INITIATION PYTHON 1/2

```
1  a = 2
2  b = 3
3  c = a+b
4  print(c)
```

- Tapez ce programme et verifier son bon fonctionnement
- Modifiez-le avec 3 variables *a, b, c* (valeur 3,5,7). La variable *d* contient l'addition des 3 variable *a, b, c*. Affichez la variable *d*

1

2

Entrée utilisateur

L'intérêt d'un programme est de permettre une interactivité avec les utilisateurs. Dans notre programme précédent les valeurs étaient 'codées en dur' et l'utilisateur ne pouvait les changer.
la commande input(« texte que voit l'utilisateur ») permet de demander des valeurs à l' utilisateur :

```
1  a = int(input("donnez la valeur de a"))
2  b = 3
3  c = a+b
4  print(c)
```

Types de données

Comme tous les langages de programmation, Python accepte plusieurs type de données. Afin d'éviter une mauvaise interprétation il est possible de préciser le type d'une variable, comme nous l'avons vu dans l'exemple plus haut.
Voici quelques types courants :
int : nombre entier. Par exemple : a = 3
float : nombre décimal; Par exemple : b = 2.25 (notez le point comme séparateur décimal)
str : suite de caractères. Par exemple : c = « mon texte » (les doubles guillemets au début et à la fin)
list : liste de variables (de n'importe quel type). Par exemple : d = [1,2,3] (crochets au début et à la fin et virgule entre les valeurs)

3

Valeur RVB d'un Pixel

Rependre le programme permettent de donner la valeur RVB d'un Pixel en demandant à l'utilisateur la valeur X et Y du pixel.

4

Structures de test

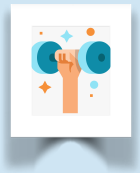
Dans un programme il faut souvent tester si une condition est remplie avant d'exécuter d'autres parties du programme. Un test consiste à comparer la valeur de deux variables (ou une variable avec une valeur prédéfinie).
Cela se traduit par : Si (quelque chose) Alors (faire 1) Sinon (faire 2)

```
1  condition = 1
2  remplie = 1
3  if condition == remplie :
4      print("1")
5  else :
6      print("2")
```

INITIATION PYTHON 2/2

5

Exercices structures de test



- Écrivez un programme qui demande l'âge de l'utilisateur et vérifie s'il est majeur ou pas.

- Écrivez un programme qui demande un code de déverrouillage de téléphone et le compare à une valeur stockée dans une variable « password » et répond en conséquence.

6



Boucles

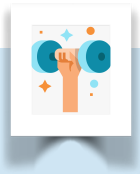
Les boucles permettent de répéter une ou plusieurs commandes plusieurs fois. Il y en a deux sortes :
Boucle bornée : on sait combien de fois la boucle doit s'exécuter.
Boucle non bornée : la fin de la boucle est déterminée par une condition.

```
1 for i in range(4):
2     print(i)
```

- le décompte va de 0 à 3, ce qui fait bien 4 passages de boucles : on commence toujours ces boucles avec la valeur 0 !!

- Notez les doubles points après la définition des règles de la boucle et la tabulation pour les instructions suivantes.

7



Exercices boucle

- Écrivez un programme qui balaye la ligne 150 de l'image pêche.jpg et donne toutes les valeurs des canaux R,V,B de cette ligne. (L'image fait 500 sur 500 pixels)

- Demandez à l'utilisateur quelle ligne il veut analyser puis afficher les valeurs des canal R, V, B de cette ligne.

Ligne 150

