

Université Grenoble Alpes - IUT2 - Département informatique
Module 4103C

TP 1 : Premiers pas en JavaScript (2H)

Objectifs du TP :

- Comprendre l'environnement d'exécution d'un programme JavaScript
- Se familiariser avec la syntaxe du langage JavaScript
- Apprendre à utiliser Le débogueur

Ce document est consultable sur l'intranet du département.

Le langage JavaScript

Lorsqu'il est utilisé dans le cadre d'une **application web**, le code JavaScript est interprété par l'interpréteur JavaScript qui est intégré au navigateur. Nous allons travailler avec **Firefox** et son interpréteur JavaScript (**SpiderMonkey**).

Les sites suivants : <https://developer.mozilla.org/fr/docs/JavaScript/Reference> et <http://www.w3schools.com/jsref/> peuvent vous être utiles dans l'apprentissage de ce langage.

0. Mise en place de l'environnement de travail

1. Lancer **Firefox**
2. Ouvrez le menu de Firefox (icône en haut à droite) et choisissez **Developer** puis **Console**.
3. Se placer sur la ligne du bas de la console et taper la commande `alert("Attention !");`. Validez par un retour chariot. Une boîte de dialogue apparaît sur la page web affichée dans votre navigateur. Testez d'autres instructions JavaScript
4. Lancez gedit et créez la page HTML `test.html` minimale suivante :

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
<h1>test</h1>
<script src='test.js'></script>
</body>
</html>
```

5. Créez le fichier `test.js` contenant la commande `alert("Attention!");`.
6. Chargez la page `test.html` dans votre navigateur et observez le résultat.
7. Modifiez `test.js` pour qu'il contienne le code suivant :

