

Zones de couleurs [if05] - Exercice

Karine Zampieri, Stéphane Rivière, Béatrice Amerein-Soltner

Unisciel  algoprogram  Version 15 mai 2018

Table des matières

| | | |
|----------|--|----------|
| 1 | Zones de couleurs / pgzcouleurs | 2 |
| 1.1 | Présentation du problème | 2 |
| 1.2 | Algorithmique, Programmation | 3 |
| 2 | Références générales | 4 |

Python - Zones de couleurs (Solution)



Mots-Clés Structures conditionnelles ■

Requis Structures de base ■

Difficulté ●●○



Objectif

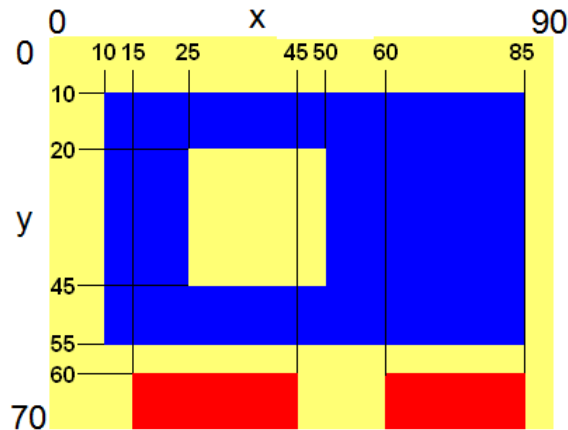
Cet exercice affiche un message en fonction de la position d'un jeton posé sur une feuille de papier composée de zones de couleur.

...(énoncé page suivante)...

1 Zones de couleurs / pgzcouleurs

1.1 Présentation du problème

Sur une table de couleur *blanche* est placée une feuille de papier rectangulaire de 90 cm de large et 70 cm de haut, composée de zones de couleur *jaune*, *bleue* ou *rouge*.



Vous disposez d'un jeton que vous devez placer sur la table **entièrement** sur une seule couleur (blanche, jaune, bleue ou rouge). Le programme vous demande les coordonnées (x, y) de votre jeton, l'origine $(0, 0)$ se trouvant en haut à gauche du rectangle jaune de la feuille. Le problème consiste à afficher **l'un** des messages en fonction de (x, y) :

- « En dehors de la feuille »
- « Dans une zone jaune »
- « Dans une zone bleue »
- « Dans une zone rouge »

Voici un exemple du résultat attendu :

```
Position x, y? 16 12
==> Dans une zone bleue
```



Attention

La difficulté de l'exercice est que vous devez écrire un script de sorte qu'il y ait au **maximum une alternative Si par message** affiché.

...(suite page suivante)...

1.2 Algorithmique, Programmation



Comment pouvez-vous procéder en vertu de l'impératif :
« au maximum une alternative **Si** par message ».

Solution simple

En écrivant d'abord des expressions conditionnelles pour chacune des différentes zones que l'on stockera dans des booléens **b1** à **b5**. Ensuite, il suffira d'écrire une alternative **Si** pour chacun des cas en composant **b1** à **b5** et les opérateurs logiques **Non**, **Et** et **Ou**.



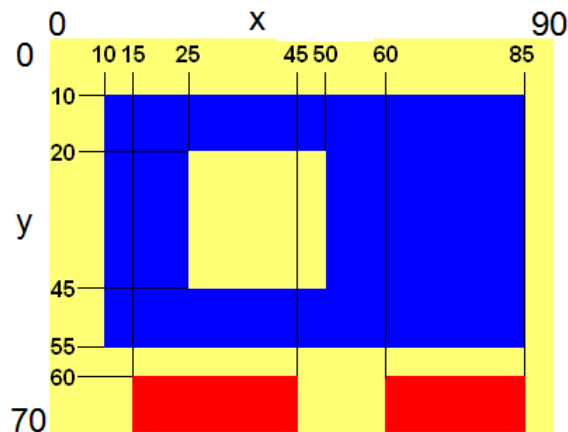
Écrivez un script qui saisit la position du jeton dans des entiers **x** et **y**.
Affichez l'invite :

Position **x**, **y**?



Déterminez les booléens **b1** à **b5** par des tests exprimés en fonction de **x** et **y** selon les différentes zones. (Il n'est pas nécessaire de définir des constantes pour identifier les différentes zones : écrivez directement les expressions.)

- Booléen **b1** : la zone Feuille
- Booléen **b2** : le rectangle bleu au centre qui comprend le rectangle jaune
- Booléen **b3** : le rectangle jaune qui est dans le rectangle bleu
- Booléen **b4** : le rectangle rouge de gauche en bas
- Booléen **b5** : le rectangle rouge de droite en bas



Pour chacun des cas, écrivez l'alternative **Si**.

Aide simple

Toujours les mêmes principes : éliminez les cas triviaux puis affinez dans le **Si non**.



Testez. Exemples d'exécution :

```
Position x, y? 16 12  
==> Dans une zone bleue
```

```
Position x, y? 30 22  
==> Dans une zone jaune
```

```
Position x, y? 64 62  
==> Dans une zone rouge
```

```
Position x, y? -5 86  
==> En dehors de la feuille
```



Validez votre script avec la solution.

Solution Python @[pgzcouleurs1.py]

```
def PGZCouleurs1():  
    print("Position x, y? ", sep="", end="")  
    x = int(input())  
    y = int(input())  
    b1 = (0 <= x and x <= 90 and 0 <= y and y <= 70)  
    b2 = (10 <= x and x <= 85 and 10 <= y and y <= 55)  
    b3 = (25 <= x and x <= 50 and 20 <= y and y <= 45)  
    b4 = (15 <= x and x <= 45 and 60 <= y and y <= 70)  
    b5 = (60 <= x and x <= 85 and 60 <= y and y <= 70)  
    if not b1:  
        print("==> En dehors de la feuille")  
    elif b4 or b5:  
        print("==> Dans une zone rouge")  
    elif b2 and not b3:  
        print("==> Dans une zone bleue")  
    else:  
        print("==> Dans une zone jaune")
```

PGZCouleurs1()

2 Références générales

Comprend [franceioi-AL1 :c3 :ex3(d5)] ■