

## Introduction à Python 1 – Données d'entrée et de sortie

Ex. 1 : Élaborez un programme qui demande à l'utilisateur d'indiquer la planète de notre système solaire qui est en moyenne la plus proche de la Terre. Après qu'il a répondu à la question, affichez la réponse : Vous avez dit [réponse]. Si vous avez dit "Mercure" la réponse est correcte, la planète Vénus est la plus proche de la Terre lorsqu'elle parcourt son orbite, mais Mercure est,

```
# demander à l'utilisateur de présenter des données d'entrée
Planète = input("Quelle est la planète la plus proche de la Terre? ")

# afficher le message final
print("Vous avez dit", planète, "Si vous avez dit \"Mercure\" la réponse est correcte, la planète Vénus est la plus proche de la Terre lorsqu'elle parcourt son orbite, mais Mercure est, en moyenne, le plus
```

en moyenne, le plus proche.

Ex. 2 : Élaborez un programme qui demande à l'utilisateur de nommer une nouvelle étoile qui a été repérée et d'indiquer le nombre de planètes qui l'entourent, puis affichez une phrase lui indiquant le nom qu'ils ont choisi et le nombre de planètes que l'on peut voir autour de cette étoile. Print(

```
# demander à l'utilisateur de présenter des données d'entrée
Nom= input("Quel est le nom de l'étoile que vous avez découverte?")
planètes = input("Combien de planètes entourent-elles l'étoile que vous avez découverte? ")

# afficher le message final
print("vous avez trouvé une étoile appelée", nom, "et il y a", planètes, "planètes qui l'entourent")
```

## Introduction à Python 2 – Mathématiques

Ex. 1 : Élaborez un programme qui demande à l'utilisateur d'indiquer le nombre de satellites que Jupiter (79) et Saturne (82) ont et calculez la différence entre les deux. Données de sortie finales. **Saturne compte 82 satellites, soit 3 de plus que les 79 satellites de Jupiter.**

```

# demander à l'utilisateur de présenter des données d'entrée
Jupiter = int(input("Combien de satellites Jupiter a-t-il? "))
Saturne = int(input("Combien de satellites Saturne a-t-il? "))

# calculer la différence
différence = Saturne - Jupiter

# afficher les résultats du calcul
print("Saturne a", saturne, "satellites qui sont", la différence, "plus que Jupiter", Jupiter,
      ".....")

```

Ex 2 : Demandez à l'utilisateur le nombre de jours terrestres d'une (1) année sur Mars (687) et d'une (1) année sur Vénus (225). Calculez le ratio entre eux. Données de sortie finales : **Une année sur Mars est 3,05 fois plus longue que celle sur Vénus.**

```

# demander à l'utilisateur de présenter des données d'entrée
Mars = int(input("Quelle est la durée d'une année sur Mars en jours terrestres? "))
Vénus = int(input("Quelle est la durée d'une année sur Vénus en jours terrestres? "))

#calculer le ratio
ratio = round(Mars/Vénus, 2)

# afficher les résultats du calcul
print("Une année sur Mars est", ratio, "nombre de fois plus longue que celle sur Vénus")

```

Ex. 3 : Demandez à l'utilisateur le rayon (en km) de la Terre (6 371) et du Soleil (695 700) et calculez les deux volumes. Ensuite, découvrez le nombre de terres entières qui peut correspondre à la taille du Soleil et affichez la réponse finale. Remarque : volume d'une sphère =  $\frac{4}{3} * \pi * r^{** 3}$

Données de sortie finales : **Vous pouvez placer 1 302 097 terres dans le Soleil. Il faudra plusieurs terres.**

```
à partir de import.math.pi

# demander à l'utilisateur de présenter des données d'entrée
rTerre = int(input("Quel est le rayon de la Terre en km"))

rSoleil = int(input("Quel est le rayon du Soleil en km"))

#faire des calculs
vTerre = 4/3 * pi * rTerre**3
vSoleil = 4/3 * pi * rSoleil**3

ratio = vSoleil/vTerre

# afficher le message final
print("Vous pouvez placer", round(ratio),"terres dans le Soleil. Il faudra plusieurs terres")
```

### Introduction à Python 3 – Instructions conditionnelles If

Ex. 1 : Élaborez un programme qui demande à l'utilisateur d'indiquer la distance entre une planète fictive et son étoile en millions de km. Si la distance est inférieure à 149 millions de km, la planète est plus proche de son étoile que la Terre ne l'est du Soleil. Sinon, elle est plus éloignée. Utilisez une instruction If et Else pour déterminer la bonne réponse et dire à l'utilisateur si sa planète fictive est plus proche ou plus éloignée que la Terre ne l'est du Soleil.

```
# demander à l'utilisateur de présenter des données d'entrée
distance = int(input("À quelle distance se trouve la planète par rapport au Soleil en millions de km?
"))

# vérifier les critères pour afficher correctement le message
if distance < 149 :
    print("La planète est plus proche de son étoile que la Terre ne l'est du Soleil")
else:
    print("La planète est encore plus proche de son étoile que la Terre ne l'est du Soleil")
```

Ex.2 : Élaborez un programme qui demande à l'utilisateur d'indiquer l'âge du Soleil en milliards d'années et l'étape de sa vie à laquelle il se trouve :

0 à 10 milliards d'années – Séquence principale

**Afficher** : Le Soleil est à l'étape de la séquence principale telle qu'elle est actuellement. À cette étape, le Soleil utilise la fusion nucléaire des noyaux d'hydrogène pour produire de l'hélium et émet de l'énergie sous forme de lumière et de chaleur.

