

1°) Comprendre : « Pour avoir du style, il faut éviter les redites »**Analyse d'un programme (11_1a.py)**

```
print("Le vol en direction de Tokyo décollera à 9h00 ")
print("-----")
print("Le vol en direction de Sydney décollera à 9h30 ")
print("-----")
print("Le vol en direction de Toulouse décollera à 9h45 ")
print("-----")
```



John McCarthy (1927–2011)

1^{ère} amélioration : utilisation d'une première fonction sans argument (11_1a_avecfonction)

```
def tireruntrait() :
    print("-----")

print("Le vol en direction de Tokyo décollera à 9h00 ")
tireruntrait()
print("Le vol en direction de Sydney décollera à 9h30 ")
tireruntrait()
print("Le vol en direction de Toulouse décollera à 9h45 ")
tireruntrait()
```

2^{ème} amélioration : passer des arguments (11_1b.py)

On ajoute la fonction **detailduvol(ville,horaire)** :

3^{ème} amélioration : utilisation de boucles et de listes (11_1c.py)

```
#python 3.2
def tireruntrait() :
    print("-----")

def detailduvol(ville,horaire) :
    .....
    .....

ville=["tokyo","Sydney","Toulouse"]
Horaires=.....

for i in range( , ) :
    .....
```

2°) Les fonctions pour récupérer une valeur

Que fait le programme suivant (11_2a.py) :

```
n = 0
s="abracadabra"
for i in range(0,len(s)):
    if s[i] == "a":
        n = n + 1
print(n)
```

Écriture avec une fonction (11_2a_avec fonction.py) :

```
def compter(s) :
    n = 0
    for i in range(0,len(s)):
        if s[i] == "a":
            n = n + 1
    return n
```

.....

Cas général : avec un mot proposé par l'utilisateur (utiliser la fonction **input()**, solution : 11_2b.py)

3°) Choisir entre un passage par valeur et une variable globale

Que fait le programme suivant ? (11_3a.py)

```
def double(x):
    a=2*x

a = 3
double(a)
print(a)
```

Correction pour atteindre l'objectif (11_3a_avec solution) :

```
def double(x):

    a=2*x

a = 3
double(a)
print(a)
```

4°) Exercice : mettre en pratique

Écrire un programme qui indique s'il est possible de construire un triangle avec 3 segments de mesure données qui sont des entiers (une proposition : 11_4.py)

Rappel :

int(input()) permet de récupérer un entier ; **float(input())** permet de récupérer un nombre à virgule.