

JAVASCRIPT – LES FONCTIONS

Lorsque le code devient trop long et répétitif, il est possible de l'organiser en petites unités que l'on appelle des fonctions, qui vont s'occuper d'une tâche précise.

L'avantage des fonctions :

- Raccourcir le code en évitant de répéter des lignes de code identiques
- Appeler une fonction suite à une action de l'utilisateur (exemple clique sur un bouton et ça appelle une fonction)

Exemple d'une fonction :

```
function sayHello(nom,prenom){  
  const hello = `Bonjour ${prenom} ${nom}`;  
  return hello;  
}  
  
const result = sayHello("Vilport","Cécile");  
console.log(result);
```

Cette fonction a pour but de retourner une chaîne de caractère pour dire bonjour.

function signifie que l'on va créer une fonction

sayHello est le nom qu'on lui donne

(nom,prenom) ce sont les **arguments**, ils permettent de spécifier que pour utiliser notre fonction, il faudra lui apporter 2 arguments, 2 données

return hello signifie que notre fonction va répondre quelque chose. Pour résumé, on va lui apporter un nom et un prénom et

elle va nous répondre / retourner « Bonjour prenom nom »

LES ARGUMENTS

Les arguments sont optionnels, il peut arriver qu'une fonction n'ait pas besoin de données en entrée. Dans ce cas, la parenthèse sera bien là mais vide.

Il peut y avoir autant d'arguments que nécessaire.

LE RETOUR

Le « return » est également optionnel, parfois une fonction ne répond rien, elle fait ce qu'on lui demande et rien de plus. Le « return » permet aussi de stopper la fonction, si un bout de code est présent après un return, il ne sera pas exécuté.

REUTILISONS NOTRE FONCTION

Comme je vous le disais, l'avantage des fonctions est de pouvoir écrire moins de code et d'être réutilisable. Alors, nous pouvons maintenant dire bonjour à d'autres personnes :

```
function sayHello(nom,prenom){  
    const hello = `Bonjour ${prenom} ${nom}`;  
    return hello;  
}  
  
console.log(sayHello("Vilport","Cécile"));  
console.log(sayHello("Hugo","Victor"));  
console.log(sayHello("Zola","Emile"));  
console.log(sayHello("De Niro","Robert"));
```

Ce qui donnera en console :

```
Bonjour Cécile Vilport  
Bonjour Victor Hugo  
Bonjour Emile Zola  
Bonjour Robert De Niro
```

VALEURS PAR DEFAUT

Il est possible de spécifier des valeurs par défaut pour nos arguments, ainsi si un utilisateur oublie ou ne connaît pas un des arguments demandé alors une valeur par défaut lui sera attribué. De ce fait, ceci permet également de rendre cet argument optionnel :

```
function sayHello(nom,prenom = "M/Me"){
  const hello = `Bonjour ${prenom} ${nom}`;
  return hello;
}

console.log(sayHello("Vilport"));
console.log(sayHello("Hugo","Victor"));
console.log(sayHello("Zola"));
console.log(sayHello("De Niro","Robert"));
```

Il suffit de mettre « = une_valeur » à côté de l'argument optionnel.

Voici ce que cela donne en console :

```
Bonjour M/Me Vilport
Bonjour Victor Hugo
Bonjour M/Me Zola
Bonjour Robert De Niro
```