

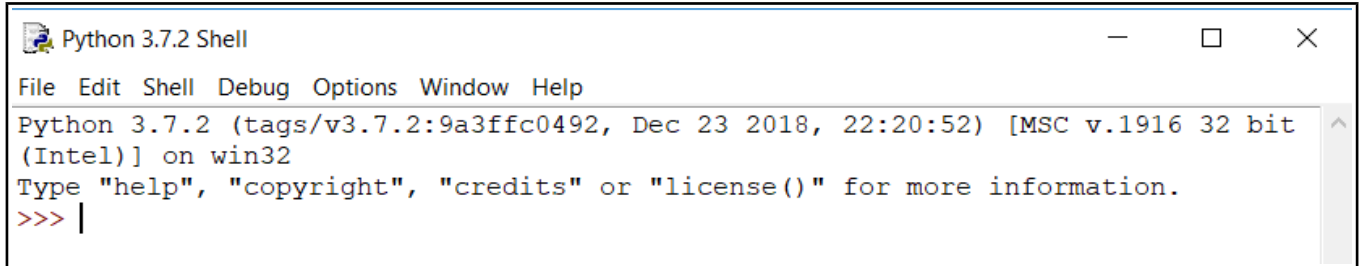
# Premières commandes

## Mode interactif

Python présente la particularité de pouvoir être utilisé dans de nombreux environnements. Vous allez d'abord l'utiliser en mode interactif, c'est-à-dire de manière à dialoguer avec lui directement depuis le clavier.

Pour cela, vous utiliserez l'environnement de travail IDLE :

Les trois caractères `>>>` constituent le signal d'invite indiquant que Python est prêt à exécuter une commande.



## Calculer avec Python, priorité des opérations

`>>> 5+3` *# Les espaces sont-ils optionnels ?*

☐ OUI☐ NON

`>>> 2 - 9`

*# La hiérarchie des opérations mathématiques est-elle respectée ?*

`>>> 7 + 3 * 4`

☐ OUI☐ NON

`>>> (7+3)*4+3`

`>>> 20 / 3`

*# Quelle différence entre / et // ?*

`>>> 20 // 3`

`>>> 20.5 / 3`

`>>> 8,7 / 5`

*# Pourquoi le logiciel renvoie-t-il ce message ?*

`>>> 10 % 3`

*# Que fait l'opérateur % ?*

`>>> 10 % 5`

`>>> 10**3`

*# Que fait l'opérateur \*\* ?*

`>>> 5**2`

## Affecter une variable à une valeur, l'afficher

```
>>> n = 7
>>> message = "Quoi de neuf ?"
>>> pi = 5.43
```

*# Que fait Python ?*

```
>>> n
```

```
>>> message
```

```
>>> pi
```

*# Que fait Python ?*

```
>>> print(message)
```

```
>>> print(n)
```

```
>>> a=7
```

```
>>> a
```

```
>>> a=a+1
```

```
>>> a
```

*# Quelle différence avec précédemment ?*

*# Que signifie cette instruction ?*

## Affectations multiples

```
>>> x = y = 7
```

```
>>> x
```

```
>>> y
```

```
>>> m, n = 7.3, 12
```

```
>>> m
```

```
>>> n
```

```
>>> y=3*m + n/5
```

```
>>> y
```

```
>>> a, b = 3,7
>>> a = b
>>> b = a
>>> print(a,b)
```

*# Commentez cette liste d'instructions*

```
>>> a, b ,c ,d = 3, 4, 5, 6
>>> print(a, b, c, d)
>>> a, b = b,a
>>> print(a, b, c, d)
```

*# Commentez cette liste d'instructions*

## Fonction print()

```
>>> print(17+3)
```

```
>>> print('17+3')
```

```
>>> print(5*' ')
```

```
>>> print('ada','est','un','joli','prénom')
```

```
>>> print('ada','est','un','joli','prénom',sep='*')
```

```
>>> print('ada','est','un','joli','prénom',sep='')
```

## Type de données : entier, flottant,booléen, caractère, tableau

```
>>> type(7)
```

*# A quoi sert la fonction type() ?*

```
>>> type(3.14)
```

```
>>> type(7.0)
```

```
>>> type(True)
```

```
>>> type([1,2,3])
```

```
>>> type('coucou')
```

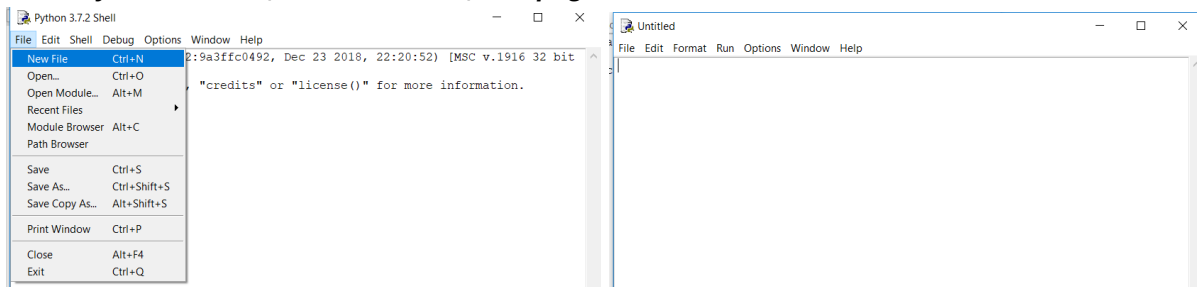
# Premier programme

## Mode édition :

*Le mode interactif est très intéressant pour tester certaines commandes ou vérifier certains calculs. Vous allez cependant être amenés à écrire des programmes de plusieurs dizaines ou centaines de lignes. Vous n'allez donc pas utiliser le mode interactif dans ce cas !*

