

Python

Utilisation de variables et de mathématiques

Types de données

a) Chaîne

- Peut contenir un texte quelconque : lettres de l'alphabet, chiffres, signes de ponctuation, caractères spéciaux
- Est généralement trouvée entre " " (guillemets)
- **Par exemple** : `nom = input("Entrez votre nom:")`

b) Nombre entier

- Peut contenir uniquement des valeurs entières (nombres entiers positifs ou négatifs)
- Pour convertir les données d'une chaîne en nombres entiers, utilisez la fonction **int()**

c) Réel ou flottant

- Peut contenir uniquement des valeurs réelles (nombres positifs ou négatifs pouvant contenir des nombres décimaux)
- Pour convertir les données d'une chaîne en nombres réels, utilisez la fonction **float()**

Par défaut, la fonction **input()** convertit les données d'entrée en une chaîne.

Utilisez la fonction **int()** avec la fonction **input()** pour entrer des données en nombres entiers dans le programme.

Exemple : `Largeur = int(input("Entrez la largeur:"))`

Utilisez la fonction **float()** avec la fonction **input()** pour entrer des données en nombres réels ou flottant dans le programme.

Exemple : `Largeur = float(input("Entrez la largeur:"))`

Opérateurs mathématiques et fonctions

Python suit les règles de PEDMAS lors de l'évaluation d'une expression mathématique.

Comme les données d'entrée qui utilisent la **fonction input()** entraînent la conversion de ces données en chaîne, toute donnée d'entrée numérique doit être convertie en nombre entier ou en valeur réelle, selon les besoins. Pour convertir une valeur en son équivalent en nombre entier, utilisez la fonction **int()**. Pour convertir une valeur en son équivalent en nombre réel, utilisez la fonction **float()**.

Opérateurs communs en mathématiques

Opération	Résultat	Opération	Résultat
-----------	----------	-----------	----------

Opération	Résultat	Opération	Résultat
$x + y$	somme de x et y	+x	<i>x demeure inchangé</i>
$x - y$	différence de x et y	abs(x)	valeur absolue ou magnitude de x
$x * y$	produit de x et y	int(x)	<i>x converti en nombre entier</i>
x / y	quotient de x et y	round(x, y)	<i>x sera arrondi à un nombre décimal y</i>
$x // y$	quotient de x et y en nombre entier	float(x)	<i>x converti en un nombre avec virgule flottante ou en nombre réel</i>
$x \% y$	reste de x / y	pow(x, y)	<i>x à la puissance y</i>
-x	<i>x, nombre négative</i>	$x ** y$	<i>x à la puissance y</i>

Comprendre la différence entre les opérateurs / et /, et %.

Pour mieux comprendre ces opérateurs, utilisez l'exemple 11/4. 11/4 est égale à 2,75 ou à un quotient de 2 avec un reste de 3.

print(11 / 4) affichera la valeur 2,75

print(11 // 4) affichera 2, car dans le nombre 11, il y a deux fois 4, également connu sous le nom de division entière.

print(11 % 4) affichera 3, car 3 est le reste lorsque 11 est divisé par 4

Exemple :

```
# introduction
print("Nombre de planètes")
print("-----")
print("") #une instruction print vide est un bon moyen d'ajouter une ligne
vierge

# demander les données d'entrée de l'utilisateur
Planètes rocheuses = float(input("Combien y a-t-il de planètes rocheuses
proches du Soleil? ")) #4
Planètes géantes gazeuses = float(input("Combien y a-t-il de planètes
géantes gazeuses externes?")) # 4

# calculs
Total = planètes rocheuses + planètes géantes gazeuses

#afficher des données de sortie pour l'utilisateur
print("")
print "Au total il y a", total, "planètes dans notre système solaire")
print("Il y a", planètes rocheuses, "planètes proches du Soleil qui sont
plus petites et faites de roche. On les appelle les planètes telluriques")
```

```
print("Il y a", planètes géantes gazeuses, "planètes externes qui sont de  
grandes boules de gaz.")
```

Le développement correcte de votre code, tout comme il est affiché ci-dessus, contenant un espacement et des commentaires, est essentielle pour le débogage (correction des erreurs) et la collaboration avec d'autres élèves. Notez également que la fonction "input" sera évaluée d'abord, puis la fonction "float", car python fonctionne à l'envers sur ces lignes demandant ainsi aux utilisateurs d'entrer leurs données.

La bibliothèque des mathématiques - L'un des avantages clés de python en tant que langage de codage est le grand nombre de bibliothèques contenant des fonctions déjà prêtes que nous pouvons utiliser de façon que nous n'ayons pas besoin de réinventer la roue. Une de ces bibliothèques est la bibliothèque des mathématiques. La bibliothèque des mathématiques contient de nombreuses fonctions mathématiques importantes qui peuvent être utilisées dans un programme python. La bibliothèque des mathématiques est mise à disposition du programme python à l'aide de l'instruction **import math**.

Elle n'est pas nécessaire pour les opérations mathématiques simples décrites dans le tableau ci-dessus.

Voici certaines des fonctions mathématiques couramment utilisées :

math.pow(x, y) - la valeur de retour x élevée à la puissance y.

math.sqrt(x) - la valeur de retour de la racine carrée de x.

math.hypot(x, y) - la valeur de retour de la norme euclidienne, racine carrée ($x^2 + y^2$). Il s'agit de la longueur du vecteur du point d'origine au point (x, y).

math.pi - La constante mathématique $\pi = 3.141592\dots$, à la précision disponible.

Exemple :

```
import math

# introduction
print("Caractéristiques de la Terre")
print("-----")
print("")

# demander les données d'entrée de l'utilisateur
rayon = float(input("Entrez le rayon de la Terre en km: ")) # réponse :
6371

#faire des calculs
surface = 4 * math.pi * rayon ** 2
masse = 5 * 10**24

#afficher les caractéristiques comme il le faut pour l'utilisateur
```

```
surface = round(surface,2)
print("")
print("La Terre a une surface de:", surface, "en km carré et une masse
de", masse, "en kg")
```

Si `math.pi` est le seul élément que nous voulons utiliser de la bibliothèque des mathématiques, nous pouvons aussi écrire : **à partir de `import math.pi`**. Si nous faisons cela, nous pouvons simplement écrire **pi** au lieu de **math.pi** quand nous l'utilisons dans un calcul.

Rappelez-vous que nous devons placer les données d'entrée entre les parenthèses de la fonction `int()` pour les convertir en nombre entier, car ces données utilisées seules deviendront une chaîne*

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.**

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.**

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 1 302 096 terres dans le Soleil. Il faudra plusieurs terres.**