

# TD N°02 JAVASCRIPT

## Exercice 1

Nous allons réaliser une page web capable de faire des calculs. Pour commencer, recopier la page html suivante dans un fichier **exercice1.html**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='utf-8'>
    <title>Exercice 1</title>
  </head>
  <body>

    <form>
      <input type="text" name="nombre">
      <input type="button" value="cube" onClick="cube(this.form)">
    </form>

    <script type="text/javascript">
      cube = function (arg) {
        console.log(arg);
        alert(puissance(parseFloat(arg["nombre"]["value"]), 3));
      }
    </script>
  </body>
</html>
```

On remarque trois balises html nouvelles : **form**, **input** et **script**. **form** permet de réaliser des formulaires ayant des cases à remplir et des boutons. **input** désigne des éléments attendant une entrée de l'utilisateur. Ici, les deux éléments **input** sont une case de texte et un bouton. **script** permet de mettre directement dans le code html un script (ici écrit dans le langage javascript).

Il y a plusieurs choses à remarquer :

- la case de texte a un attribut **name** qui définit à peu près le nom de variable qu'aura son contenu.

- b. le bouton a un attribut **onClick** définissant le comportement à adopter quand on clique sur le bouton. Ici, on fait appel à la fonction **cube** avec pour argument **this.form** c'est-à-dire le contenu du formulaire.
- c. Dans la fonction **cube**, le paramètre formel est nommé **arg**. Il s'agit d'un objet de type formulaire. On peut accéder à la valeur d'une variable **x** contenue dans le formulaire en écrivant **arg["x"]["value"]**. Attention, elle est de type **string**. Nous l'avons donc ici **convertie explicitement** en nombre à l'aide la fonction **parseFloat** (équivalent javascript de la fonction **int** de python (ou **tonumber** de lua)).
- d. **alert** est une *procédure* créant une fenêtre de dialogue contenant le texte passé en argument.
- e. **console.log** est une *procédure* écrivant dans la console du navigateur.  
Ouvrons maintenant cette page web avec un navigateur (firefox ou chrome). Quand on clique sur le bouton, rien ne se passe car le code n'est pas complet. Pour voir l'erreur, cliquer avec le bouton droit sur la page puis sélectionner l'onglet *Console* ou *Web Console*. Il doit apparaître un texte signalant une *ReferenceError* en lien avec *puissance*. Il doit aussi apparaître le résultat de la procédure **console.log** qui est un dictionnaire représentant un élément html.

## Questions

1. Il manque dans notre script la fonction **puissance(x, n)** qui calcule la puissance  $n$ -ième d'un nombre  $x$ . L'implémenter à l'intérieur de la balise **script** déjà existante, recharger la page et tester.
2. Ajouter un bouton et le code nécessaire pour calculer le carré du nombre donné.
3. Ajouter un bouton et le code nécessaire pour calculer la racine carrée du nombre donnée à 0.001 près.
4. Ajouter un bouton et le code nécessaire pour tester si le nombre est premier.

## Exercice 2

Notre page html commence à contenir plus de javascript que de html et devient de moins en moins lisible. Nous allons donc séparer le script du fichier html (comme nous séparons le style du contenu dans un fichier css). Pour cela, nous allons mettre le contenu de la balise **script** dans un fichier **exercice2.js** et y faire référence dans le fichier html:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='utf-8'>
    <title>Exercice 1</title>
    <script type="text/javascript" src="exercice2.js"></script>
  </head>
  <body>
    <form>
      <input type="text" name="nombre">
      <input type="button" value="cube" onClick="cube(this.form)">
    </form>
  </body>
</html>
```

## Questions

1. Réaliser les manipulations et tester la page obtenue.
2. Il est possible de modifier le contenu d'un champ de texte : `arg["x"]["value"]` se comporte comme une variable, on peut donc lui affecter une valeur ! Ajouter un élément de texte dans la page html qui contiendra le résultat du dernier calcul effectué (en profiter pour enlever les `alert` du code).
3. Ajouter le nécessaire pour, étant donnés  $x$  et  $n$ , afficher  $x^n$ .