

1 / préparation

Copiez le répertoire « [Python-TP FONCTIONS](http://ressource.elec.free.fr/softs.html) » qui se trouve sur <http://ressource.elec.free.fr/softs.html> dans votre répertoire habituel de travail.

2/ testons ce que nous avons importé.

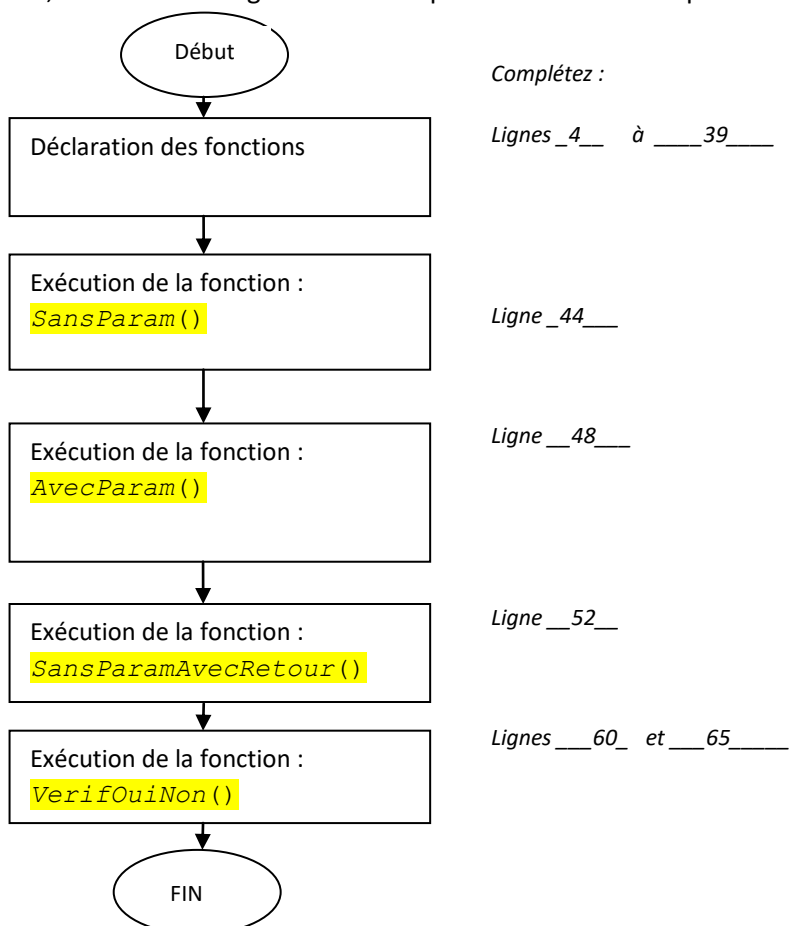
Ouvrez le fichier *fonction.py* dans l'éditeur.

Ce programme contient quelques exemples de ce que nous avons présenté dans le cours.

-> testez tout d'abord le programme pour vérifier que l'importation s'est bien passée.

3/ analysons et modifions le programme

D'abord, identifions les lignes de codes qui concernent les étapes de cet organigramme :



Ensuite, analysons chacune des fonctions :

SansParam()**Analyse de l'appel de la fonction (ligne 43)**

Lors de cet appel de fonction, un paramètre est-il envoyé ? non

Si oui, lequel et quel est son type (entier, réel, caractère...)? _____

Analyse du rôle de la fonction (ligne 4 à 8)

Cette fonction reçoit-elle un paramètre ? non

Si oui, lequel et quel est son type (entier, réel, caractère...)? _____

Cette fonction renvoie-t-elle un paramètre ? non

Si oui, lequel et quel est son type (entier, réel, caractère...)? _____

AvecParam()**Analyse de l'appel de la fonction (ligne 46)**

Lors de cet appel de fonction, un paramètre est-il envoyé ? oui

Si oui, lequel et quel est son type (entier, réel, caractère...)? _string (chaîne de caractère)

