

## Fonctions

Quand dans un programme on est amené à répéter la même séquence d'instructions un grand nombre de fois, on peut souvent délimiter le bloc contenant les instructions et en faire une fonction : un mini programme que l'on pourra appeler autant de fois qu'on voudra dans le programme principal.

### Python

```
def nomDeFonction(paramètres1,para2,...) :  
    instruction1  
    instruction2  
    ...  
    return valeur(s)
```

### JavaScript

```
function nomDeFonction(para1,para2,...) {  
    instruction1 ;  
    instruction2 ;  
    ...  
}
```

### Exercice :

On a déjà fait un programme (en contrôle) affichant une table de multiplication :

```
1 n=int(input("nombre dont on veut afficher la table"))  
2 for i in range(1,11) :  
3     print(i, "x",n, "=",n*i)
```

On se propose de créer la fonction table(n,max) qui affichera la table de n, où ce nombre sera multiplié par 1, 2, ... jusqu'à max. Puis on l'intégrera dans un programme plus complet qui posera à l'utilisateur la question des valeurs de n et de max

### Python

```
def table(n,max):  
    for i in range(1,max) :  
        print(i, "x",n, "=",n*i)  
  
nbr=int(input("table de n="))  
maximum=int(input("la table ira de 1 à max="))  
table(nbr,maximum)
```

### JavaScript

```
function table(n,max){  
    for (var i=1;i<=max;i++) {  
        document.write(i+"x"+n+"="+i*n+"<br>")  
    }  
}  
var nbr=parseInt(prompt("table de n="))  
var maximum=parseInt(prompt("valeur maximale max="))  
table(nbr,maximum)
```

## Mini projet : TIC TAC TOE

Objectif : programmer le jeu tic-tac-toe / le morpion en utilisant un langage choisi.

Le jeu pourra être fait entièrement en interface textuelle ou en utilisant une librairie graphique.

Langage : Python ou JavaScript (et sa librairie P5)

Outils principaux : fonctions et listes de listes

Listes de listes :

Attention si on tape Grille=[[0,0,0],[0,0,0],[0,0,0]] puis Grille[1][2]=3 et enfin Grille alors on obtient :

- ▶ 0: (3) [0, 0, 0] ou [[0, 0, 0], [0, 0, 3], [0, 0, 0]] en python
- ▶ 1: (3) [0, 0, 3] donc pour accéder à l'élément colonne i, ligne j faut il taper Grille[i][j] ou Grille[j][i]
- ▶ 2: (3) [0, 0, 0]

Les fonctions que l'on va devoir créer :

**jouabilite(grille,i,j)** qui déterminera si la case colonne i, ligne j est jouable. Il renverra un booléen (true ou false)

**joue(grille,i,j,player)** : qui remplit la case

**testeHoriz(grille,player)** : qui déterminera si le joueur « player » a réussi à remplir une ligne horizontale

**testeVert(grille,player)** : qui déterminera si le joueur « player » a réussi à remplir une ligne verticale

**testeOblique(grille,player)** : qui déterminera si le joueur « player » a réussi à remplir une ligne oblique

**testeGagne(grille,player)** : qui déterminera si le joueur « player » a réussi à aligner trois pions et donc à gagner (cette fonction utilisera les trois fonctions précédentes).

**affiche(grille)** qui affichera la grille et les actions des joueurs

généralisations possibles : vous pourrez vous inspirer de ce travail pour créer un puissance 4