



Apprentissage Performant & Innovant du Numérique

WORKSHOP – PYTHON RECHERCHE PAR DICHOTOMIE

Step 1

Ecrire une fonction main, qui renvoie comme valeur de retour un int.

La fonction doit afficher dans la console : « Bienvenue dans le jeu de la dichotomie ! » et fermer le programme.

Correction

```
def main() -> int:
    print("Bienvenue dans le jeu de la dichotomie !")
    return 0

main()
```

Step 2

Dans la fonction main, récupérer le nom du joueur dans une variable userName, qu'il rentrera directement en console avec un commentaire.

Correction

```
def main() -> int:
    print("Bienvenue dans le jeu de la dichotomie !")
    #faire connaissance
    userName = input('Quel est votre nom ? ')
    return 0

main()
```

Step 3

Donner les instructions du jeu en mettant en variable le nombre minimum et maximum.

Exemple « Paul pense à un nombre entre 0 et 100 » où 0 est la valeur de la variable minVal et 100 est la valeur de la variable maxVal.

Pourquoi ne peut-on pas mettre la variable directement dans la fonction **input** ?

Correction

Car il faut convertir le int en string si on utilise la concaténation '+'

```
def main() -> int:
    print("Bienvenue dans le jeu de la dichotomie !")
    #faire connaissance
    userName = input('Quel est votre nom ? ')
    #initialisation des variables
    minValue = 0
    maxValue = 100
    #lancement du jeu
    print( userName + " pense à un nombre entre " + str(minValue) + " et " +
str(maxValue))
    print("Le jeu commence")
    return 0

main()
```

Step 4

Maintenant créer une structure itérative avec les conditions nécessaires pour deviner le nombre par la méthode de « dichotomie ».

N'oubliez pas d'initialiser un compteur et l'utiliser pour compter le nombre de fois nécessaire pour trouver le bon nombre.

Quel type de boucle est la plus adaptée ?

Si la réponse est bonne le jeu doit s'arrêter.

Correction

Puisque nous ne savons pas à l'avance le nombre total d'itérations, la boucle tant que « while » est la plus adaptée avec une condition pour terminer (ne pas utiliser « Break »).

```
Entrée [1]: def main() -> int:
    print("Bienvenue dans le jeu de la dichotomie !")
    #faire connaissance
    userName = input('Quel est votre nom ? ')
    #initialisation des variables
    minValue = 0
    maxValue = 100
    compt = 0
    reponse = '+'
    #Lancement du jeu
    print(userName + " pense à un nombre entre " + str(minValue) + " et " + str(maxValue))
    print("Le jeu commence ")
    #Le boucle pour deviner
    while reponse != '=':
        milieu = int((minValue + maxValue)/2)
        reponse = input('Est-ce ton nombre '+userName+' est plus grand, plus petit ou égal à '+str(milieu)+' ? [+/-/=] ')
        compt = compt + 1
        if reponse == '+':
            minValue = milieu
        elif reponse == '-':
            maxValue = milieu
```

Step 5

Pour finir en sortant de la boucle, il faut afficher un compte rendu qui contient le nombre deviné et le nombre de questions posées pour trouver et un message de sortie.

Correction

Pour l'exécuter n'oubliez pas d'écrire :

main()

```
Entrée [8]: def main() -> int:
    print("Bienvenue dans le jeu de la dichotomie !")
    #faire connaissance
    userName = input('Quel est votre nom ? ')
    #initialisation des variables
    minValue = 0
    maxValue = 100
    compt = 0
    reponse = '+'
    #Lancement du jeu
    print(userName + " pense à un nombre entre", minValue, "et", maxValue)
    print("Le jeu commence ")
    #Le boucle pour deviner
    while reponse != '=':
        milieu = int((minValue + maxValue)/2)
        reponse = input('Est-ce ton nombre ' + userName + ' est plus grand, plus petit ou égal à ' + str(milieu) + ' ? [ +/-/=] ')
        compt = compt + 1
        if reponse == '+':
            minValue = milieu
        elif reponse == '-':
            maxValue = milieu
    #Le resultat final
    print("Ton nombre est", milieu)
    print("J'ai trouvé en", compt, "questions !")
    print("Au revoir " + userName)
    return 0
```