

Les fonctions, déclaration et nommage, variable locale et externe, paramètres d'une fonction et valeur par défaut, valeur de retour, ...

Déclaration et nommage

Une fonction est une procédure JavaScript, un ensemble d'instructions effectuant une tâche ou calculant une valeur.

Il existe une syntaxe spécifique pour la déclaration des fonctions (vous pouvez consulter la page de l'instruction [function](#) pour plus de détails) :

```
function nom(param1,param2,...){  
    instructions  
}
```

nom

Le nom de la fonction.

param1,param2

Le nom d'un argument à passer à la fonction (une fonction pouvant avoir jusqu'à 255 arguments).

instructions

Les instructions qui forment le corps de la fonction.

Variable locale et externe(globale)

▷ Les variables déclarées hors de tout bloc sont appelées globales et utilisables dans tout le fichier (à éviter)

▷ Une variable locale masque obligatoirement une variable de

Même nom située dans une portée plus englobante

```
let a=5, b=12;  
function main(void)  
{  
    let a=3, i=0;  
    for(i=0; i<10; i=i+1){  
        let a=4;//A EVITER  
    }  
}
```

Les paramètres d'une fonction et valeur par défaut

Grâce aux paramètres par défaut qui existent depuis ECMAScript 2015 (ES6), on peut se passer de cette vérification et alléger le code de la fonction :

```
function multiplier(a, b = 1) {  
    return a * b;  
}  
multiplier(5, 2); // 10  
multiplier(5, 1); // 5  
multiplier(5, undefined); // 5  
multiplier(5); // 5
```

Les valeurs de retour d'une fonction

Les valeurs de retour sont, comme l'indique, les valeurs retournées par une fonction après son exécution.

L'instruction `<<return>>` met fin à l'exécution d'une fonction et définit une valeur à renvoyer à la fonction appelante.

```
function sum( a, b){  
  
    return a+b;  
}
```

Certaines fonctions n'ont pas de valeur de retour, ils envoient undefined