*Pour cela , vous allez écrire des programmes dans des fichiers au format .py que vous exécuterez ensuite à l'aide d'IDLE (ou d'un autre interpréteur).*

**Allons-y : Dans IDLE, File > New File, une page blanche s'ouvre :**



## Exécutions conditionnelles : l'instruction if

*Tapez le code suivant, puis dans le menu Run > Run module ( touche F5), sauvez votre fichier. Le code va ensuite s'exécuter et afficher quelque chose :*

*# Décrivez ce que fait ce programme et le modifier pour qu'il affiche autre chose*

```
a = 0.5
if a < 0.3 :
    print('gagné')
else :
    print('perdu')
```

## Boucle non bornée: instruction while

*Dans un autre fichier (File > New File), exécuter le code suivant, puis le corriger*

```
a = 0
while (a < 7):
    print(a)
```

## Boucle bornée: instruction for

```
for i in [1,2,3,4,5] :
    print(i)

for lettre in 'abcde' :
    print(lettre)

for n in range(5) :
```

```
print(n)
```

## Trace d'un programme

Il peut être utile de décrire le contenu des variables aux différentes étapes de l'exécution du programme, « à la main ». Nous aurons l'occasion de préciser ce que l'on appelle la trace, mais pour l'instant, avant de taper ce code, compléter le tableau ci-dessous :

```
a, b, c = 1, 1, 1
while (c < 7):
    print(b, end=" ")
    a,b,c = b, a+b, c+1
```

# Que fait ce programme ?  
# Le taper et l'exécuter ensuite

## Mots réservés

Sachez que les mots suivants ne peuvent pas être utilisés comme nom de variables, ni comme nom de fonctions :

and	as	assert	break	class	continue	def
del	elif	else	except	False	finally	for
from	global	if	import	in	is	lambda
None	nonlocal	not	or	pass	raise	return
True	try	while	with	yield		

## Opérateurs de comparaison

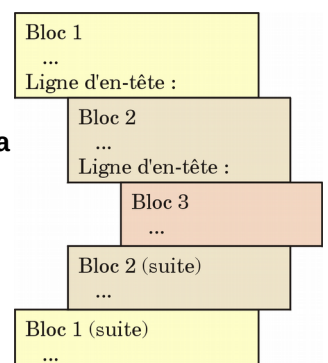
x == y	
x != y	
x > y	
x < y	
x >= y	
x <= y	

## Quelques remarques :

Dans le langage python, le caractère # désigne le début d'un commentaire. Tout ce qui suit ce caractère est ignoré par python jusqu'à la prochaine ligne. Il permet de documenter ses programmes :

```
#ceci est un commentaire
print ('ceci n'est pas un commentaire')
```

Les blocs d'instructions sont toujours associés à une ligne d'en-tête contenant une instruction bien spécifique (*if, elif, else, while, def, etc.*) se terminant par un double point. Les blocs sont délimités par l'indentation : toutes les lignes d'un même bloc doivent être indentées exactement de la même manière (c'est-à-dire décalées vers la droite d'un même nombre d'espaces). Le nombre d'espaces à utiliser pour l'indentation est quelconque, mais la plupart des programmeurs utilisent des multiples de 4.



## Premiers travaux

### Exercice 1 :

1. Ecrire un programme qui dessine le motif ci-dessous :

```
*
**
***
****
*****
```

2. Ecrire un programme qui dessine le motif ci-dessous :

```
*****
****
***
**
*
```

3. Ecrire un programme qui dessine le motif ci-dessous :

```
      *
     ***
    *****
   ********
  **********
 **********
```

### Exercice 2:

Dans un conte américain, huit petits canetons s'appellent respectivement : Jack, Kack, Lack, Mack, Nack, Oack, Pack et Qack  
En utilisant la boucle for, compléter ce programme pour qu'il affiche tous ces noms :

```
initiales = 'JKLMNOPQ'
fin = 'ack'
```

### Exercice 3:

En utilisant les instructions if et while ou for ,écrire un programme qui affiche les 20 premiers termes de la table de multiplication par 5, en signalant au passage par une astérisque ceux qui sont des multiples de 10, Exemple : 5 10\* 15 20 ...

### Exercice 4:

Une légende de l'Inde ancienne raconte que le jeu d'échecs a été inventé par un vieux sage, que son roi voulut remercier en lui affirmant qu'il lui accorderait n'importe quel cadeau en récompense. Le vieux sage demanda qu'on lui fournisse simplement un peu de riz pour ses vieux jours, et plus précisément un nombre de grains de riz suffisant pour que l'on puisse en déposer 1 seul sur la première case du jeu qu'il venait d'inventer, deux sur la suivante, quatre sur la troisième, et ainsi de suite jusqu'à la 64e case. Écrivez un programme Python qui affiche le nombre de grains à déposer sur chacune des 64 cases du jeu avec le numéro de la case.