```
function mafonction()
{
    i=i*10;
    return (i+j);
}
var i,j;
i=prompt("entrez une valeur");
i=parseInt(i);
j=parseInt(prompt("entrez une valeur"));
r=mafonction();
alert("resultat:"+r);
```

8. Rechargez la page (pensez à forcer le rechargement en combinant [CTRL + F5](#)) pour faire exécuter le nouveau script.
9. Ajouter une erreur dans votre code (supprimez par exemple une parenthèse) et visualisez le message d'erreur affiché dans la console (onglet **Console**). Pensez à bien tout faire afficher (en réglant le filtre). Corrigez le bug dans `test.js` et relancez le chargement de la page `test.html` dans votre navigateur
10. Visualisez votre script grâce à l'onglet **Debugger**. Un débogueur permet d'exécuter pas à pas votre code et de visualiser à tout instant l'état de vos variables. Essayez de l'utiliser en vous appuyant sur le document suivant <https://developer.mozilla.org/en-US/docs/Tools/Debugger>

Vous devez notamment apprendre à :

- Créer des points d'arrêt dans votre code (il suffit de cliquer sur le numéro de la ligne)
- Exécuter votre code pas à pas (avec Step Over)
- Descendre et remonter dans les fonctions appelées (ici `mafonction`) avec Step In et Step Out

11. A l'aide du débogueur (faites exécuter pas à pas et/ou observez la valeur des différentes variables affichées dans la fenêtre de droite), répondez aux questions suivantes :
- Quelle est la valeur de la variable globale `i` avant l'interprétation de la première instruction ?
 - Quel est le type de la variable `i` juste après l'instruction `prompt` ?
 - Quelle est la valeur de la variable globale `i` juste après l'instruction `i=i*10` dans la fonction `mafonction` ?
 - Quelle est la version de votre navigateur ?
 - Quelles sont les dimensions de la zone d'affichage de la fenêtre du navigateur ?
12. Modifiez le programme `test.js` pour que le résultat soit affiché :
- dans la console
 - dans la page web comme contenu de l'élément `h1` en remplacement de "test" (en recherchant, avec le débogueur, dans l'objet `document`, l'attribut contenant "test")

Voilà vous êtes fin prêt pour développer vos premiers programmes en JavaScript.

1. Petits programmes indépendants pour se familiariser avec la syntaxe JavaScript

1. Ecrire un programme `p1.js` (et son `p1.html` associé) qui demande à l'utilisateur de saisir plusieurs valeurs par l'intermédiaire de boîtes de dialogue (`prompt`) s'affichant successivement (donc une saisie par boîte et on terminera la saisie par "-1") et qui affiche la somme des nombres saisis et leur moyenne ainsi que le nombre de 0 dans une nouvelle boîte de dialogue (`alert`).
2. Aller voir la documentation de la fonction [parseInt\(\)](#) et modifier votre programme pour faire en sorte d'ignorer si l'utilisateur saisit une chaîne de caractères (somme et moyenne ne doivent se baser que sur les valeurs numériques saisies)
3. Ecrire un programme `p2.js` (et son `p2.html` associé) réalisant le même traitement mais stockant cette fois-ci les valeurs saisies dans un tableau et appelant les fonctions `somme`, `moyenne` et `nbzeros` en passant le tableau en paramètre
4. Pour expérimenter un peu les aspects objets de JavaScript, en reprenant votre cours et en lisant le document suivant [Mozilla détail](#)

objets JS écrire un programme `chat.js` (et son `chat.html` associé) dans lequel on crée plusieurs objets "chat" ayant un nom, une couleur, un age. Donnez à ces chats la méthode prototypique `miauler` qui affichera "miaou !". Créez un nouvel objet chat et associez lui une méthode `miauler` affichant "uoaim !". Faites miauler l'ensemble des chats et vérifiez que tous sauf un miaulent convenablement (Sachant que, si plusieurs `console.log()` affichent le même message à la suite, Firefox va agréger les messages dans la console). Visualisez les structures de données avec le débogueur (fenêtre de droite). Comment se nomme la propriété référençant le prototype ?

2. Ecriture dans une page web avec "document.write"

1. On propose maintenant d'afficher les résultats de nos calculs dans la page web (donc en HTML). On se contentera dans un premier temps d'utiliser la méthode `write` de l'objet `document`. Il suffit donc de construire une chaîne de caractères `S` correspondant à un code HTML (valide de préférence), puis ensuite de faire l'appel à `document.write(S)`. Un exemple simple déjà traité en PHP, consiste à construire la table de multiplication entre 1 et 9 (donc de 9 lignes et 9 colonnes). Créez cette table en regroupant tout (`html`, `js`, `css`) dans le fichier `table.html`. Ajoutez ensuite la possibilité pour l'utilisateur de préciser le nombre de lignes et de colonnes par une fenêtre de dialogue (prompt)
2. Vérifiez ensuite le code HTML généré. Attention, vous ne pourrez pas voir le code généré en affichant le code source (on voit le script et pas le résultat de son interprétation). On peut par exemple utiliser l'onglet **Inspecteur** pour répondre à la question suivante : Où le code HTML généré a-t-il été inséré dans la page ?
3. Placez `table.html` dans votre `public-html` et vérifiez le bon fonctionnement du programme

3. Question de synthèse

Résumez les différentes étapes d'exécution du programme JavaScript précédent (table de multiplication) en partant de la saisie de l'url de `table.html` dans le navigateur. Rédigez un petit texte en décrivant à chaque étape les opérations effectuées, les programmes impliqués dans ces opérations et les éventuels protocoles utilisés pour les communications.

Demandez à votre enseignant de relire votre texte.

4. S'il vous reste du temps

> Créer une mini-WebExtension pour Firefox et Chrome grâce au tuto :
[MDN - Votre première WebExtension \(FR\)](#)

> Sur <http://www.w3schools.com/js/default.asp> :

1. Explorez les parties JS Tutorial (allez voir ce qui est en rapport avec les évènements, vous en aurez besoin pour le prochain TP), JS Objects et JS Functions
2. Faites le quiz suivant : http://www.w3schools.com/js/js_quiz.asp