

# TD1 – Boucles et conditionnelles en Javascript

## Introduction

L'objectif de ce TD est de manipuler les concepts algorithmique de base avec Javascript.

Vous implémenterez les défis proposés dans le cours sur les [base du langage javascript](#).

## Environnement de développement

Vous aurez besoin :

- d'un éditeur de code pour HTML, CSS et Javascript : `gedit`, `sublime-text`, `atom`.
- de la **console Javascript** du navigateur pour afficher :
  - les messages d'erreur
  - et la sortie de la fonction `console.log()`raccourcis pour accéder à la console :
  - Sous Firefox : Ctrl-Maj-K
  - Sous Chrome : Ctrl-Maj-J
- pour garder une arborescence organisée, créez un dossier `L2/DAI/TD1/`
- pour vous faciliter la mise en route, vous pourrez partir de ce modèle : [mapage.xhtml](#) que vous modifierez à loisir après l'avoir téléchargé dans votre dossier et renommé.

## Exercice 1 - Boucle for

### Préparation

Copiez les fichiers [mapage.xhtml](#) et [dixchiffres\\_for.js](#) dans le dossier `L2/DAI/TD1/`.

Renommez le fichier html en `exo1.xhtml` et connectez-lui le fichier javascript.

Au chargement de la page, vous devez voir apparaître une fenêtre avec les 10 chiffres.

### Modficiation du code

Modifiez le code afin de supprimer la virgule finale.

## Exercice 2 - Boucle while

Dans des fichiers `exo2.xhtml` et `dixchiffres_while.js`, modifiez le code suivant :

JavaScript

```
let message = '';
let i = 0;
while (i<10) {
    message = message + i + ', ';
    i = i+1;
}
alert(message);
```

afin qu’il affiche les 10 chiffres par ordre décroissant de 9 à 0, et sans virgule finale !

## Exercice 3 - Boucle do ... while

Dans des fichiers `exo3.xhtml` et `controle_saisie.js`, transformez le code suivant :

JavaScript

```
let nombre;
do {
    nombre = prompt('Entrez un nombre supérieur à 10');
} while (nombre <= 10);
alert(nombre + ' est bien supérieur à 10 !');
```

afin de s'assurer de la saisie d’un nombre **entier** et **impair** compris dans l’intervalle **[10; 20]**.

**Indice** : utilisez la fonction `parseInt` et l’opérateur `%` (modulo).

Testez avec les valeurs : 12, 8, 2.5, toto et 15 par exemple.

## Exercice 4 - Affichage des erreurs

Dupliquez les fichiers de l'exercice précédent et renommez-les en `exo4.xhtml` et `controle_saisie_v2.js`.

Il s'agit maintenant de donner à l'utilisateur le nombre et le détail des erreurs commises dans sa saisie.

Vous imitez le plus fidèlement possible les messages affichés sur la [variante 2 du cours](#).