

Structures conditionnelles

En algorithmique, on est très souvent amené à effectuer des instructions sous certaines conditions. Les conditions sont des expressions logiques qui peuvent être vraies ou fausses (par exemple la comparaison de deux valeurs). Pour effectuer un test conditionnel, on utilise une boucle `if`. Il y a trois syntaxes différentes pour une boucle `if`.

1 Version courte : `if condition : instructions`

Dans ce cas, on fait un seul test.

- Si la *condition* est vraie, les *instructions* sont effectuées.
- Si la *condition* est fausse, les *instructions* sont ignorées.

La syntaxe Python est la suivante :

```
1 if condition :  
2     instructions
```

Essayons de mettre en place une boucle `if` sur un exemple simple.

▷ Recopier la fonction suivante dans l'éditeur de texte :

```
1 def calcul_racines(a, b, c):  
2     delta=b**2-4*a*c  
3     if delta >0 :  
4         x1=(-b-np.sqrt(delta))/(2*a)  
5         x2=(-b+np.sqrt(delta))/(2*a)  
6     return x1, x2
```

▷ Enregistrer et exécuter. Tester avec les polynômes suivants :

$$P_1(x) = x^2 + 5x + 4, \quad P_2(x) = x^2 - 4x + 4, \quad P_3(x) = 2x^2 + x + 2.$$

2 Version intermédiaire : `if condition : instructions else : instructions`

Ici aussi, on fait un seul test.

- Si la *condition* est vraie, le premier bloc d'instructions (*instructions_1*) est effectué.
- Si la *condition* est fausse, le deuxième bloc d'instructions (*instructions_2*) est effectué.

La syntaxe Python est la suivante :

```
1 if condition :  
2     instructions_1  
3 else :  
4     instructions_2
```

Exercice 2.1 Construire une suite d'instructions qui étant donnés deux réels a et b donne le maximum de a et de b .

Exercice 2.2 Ecrire une fonction qui prenant en entrée un réel x renvoie la valeur absolue de x .
On n'utilisera **évidemment pas** la fonction `np.abs` déjà présente dans la bibliothèque `numpy`.

3 Version longue : `if condition : instructions elif : condition : instructions else :instructions`

Dans ce cas, on fait plusieurs tests.

- Si la `condition_1` est vraie, le premier bloc d'instructions (`instructions_1`) est effectué.
- Si la `condition_1` est fausse, on fait un deuxième test.
- Si la `condition_2` est vraie, le deuxième bloc d'instructions (`instructions_2`) est effectué.
- Ce procédé est répété jusqu'à la `condition_k-1`.
- Si elle est vraie, le bloc d'instructions (`instructions_k-1`) est effectué.
- Si la `condition_k-1` est fausse, le bloc d'instructions (`instructions_k`) est effectué.

```
1 if condition_1 :
2     instructions_1
3 elif condition_2:
4     instruction_2
5 ...
6 elif condition_k-1:
7     instructions_k-1
8 else:
9     instructions_k
```

Exercice 3.1 1. Modifier les instructions données dans la Partie 1 pour que, étant donnés trois coefficients a, b, c d'un polynôme $P(x) = ax^2 + bx + c$, on ait en sortie les deux racines distinctes de P (si le discriminant est strictement positif) ou la racine double de P (si le discriminant est nul) ou le message " P n'a pas de racine" (si le discriminant est négatif).

Pour tester un égalité sur Python, on utilise la commande `==`.

2. Enregistrer et exécuter. Tester avec les polynômes donnés en Partie 1.

4 A vous de jouer

Exercice 4.1 1. Montrer que n est un entier pair ssi $(-1)^n = 1$
2. Ecrire alors une fonction de paramètre d'entrée n , qui renvoie l'information pair/impair.

Exercice 4.2 Une séance de cinéma est interdite aux moins de 16 ans. De plus, le prix du billet varie avec l'âge : les seniors (plus de 65 ans) et les mineurs (moins de 18 ans) paient un tarif réduit, les autres un tarif plein.

Écrire une fonction prenant en entrée un entier n correspondant à l'âge de la personne arrivant au cinéma et qui renvoie une réponse quant à son autorisation de voir le film ainsi que le tarif du billet.

Par exemple, si $n=22$, la fonction doit renvoyer : « Vous pouvez voir ce film, le billet est au plein tarif ».

Exercice 4.3 Construire une suite d'instructions qui étant donnés trois réels a, b et c donne le maximum des trois nombres.