



- 1 Commentaires en Python..... 1
- 2 Commentaires en Python..... 1
- 3 Formatage des entrées..... 1
 - 3.1. Formatage d'une variable..... 1
 - 3.2. Fonction input 2
- 4 Affichage des sorties..... 2
 - 4.1. Affichage formaté 2
 - 4.2. Formatage numérique 3
- 5 Recherche des erreurs – Débogage..... 3

1 Commentaires en Python

```
from math import sqrt #ceci est un commentaire sur une seule ligne
"""
Ceci est commentaires sur
plusieurs
lignes
"""
```

2 Commentaires en Python

N'importe où sur une seule ligne : #

Sur plusieurs lignes tout e la largeur de l'éditeur : """ """

3 Formatage des entrées

3.1.Formatage d'une variable

```
# base 10 par default
d =171
print("décimal ",d)
# binaire (base 2 ) avec le préfixe 0b
b = 0b10101011
print("binaire ",b)
# hexadécimal (base 16 ) avec le préfixe 0x
h = 0xAB
print("hexadécimal ",h)
# octal (base 8 ) avec le préfixe 0o
o = 0o253
print("octal ",o)
```

PROBLÈMES	SORTIE	DEBUG
	décimal	171
	binaire	171
	hexadécimal	171
	octal	171



3.2.Fonction input

```
chaine = input("Ceci entre une chaine de caractères ")
print(chaine)
entier = int(input("Ceci entre un un entier "))
print(entier)
flottant = float(input("Ceci entre un floattant "))
print(flottant)
booleen = bool(input("Ceci entre un booléen "))
print(booleen)
```

4 Affichage des sorties

4.1.Affichage formaté

```
print("La racine carré de 2 vaut {0:.10f}".format(sqrt(2)))
print("Le périmètre du rectangle vaut : {0:e} mètres carrés".format(100))
print("La conversion de {0:d} en binaire vaut : {1:b}".format(10,10))
```

La racine carré de 2 vaut 1.4142135624

Le périmètre du rectangle vaut : 1.000000e+02 mètres carrés

La conversion de 10 en binaire vaut : 1010

```
str.format(*args, **kwargs)
```

Formate une chaîne. La chaîne sur laquelle cette méthode est appelée peut contenir du texte littéral ou des emplacements de remplacement délimités par des accolades `{}`. Chaque champ de remplacement contient soit l'indice numérique d'un argument positionnel, ou le nom d'un argument donné par mot-clé. Renvoie une copie de la chaîne où chaque champ de remplacement est remplacé par la valeur de chaîne de l'argument correspondant.

s – strings

d – decimal integers (base-10)

f – floating point display

c – character

b – binary

o – octal

x – hexadecimal with lowercase letters after 9

X – hexadecimal with uppercase letters after 9

e – exponent notation



4.2. Formatage numérique

Formatage numériques, nombres entiers dans différentes bases , nombres flottants sous différentes notations et différentes base.

```
import math
import sys

s="int:{0:d};hex:{0:x};oct{0:o};bin:{0:b}".format(42)
print(s)
print(math.pi)
print("{:.4e}".format(math.pi))
print("{:g}".format(math.pi))
print("{:.2f}".format(math.pi))
```



5 Recherche des erreurs – Débogage

Pyzo

Création d'un point d'arrêt : cliquez à gauche dans l'éditeur pour faire apparaitre un rond rouge en face de la ligne où vous voulez contrôler votre programme.

```
utilisation espions_pointdarret.py
1
2 i = 0
3
4 while i<10 :
5     i+=1
6     print("valeur de l'indice :" , i)
```

Demander l'exécution du programme. Celui-ci démarre et s'arrête en face de votre rond rouge :

Dans Workspace vous avez pour le même instant la valeur courantes de vos différentes variables :

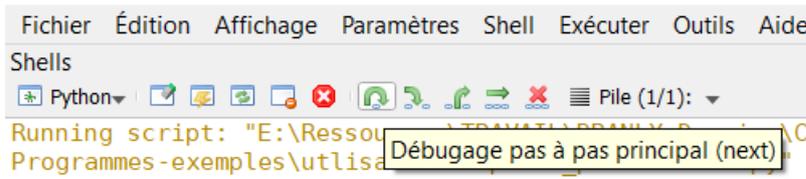
```
utilisation espions_pointdarret.py
1
2 i = 0
3
4 while i<10 :
5     i+=1
6     print("valeur de l'indice :" , i)
```

Name	Type	Repr
i	int	0

Faites exécuter débogage pas à pas principal :



Vous pouvez alors observer l'évolution de vos variables pas à pas :



Workspace

Name	Type	Repr
i	int	2

VSCode

Même principe

