

TD n°4 : Tests simples

Rappel sur la structure alternative

Une structure alternative complète s'écrit :

```
if (cond) :  
    instruction_Oui_1  
    ...  
    instruction_Oui_N  
else :  
    instruction_Non_1  
    ...  
    instruction_Non_N  
instruction_Suite
```

Les instructions `instructions_Oui_x` sont réalisées qui si `cond` est vraie

Les instructions `instructions_Non_x` sont réalisées si `cond` est fausse

L'instructions `instruction_Suite` est hors de la structure et réalisée quel que soit `cond`

Rappel sur les conditions composées

Composé en ET :

```
if (CondA and CondB) :  
    instructions_vrai  
else : #(facultatif)  
    instructions_faux
```

Composé en OU :

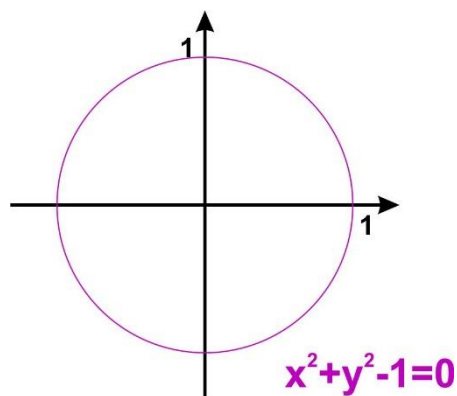
```
if (CondA or CondB) :  
    instructions_vrai  
else : #(facultatif)  
    instructions_faux
```

Composé en NOT :

```
if (not(CondA)) :  
    instructions_vrai  
else : #(facultatif)  
    instructions_faux
```

Application directe n°1 :

Ecrire dans le module `Func_Cible.py` une fonction `INSIDE_V1` qui retourne +1 si le point de coordonnées (X,Y) passées en argument d'entrée se situe à l'intérieur du cercle unité. Dans le cas contraire la fonction retourne la valeur -1. Tester directement cette fonction depuis une console Python.



Le cercle unité

Application directe n°2 :

Modifier la fonction précédente en `INSIDE_V2` pour qu'elle retourne le booléen (True/False) au lieu des valeurs entières +1 et -1.

Application directe n°3 :

Ecrire un programme `Cible.py` qui demande à l'utilisateur deux nombres réels A et B . Le programme affichera si le point de coordonnées (A, B) appartient au cercle unité, elle utilisera pour cela la fonction `INSIDE_V1` ou la fonction `INSIDE_V2`.

Application directe n°4 :

Modifier `INSIDE_V2` pour créer la fonction `INSIDE_V3` afin de vérifier l'appartenance du point (A, B) à un cercle de centre (x_0, y_0) et de rayon R .

On vérifiera au préalable que x_0 et y_0 sont des nombres réels positifs et que si le rayon vérifie : $2 < R < 20$.

La fonction retournera +1 si tout est correct, 0 si la cercle ne vérifie pas les conditions requises, et -1 si le cercle est correctement défini mais que la point (A, B) ne lui appartient pas.

Nb : la fonction Python `type` retourne le type d'une variable.