LES FONCTIONS

**Créer sa première fonction**

On ne va pas y aller par quatre chemins, voici comment écrire une fonction :

*function* myFunction(*arguments*) {

// Le code que la fonction va devoir exécuter

}

Décortiquons un peu tout ça et analysons un peu ce que nous pouvons lire dans ce code :

* Le mot-clé **function** est présent à chaque déclaration de fonction. C'est lui qui permet de dire « Voilà, j'écris ici une fonction ! » ;
* Vient ensuite le nom de votre fonction, ici **myFunction** ;
* S'ensuit un couple de parenthèses contenant ce que l'on appelle des **arguments**. Ces arguments servent à fournir des informations à la fonction lors de son exécution. Par exemple, avec la fonction **alert()** quand vous lui passez en paramètre ce que vous voulez afficher à l'écran ;
* Et vient enfin un couple d'accolades contenant le code que votre fonction devra exécuter.

Il est important de préciser que tout code écrit dans une fonction ne s'exécutera que si vous *appelez* cette dernière (« appeler une fonction » signifie « exécuter »). Sans ça, le code qu'elle contient ne s'exécutera jamais.

Bien entendu, tout comme les variables, les noms de fonctions sont limités aux caractères alphanumériques (dont les chiffres) et aux deux caractères suivants : **\_** et **$**.

Comme nous le disions plus haut, l'intérêt d'une fonction réside notamment dans le fait de ne pas avoir à réécrire plusieurs fois le même code.

**Toute variable déclarée dans une fonction n'est utilisable que dans cette même fonction** ! Ces variables spécifiques à une seule fonction ont un nom : les **variables locales**.

Lorsqu'une variable n'est accessible que dans une partie de votre code, on dit qu'elle se trouve au sein d'un « scope ». Retenez bien ce terme, il vous servira à l'avenir.

Les arguments et les valeurs de retour

Maintenant que vous connaissez le concept de la portée des variables, nous allons pouvoir aborder les arguments et les valeurs de retour. Ils permettent de faire communiquer vos fonctions avec le reste de votre code. Ainsi, les arguments permettent d'envoyer des informations à votre fonction tandis que les valeurs de retour représentent tout ce qui est retourné par votre fonction une fois que celle-ci a fini de travailler.

**En résumé**

* Il existe des fonctions natives, mais il est aussi possible d'en créer, avec le mot-clé function.
* Les variables déclarées avec var au sein d'une fonction ne sont accessibles que dans cette fonction.
* Il faut éviter le plus possible d'avoir recours aux variables globales.
* Une fonction peut recevoir un nombre défini ou indéfini de paramètres. Elle peut aussi retourner une valeur ou ne rien retourner du tout.
* Des fonctions qui ne portent pas de nom sont des fonctions anonymes et servent à isoler une partie du code.