

# JavaScript asynchrone et services web

Licence Informatique 3ème année

Alexandre Niveau — Jean-Marc Lecarpentier

## JavaScript asynchrone et services web

### Notes de cours

#### 📖 JavaScript (rappels / pour référence)

- Introduction à JavaScript
  - Utilité d'un langage de script côté client
  - Introduction à JavaScript
- Modèle objet de JavaScript
- Modifier le DOM avec JavaScript
  - Présentation du DOM
  - Manipulation du DOM avec JavaScript
- Programmation événementielle avec le DOM
- JS asynchrone / Ajax

#### 📖 Services web

- Introduction aux services web
  - SOAP et WSDL
  - Architecture REST
- Ajax et services web
  - Principes de la *same-origin policy*
  - Utilisation de JSONP
  - Utilisation de CORS

### Travail personnel

## Objectifs

Ce TP vous fait expérimenter avec XHR et implémenter un petit service web.

### Exercice 1 — Récupération de couleurs

#

On met à votre disposition un [service JSON](#). À chaque appel, il renvoie un tableau contenant dix objets représentant des couleurs. Chaque couleur a un nom (attribut name) et un code hexadécimal (attribut hex). (Les noms et les codes hexa sont tirés aléatoirement à chaque appel.)

Télécharger [l'archive de l'exercice](#) : elle contient un fichier couleurs.html et un fichier couleurs.js vide. **On ne modifiera que le fichier JS.**

**Attention** : si vos modifications du fichier JS ne semblent avoir aucun effet, c'est sûrement parce qu'il a été mis en cache par le navigateur : **faites un hard reload** (ctrl-shift-R).

1. Faire en sorte que chaque clic sur le bouton fasse appel au service et ajoute dix nouvelles couleurs dans la liste. Chaque item ajouté doit contenir le nom de la couleur et avoir pour couleur de fond la couleur elle-même, comme sur l'image suivante.



2. Certaines couleurs sont trop sombres pour que le nom soit lisible. Faire en sorte qu'un clic sur une couleur affiche son nom en blanc, en utilisant la classe CSS prévue à cet effet. Un deuxième clic doit à nouveau retirer la classe.
3. **Optionnel** (sort largement du cadre de l'exercice) : remplacer le clic sur un bouton par de l'*infinite scrolling*. Attention, l'*infinite scrolling* est [rarement une solution adaptée d'un point de vue ergonomique](#).

### Exercice 2 — Codes postaux

#

L'objectif de cet exercice est d'implémenter et d'utiliser un service web très simple, qui indique tous les noms de communes françaises correspondant à un code postal donné.

Pour gagner du temps de mise en place, au début on utilisera [le fichier PHP contenu dans cette archive](#), qui déclare une variable contenant un tableau avec toutes les communes associées à tous les code postaux du Calvados sous la forme de tableaux associatifs de la forme suivante :

```
[ 'cp' => '14630', 'ville' => 'CAGNY' ]
```

Dans la suite de l'exercice on utilisera la base de données complète.

1. Implémenter le service web demandé : écrire un script PHP qui prend en paramètre d'URL **un code postal**, et qui renvoie au format JSON une **liste des noms de communes** correspondant au code postal donné.

Par exemple, le code 14630 correspond à trois communes : Cagny, Émiéville et Frénuville. Supposons que le service se trouve à l'URL `http://example.com/service.php` et que le paramètre soit `cp` (mais vous êtes libres de choisir autrement). Un appel à `http://example.com/service.php?cp=14630` doit donc renvoyer le JSON suivant :

```
[ "Cagny", "Émiéville", "Frénuville" ]
```

**NB: votre script ne doit pas renvoyer une page HTML.** Lire l'énoncé soigneusement.

2. Créer une page HTML contenant un formulaire avec un champ texte destiné à recevoir un code postal, et un bouton. En utilisant votre service web, faire en sorte que lors du clic sur le bouton, un menu déroulant (`select`) apparaisse (sans changement de page), contenant toutes les communes correspondant au code postal saisi.

**NB: vous ne devez pas faire un script PHP dans cette question.** On veut une page qui fait des appels à votre service depuis un script côté client. Lire l'énoncé soigneusement.

3. Faire en sorte que votre service web puisse être utilisé depuis n'importe quel domaine. Tester en ouvrant votre HTML « en local » avec votre navigateur (pseudo-protocole `file://`), et demandez à votre voisin·e de le tester depuis sa page HTML sur son serveur. Tout ça doit fonctionner !
4. Modifier le PHP pour utiliser [cette base de données](#), contenant toutes les communes de France avec leur code postal. Seuls les deux premiers champs nous intéressent ici.
5. **Optionnel** : Implémenter une autocomplétion du champ « code postal », qui propose à l'internaute des complétions possibles du code postal au fur et à mesure de sa frappe. Cela nécessite une modification du service web et du script côté client. Pour simplifier, afficher simplement la liste comme du texte sur

la page. (On pourra ensuite essayer d'utiliser un élément [datalist](#) à associer à l'input.).