entrée par l'utilisateur. Cette fonction s'utilise comme suit :

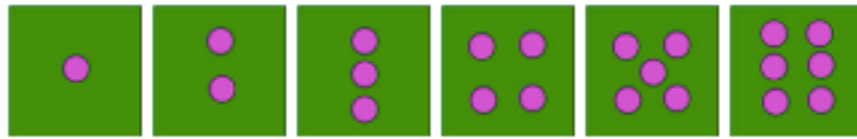
```
<html>
  <body>
    <script>
      valeur = prompt("Entrez une valeur :");
      alert(valeur);
    </script>
  </body>
</html>
```

A faire : Affichez la table de multiplication d'un entier compris entre 0 et 12 rentré

par l'utilisateur via un prompt.

Exercice 2 : Un jeu de dé

- a- A l'aide de tout ce que vous avez vu précédemment, réalisez une page HTML qui affiche le résultat du lancer de deux dés. Ce résultat sera affiché sous la forme d'images des deux dés. Pour ce faire vous pouvez utiliser les images ci-jointes :



- b- Ajoutez à la fin de votre page, un lien vers cette même page ce qui permet un nouveau jet de dés.
- c- Réalisez une application de lancer de 421 qui calcule le nombre de points faits par un lancer de dé

Exercice 3 : Les nombres premiers

Un nombre premier est un nombre naturel qui admet exactement deux diviseurs : 1 et lui-même. Cette définition exclut 1, qui n'a qu'un seul diviseur naturel ; elle exclut aussi 0, qui est divisible par tous les nombres naturels. Nous allons créer une page HTML contenant un script en langage JavaScript qui génère une table des premiers N nombres entiers, où la valeur de N sera déterminée par un morceau de texte de la page elle-même. Les nombres premiers devront être affichés par colonnes alignées à droite, de sorte à obtenir un tableau carré, mais sans dépasser les 20 colonnes. Par exemple, la table des premiers 25 nombres premiers devrait être affichée :

2	13	31	53	73
3	17	37	59	79
5	19	41	61	83
7	23	43	67	89
11	29	47	71	97

1. Créez un fichier HTML primes.html en utilisant le code montré ci dessous comme exemple.

```
<html>
  <head>
    <title>Nombres premiers</title>
  </head>
  <body>
    <h1>Table des premiers <span id="NPRIMES">25</span> nombres premiers</h1>
    <script type="text/javascript">

      // À FAIRE : placez votre code ici

    </script>
    <hr>
    <p>Généré automatiquement par un script.</p>
  </body>
</html>
```

