

Algorithmes et Pensée Computationnelle

Programmation de base - Exercices avancés

Cette feuille d'exercices avancés vous permettra d'approfondir vos connaissances des notions vues en cours. Le code présent dans les énoncés se trouve sur Moodle, dans le dossier "Ressources".

Les langages qui seront utilisés pour cette séance sont Java et Python. Assurez-vous d'avoir bien installé IntelliJ. Si vous rencontrez des difficultés, n'hésitez pas à vous référer au guide suivant : [tutoriel d'installation des outils et prise en main de l'environnement de travail](#).

1 Représentation de nombres entiers

2 Bases en programmation

Question 1: (🕒 10 minutes) Manipulation des chaînes de caractères (Java ou Python)

Il est possible d'obtenir la longueur d'une chaîne de caractère (ou d'une liste ou d'un dictionnaire) en utilisant la fonction `len()`. Gardez votre variable `mon_mot` et créez une nouvelle variable nommée `ln_mon_mot` contenant le nombre de caractère de la variable `mon_mot`, puis une nouvelle variable `moitie` contenant la première moitié de la variable `mon_mot` (utilisez la variable que vous venez de créer). Affichez le résultat.

💡 Conseil

La fonction présentée dans l'énoncé de la question n'est valable que pour python. L'équivalent pour Java est la fonction `length()`.

>_ Solution

Python :

```
1 mon_mot = "Hard But Cool !!"
2 ln_mon_mot = len(mon_mot)
3 moitie_mon_mot = mon_mot[:ln_mon_mot//2]
4 print(ln_mon_mot)
5 print(moitie_mon_mot)
```

Java :

```
1 String mon_mot = "Hard But Cool !!";
2 int ln_mon_mot = mon_mot.length();
3 String moitie = mon_mot.substring(0,ln_mon_mot/2);
4 System.out.println(ln_mon_mot);
5 System.out.println(moitie);
```

3 Opérateurs et conditions Booléennes (Python uniquement)

Question 2: (🕒 20 minutes) Le juste prix Optionnel

Dans le programme suivant, nous vous donnons un nombre aléatoire entre 0 et 30 dans la variable `number`, écrivez un programme qui demande à l'utilisateur de deviner le nombre tiré au sort. L'utilisateur a 5 chances pour le trouver. S'il se trompe, donnez-lui un indice (le nombre qu'il a écrit est-il plus grand ou plus petit que celui qu'il cherche?). Vous pouvez vous amuser à modifier le nombre de chances ou le nombre de possibilités (par exemple 10 chances pour trouver un nombre entre 0 et 100).

```
1 # Programme écrit en Python
2 from random import randint
```

```
3 number = randint(0,30)
4
5 #Votre code
```

>_ Solution

```
1 # Programme écrit en Python
2 from random import randint
3 number = randint(0,30)
4 x = int(input("Choisissez un nombre: "))
5 if x == number:
6     print("Yeeah!")
7 elif x < number:
8     print("Trop bas!")
9 else:
10    print("Trop haut!")
11
12 x = int(input("Choisissez un nombre: "))
13 if x == number:
14     print("Yeeah!")
15 elif x < number:
16     print("Trop bas!")
17 else:
18     print("Trop haut!")
19
20 x = int(input("Choisissez un nombre: "))
21 if x == number:
22     print("Yeeah!")
23 elif x < number:
24     print("Trop bas!")
25 else:
26     print("Trop haut!")
27
28 x = int(input("Choisissez un nombre: "))
29 if x == number:
30     print("Yeeah!")
31 elif x < number:
32     print("Trop bas!")
33 else:
34     print("Trop haut!")
35
36 x = int(input("Choisissez un nombre: "))
37 if x == number:
38     print("Yeeah!")
39 elif x < number:
40     print("Trop bas!")
41 else:
42     print("Trop haut!")
```

Le problème avec cette solution est le suivant : Si le joueur trouve la réponse, le jeu va continuer, une façon plus propre et correcte de coder ce jeu est d'utiliser une boucle (prochain chapitre).

```
1 from random import randint
2 number = randint(0, 30)
3 for i in range(5):
4     x = int(input("Choisissez un nombre: "))
5     if x==number:
6         print("Yeah!")
7         break
8     elif x<number:
9         print("Trop petit!")
10    else:
11        print("Trop grand!")
```

Ici le code est plus concis et permet de s'arrêter lorsque le joueur aura trouvé la bonne réponse.

Question 3: (🕒 20 minutes) Pierre, Feuille, Ciseaux Optionnel

Demandez à l'utilisateur d'entrer soit pierre, soit feuille, soit ciseaux. L'ordinateur choisira son coup au

hasard (s'il choisi 1 ce sera pierre, si c'est 2 ce sera feuille et si c'est 3 ce sera ciseaux). Les règles sont les règles classiques, une manche gagnante.

```
1 # Programme écrit en Python
2 from random import randint
3 number = randint(1,3)
4
5 #Votre code
```

>_ Solution

```
1 # Programme écrit en Python
2 from random import randint
3 number = randint(1,3)
4 if number == 1 :
5     ordi = "pierre"
6 elif number == 2 :
7     ordi = "feuille"
8 else :
9     ordi = "ciseaux"
10
11 player = input("Choisissez un signe (pierre, feuille, ciseaux) : ")
12
13 print("ordi a choisi " + ordi)
14
15 if player != "pierre" and player != "feuille" and player != "ciseaux" :
16     print("symbole invalide")
17 else :
18     if ordi == "pierre" :
19         if player == "pierre" :
20             print("égalité")
21         elif player == "feuille" :
22             print("gagné")
23         else :
24             print("perdu")
25     elif ordi == "feuille" :
26         if player == "pierre" :
27             print("perdu")
28         elif player == "feuille" :
29             print("égalité")
30         else :
31             print("gagné")
32     else :
33         if player == "pierre" :
34             print("gagné")
35         elif player == "feuille" :
36             print("perdu")
37         else :
38             print("égalité")
```

Vous pouvez également utiliser une boucle pour augmenter le nombre de manches.