10 à 11 milliards d'années – Géante rouge

**Afficher** : Le Soleil est maintenant une géante rouge. Au cours de cette étape de sa vie, le Soleil a épuisé tout l'hydrogène qui se trouve dans son noyau et va maintenant atteindre 400 fois sa taille d'origine et engloutir complètement la Terre. Il refroidira et rougeoiera également, mais nous ne serons pas là pour le voir.

Plus de 11 milliards d'années – Naine blanche

**Afficher** : Le Soleil est maintenant une naine blanche. Il s'agit de l'étape finale de notre Soleil et il se rétrécira pour atteindre à peu près la taille de la Terre. Le Soleil n'est pas assez grand pour finir dans une explosion dite supernova comme le font les étoiles de plus grande taille.

```
# demander à l'utilisateur de présenter des données d'entrée
Âge = int(input("Quel est l'âge du Soleil en milliards d'années? "))

#vérifier les critères pour afficher correctement le message
if âge <= 10 :
    print("Le Soleil est à l'étape de la séquence principale telle qu'elle est actuellement. À cette étape,
le Soleil utilise la fusion nucléaire des noyaux d'hydrogène pour produire de l'hélium et émet de
l'énergie sous forme de lumière et de chaleur.")

elif âge <= 11 :
    print("Le Soleil est maintenant une géante rouge. Au cours de cette étape de sa vie, le Soleil a
épuisé tout l'hydrogène qui se trouve dans son noyau et va maintenant atteindre 400 fois sa taille
d'origine et engloutir complètement la Terre. Il refroidira et rougeoiera également, mais nous ne
serons pas là pour le voir.")

else:
    print("Le Soleil est maintenant une naine blanche. Il s'agit de l'étape finale de notre Soleil et il se
rétrécira pour atteindre à peu près la taille de la Terre. Le Soleil n'est pas assez grand pour finir dans
une explosion dite supernova comme le font les étoiles de plus grande taille.
```

**Introduction à Python 4 – La boucle While**

**Exercice** – Élaborez un programme qui utilise une boucle While pour calculer la force gravitationnelle entre deux éléments compte tenu de leurs masses  $m_1$  et  $m_2$  en kg, de la distance  $r$  qui les sépare l'un de l'autre en mètres et de la constante gravitationnelle  $G = 6,67 * 10^{-11}$ . La boucle doit se poursuivre autant de fois que l'utilisateur le souhaite en utilisant une variable comme la variable "de nouveau".

Deux éléments, même vous et votre ami, auront une attraction gravitationnelle l'un sur l'autre. L'attraction gravitationnelle entre vous et votre ami sera négligeable par rapport à celle de la Terre ou du Soleil, mais elle existe. La formule utilisée pour calculer la force gravitationnelle est la suivante :

$F_g = (G * m_1 * m_2) / r^2$  et les unités finales sont N pour Newtons, l'unité de mesure utilisée pour mesurer toutes les forces. Affichez votre réponse par la suite pour dire :

La force entre les deux éléments est [réponse] N.

N'oubliez pas de demander à l'utilisateur s'il souhaite répéter le code pendant qu'il est encore dans la boucle.

```
#initialiser la variable de nouveau et en la remplaçant par un Y en majuscule afin que le code rentre
dans la boucle
de nouveau = "Y"

# créer une boucle pour continuer à exécuter l'instruction tant que l'utilisateur répond par un Y pour
dire Oui
while de nouveau == "Y" :

    # demander à l'utilisateur de présenter des données d'entrée, vous pourriez également utiliser la
fonction float au lieu de int

    m1 = int(input("Entrez la masse du premier élément en kg: "))
    m2 = int(input("Entrez la masse du second élément en kg: "))
    r = int(input("Entrez la distance qui les sépare les deux éléments l'un de l'autre en mètres : "))

    #constante pour la formule
    G = 6,67 * 10 ** -11

    # calculer la force gravitationnelle entre les deux éléments
    F = (G * m1 * m2)/r**2

    #afficher des données de sortie finales sur la force
    print("La force entre les deux éléments est ", F, "N.")

    #demander à l'utilisateur s'il veut initialiser de nouveau la variable. La fonction upper() garantie
que les caractères d'une réponse sont en majuscule.
    de nouveau = input("Souhaitez-vous calculer la force entre les deux éléments? (O/N").upper()
```