Figure 1: Code proposé

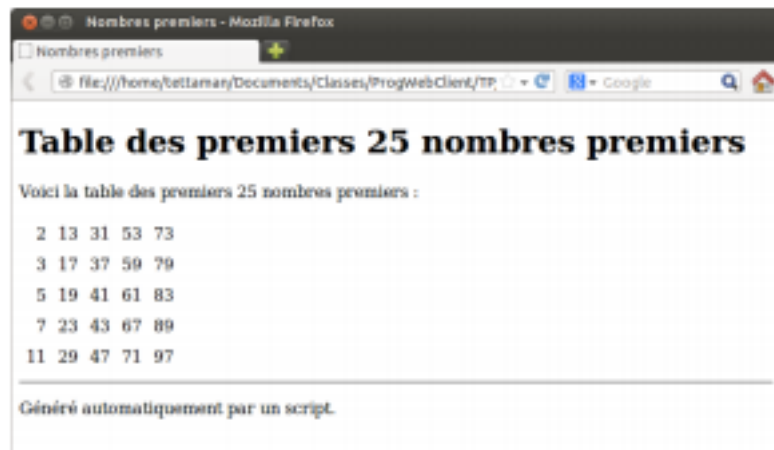


Figure 2: Exemple de rendu

2. Vous pouvez utiliser la propriété **innerHTML** du nœud HTML renvoyé par la méthode **getElementById()** de l'objet document pour lire la valeur de N contenue dans la balise avec **id = NPRIMES** :
N=document.getElementById("NPRIMES").innerHTML;
3. Créez un tableau P de nombres premiers connus et initialisez-le avec le nombre 2 uniquement :
P = new Array();
P[0] = 2;
4. Maintenant, écrivez **une boucle for** qui essaye tous les entiers positifs impairs à partir de 3 et les ajoute au tableau P s'ils sont premiers jusqu'à ce que P contienne exactement N éléments. Évidemment, pour vérifier la primalité d'un nombre i, il faudra essayer de le diviser par P[0], P[1], ...
 S'il se trouve que i est divisible par un des nombres premiers déjà connus, on peut en conclure que i n'est pas premier.
5. Une fois calculé le tableau P des premiers N nombres premiers, il faut s'occuper de leur visualisation. Vous devez écrire du code JavaScript qui ajoute au document HTML (avec la méthode write() de l'objet document) le code HTML d'une table (en utilisant donc les balises avec les caractéristiques requises. Vous devez obtenir une page Web comme celle montrée en Figure 2.

Exercice 4 : Un Jeu de Quiz

L'objectif de cet exercice est de construire un QCM (Questions aux choix multiples) d'auto-évaluation. Votre rendu doit comporter ces éléments :

- a- Un formulaire (ou questionnaire) sous forme de QCM et utilisant les objets du langage JavaScript,
- b- Une procédure de correction dynamique du questionnaire (écrit en JavaScript),
- c- Un corrigé complet du test dans une page HTML.

Pour ce faire, vous devez suivre les étapes suivantes :

Etape 1 : Récupérer les éléments d'un formulaire à travers un code JS

- Proposez un formulaire (écrit avec les balises et les contrôles de HTML 5)
- L'appui sur un bouton doit permettre d'afficher dans une fenêtre alerte les valeurs des champs de saisie

Etape 2 : Le formulaire type QCM

La plupart des questionnaires de type « QCM » utilisent le principe de questions - réponses. Chaque question est suivie d'un certain nombre de réponses, dont une ou plusieurs peuvent être correctes.

A ce niveau, une certaine interactivité (limitée) est proposée à l'utilisateur, elle se matérialise par **trois boutons disposés** au bas du formulaire à cet effet et consiste à :

1. Apporter une correction dynamique du formulaire en lançant la fonction correspondante,
2. Effacer l'ensemble des sélections,
3. Visualiser le corrigé complet du test.

Il existe une différence entre les notions de correction dynamique et de corrigé :

- La correction est dynamique car elle teste la véracité des cases cochées par l'intermédiaire d'un petit algorithme,
- Le corrigé affiche directement la réponse globale sans corriger le test.

a- La correction dynamique.

Elle est réalisée par la fonction "**testqcm(form)**". Cette fonction est appelée par une action sur le bouton "**Correction**", qui lit et teste les valeurs des réponses du formulaire (passé en paramètre). Le résultat des tests est affiché dans une fenêtre externe, et ne donne à l'utilisateur que deux indices par question :

- Soit la réponse à la question est correcte,
- Soit la réponse à la question est incorrecte.

Pour des raisons essentiellement pédagogiques, les résultats affichés sont volontairement vagues. En effet, l'objectif n'est pas de donner rapidement les bonnes réponses quels que soient les choix de l'usager, mais au contraire, de forcer l'utilisateur à rechercher par lui-même le résultat global parfait, en une ou plusieurs itérations du questionnaire.

b- Le corrigé du questionnaire et le calcul du score.

Dans le cas où l'utilisateur ne souhaite pas effectuer ce travail, il lui est toujours possible de pouvoir consulter directement le corrigé du questionnaire en cliquant sur le bouton "**Terminer le test**". Le corrigé du questionnaire n'est autre qu'une simple page HTML contenant du texte et qui s'affiche dans une fenêtre externe. Le score du test est calculé suivant les réponses de l'utilisateur.