

Programmation Web Avancée Côté Client

TP n° 2

Licence Informatique 2ème année
Université de Nice-Sophia Antipolis

Introduction

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.

Nous commencerons donc par développer un script qui génère de façon automatique du contenu pour une page Web « statique » ; dans un deuxième temps, nous ajouterons à cette page quelques éléments de dynamisme.

1 Une table de 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 en Figure 1 comme exemple.
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 `<span>` 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 :

```

<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 – Prototype du fichier HTML à créer.

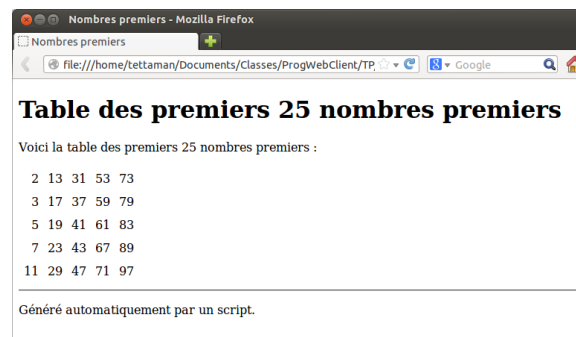


FIGURE 2 – La page Web statique que vous devez obtenir.

```

P = new Array();
P[0] = 2;

```

- 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 $P[0]$, $P[1]$, $\dots$. 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.
- 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 `<table>`, `<tr>` et `<td>`) avec les caractéristiques requises.

Vous devez obtenir une page Web comme celle montrée en Figure 2.

2 Une table dynamique

- Faites une copie du fichier HTML que vous avez créé à l'exercice précédent et changez son nom en `primes-dyna.html`.

```

<html>
  <head>
    <title>Nombres premiers</title>
  </head>
  <body>
    <h1>Table des premiers <span id="NPRIMES">25</span> nombres premiers</h1>
    <button onclick="nextPrime()">Calculer prochain premier</button>
    <script type="text/javascript">

      // Votre code JavaScript...

    </script>
    <p>Voici la table des premiers <span id="N1">xx</span> nombres premiers :</p>
    <table id="PrimeTable" cellpadding="5"></table>
    <script type="text/javascript" >

      // Placez ici le code qui remplit la table...

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

```

FIGURE 3 – Prototype du fichier HTML à créer.

2. Modifiez le fichier de sorte à ce qu'il ressemble au fichier montré en Figure 3 : notamment, vous devez ajouter un bouton juste après la balise `<h1>` et insérer le paragraphe et la balise `<table>` qui, avant, étaient générés automatiquement.
3. Maintenant, vous devez modifier votre code JavaScript pour que, à chaque fois que l'utilisateur appuie sur le bouton, le prochain nombre premier soit calculé et le contenu de la table soit remplacé par une nouvelle table avec un élément en plus. Construisez la nouvelle table en mettant son code HTML (avec les balises `<tr>` and `<td>` et leur contenu) dans une chaîne de caractères `str`, puis remplacez le contenu de la table avec l'instruction

```
document.getElementById("PrimeTable").innerHTML = str;
```

4. Naturellement, lorsque la page est (re)chargée, la table doit s'afficher exactement comme elle s'affichait dans la version statique de la page.