Analyse du rôle de la fonction (ligne 12 à 18)

Cette fonction reçoit-elle un paramètre ? oui

Si oui, lequel et quel est son type (entier, réel, caractère...)? _string

Cette fonction renvoie-t-elle un paramètre ? non

Si oui, lequel et quel est son type (entier, réel, caractère...)? _____

SansParamAvecRetour() ;**Analyse de l'appel de la fonction (ligne 43)**

Lors de cet appel de fonction, un paramètre est-il envoyé ? non

Si oui, lequel et quel est son type (entier, réel, caractère...)? _____

Cette fonction va-t-elle renvoyer un paramètre (si oui, donner son type) ? oui, int

La variable **a** va-t-elle prendre pour état la valeur retournée par la fonction ? oui

Analyse du rôle de la fonction (ligne 23 à 28)

Cette fonction reçoit-elle un paramètre ? non

Si oui, lequel et quel est son type (entier, réel, caractère...)? _____

Cette fonction renvoie-t-elle un paramètre ? oui

Si oui, lequel et quel est son type (entier, réel, caractère...)? int

Cette fonction appelle elle-même une fonction prédéfinie. Laquelle ? print()_et random.randint()

VerifOuiNon() ; // C'est celle que nous avons vu lors du cours**Analyse de l'appel de la fonction (ligne 59)**

Lors de cet appel de fonction, un paramètre est-il envoyé ? oui

Si oui, lequel et quel est son type (entier, réel, caractère...)? _string

Cette fonction va-t-elle renvoyer un paramètre (si oui, donner son type) ? _string

Analyse du rôle de la fonction (ligne 32 à 36)

Cette fonction reçoit-elle un paramètre ? oui

Si oui, lequel et quel est son type (entier, réel, caractère...)? _string

Cette fonction renvoie-t-elle un paramètre ? oui

Si oui, lequel et quel est son type (entier, réel, caractère...)? _string

Cette fonction appelle elle-même deux fonctions prédéfinies. Lesquelle ? print() et input()

Maintenant que nous avons vu à l'œuvre et analysé des fonctions, nous allons modifier un programme en créant nos fonctions :

➔ Ouvrez dans l'éditeur *fonction2.py* .

Ce petit programme affiche quelques annonces de vols internationaux.

On s'aperçoit tout d'abord que les lignes ci-dessous sont écrites plusieurs fois à l'identique :

```
print()
print("-----")
print()
```

➔ Déclarer et appelez une fonction (qu'on nommera par exemple *tirerUnTrait*) pour modifier le programme à son avantage. (c'est-à-dire que les trois lignes ci-dessous n'apparaissent qu'une seule fois dans tout le code)

Le programme maintenant transformé est composé de trois blocs qui annoncent chacun l'horaire d'un vol. Nous allons aller plus loin dans l'organisation de ce programme et écrire une fonction *annoncerUnVol*, qu'il suffira d'appeler trois fois dans le programme principal.

Contrairement à l'exemple de la fonction *tirerUnTrait*, ces trois blocs ne sont pas absolument identiques: la destination et l'horaire du vol diffèrent d'un cas à l'autre. Il va donc falloir envoyer des paramètres à cette nouvelle fonction !

On nommera *destination* la destination et *horaire* l'horaire de vol.

➔ Modifier le programme pour qu'il fasse la même chose en appelant 3 fois une fonction *annoncerUnVol()* de cette manière :

```
annoncerUnVol("Tokyo","9h00");
annoncerUnVol("Sydney","9h30");
annoncerUnVol("Toulouse","9h45");
```

Retour à notre « Photoshop » :

➔ Modifier votre programme *Image.py* pour proposer le choix de la modification à l'utilisateur ; chaque modification devra alors faire l'objet d'une fonction appelée :

```
AugmentationLuminosité( )
Invert( )
EliminationVert( )
....
```