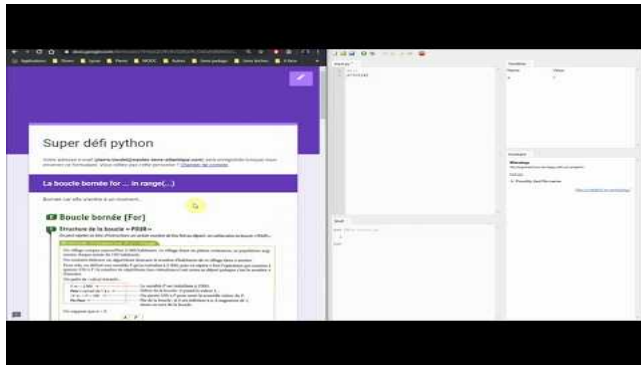


Super défi python

Bonjour à tous, je teste le formulaire autonome. Donc si cela ne marche pas ne vous jetez pas par la fenêtre du rez-de-chaussée ;-)) et faites vos exercices de maths ;-)
Moi je suis dans un train qui me remonte de Sète...

Votre adresse e-mail (pierre.troulet@nantes-terre-atlantique.com) sera enregistrée lorsque vous enverrez ce formulaire. Vous n'êtes pas pierre.troulet ? [Se déconnecter](#)



<http://youtube.com/watch?v=2XJdpoVDqzg>

Python et affectation de valeurs dans des variables

A faire avec Thonny

A Algorithmes – Variables – Affectation

1 Algorithmes et notion de variable

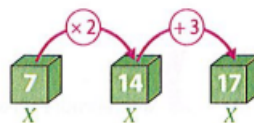
Un algorithme est une suite finie d'instructions à appliquer dans un ordre déterminé à un nombre fini de données pour arriver, en un nombre fini d'étapes, à un certain résultat.

Pour stocker un résultat, on utilise une variable. On peut se représenter une variable comme une « boîte », un emplacement de la mémoire d'un ordinateur... Pour pouvoir accéder à son contenu, on lui donne un nom.

Exemple 1: un programme de calcul

Choisir le nombre 7
Multiplier ce nombre par 2
Ajouter 3 au résultat

On peut ici utiliser une variable, que l'on nomme X , qui contient les résultats obtenus à la fin de chaque étape du programme de calcul.



Une variable est désignée par un nom. Elle contient une valeur. On utilise quatre types de valeurs :

- entier (nombre entier relatif) ;
- flottant (nombre à virgule) ;
- chaîne de caractères : suite ordonnée de caractères, un caractère étant un chiffre, une lettre, un symbole...
- booléen : variable qui ne prend que deux valeurs (Vrai ou Faux), sa valeur est en général donnée par un test.

Remarques :

- 1) "Année" est une chaîne de cinq caractères. On dit que c'est une chaîne de longueur 5.
Le premier caractère de cette chaîne est A, son deuxième caractère est n, son troisième est n...
- 2) Les chaînes de caractères peuvent « s'ajouter », c'est-à-dire se mettre bout-à-bout (on parle aussi de concaténation de chaînes de caractères). Par exemple, "Année" + "2017" = "Année2017".
- 3) La variable b contenant le test $(6 > 4)$ est de type booléen, sa valeur est Vrai.

2 L'affectation

Lorsque l'on donne une valeur à une variable X , on écrit l'instruction : $X \leftarrow \dots$

On lit : « X reçoit \dots » ou « X prend la valeur \dots »

La nouvelle valeur remplace la valeur précédente.

Algorithme correspondant au programme de calcul de l'exemple 1 :

La variable X contient d'abord 7.

$X \leftarrow 7$

La valeur de X est ensuite multipliée par 2 : elle devient donc $2X$.

$X \leftarrow 2X$

Elle est enfin augmentée de 3 : elle devient donc $X + 3$.

$X \leftarrow X + 3$

Exemple 2 : un autre algorithme

Algorithme :

$A \leftarrow 3$

$B \leftarrow A + 1$

$A \leftarrow A + B$

A
3

B
3

3

4

7

4

Valeur de la variable A et valeur de la variable B
après l'exécution de chaque instruction :

← La valeur de A est 3 et B n'a pas encore de valeur.

← Comme la valeur de A est 3, la valeur de B est : $3 + 1$, soit 4.
La valeur de A ne change pas : elle reste égale à 3.

← Comme la valeur de A est 3 et que celle de B est 4, la nouvelle valeur
de A est : $3 + 4$, c'est-à-dire 7.
La valeur de B ne change pas : elle reste égale à 4.

1- Saisie à faire sous Thonny

Thonny - C:\Users\Pierre\Downloads\test.py @ 11:12

File Edit View Run Device Tools Help

test.py x

```

1 i=2
2 j=i*i
3
4 a=i**i
5 b=i**i**i
6 c=i**i**i**i
7
8 print(i,j)
9 print("a=",a)
10 print("b=",b)
11 print("c=",c)

```

Variables x

Name	Value
a	4
b	16
c	65536
i	2
j	4

Shell x

```

>>>

```

Assistant x

Warnings
May be ignored if you are happy with your program.

[test.py](#)

Possibly bad file name

[Was it helpful or confusing?](#)

1. 1.1- QCM sur le script du dessus*Une seule réponse possible par ligne.*

	Vrai	Faux	Peut-être, il faut que j'en parle avant à mon chien savant
Pour ne pas tenir compte d'une ligne dans un script il suffit de mettre au tout début le symbole #?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Si je remplace <code>i=2</code> par <code>#i=2</code> alors j'aurai une erreur en lançant le script?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>j</code> et <code>a</code> sont égaux mais pas pour les mêmes raisons?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La valeur de <code>c</code> est égale à 2 puissance 16 donc <code>2*2*2*2...*2</code> et ceci 16 fois?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Une <code>*</code> c'est pour multiplier, mais deux <code>*</code> (<code>**</code>) c'est pour élever au carré?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Si j'ajoute ligne 12 du script <code>print("d=",i***i***i)**i)</code> alors j'hallucine ;)?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. 1.2- Question pour que ce soit bien compris

$$2^{2^{2^2}} = 2^{2^{2^4}} = 2^{2^{16}} = 2^{65536}$$

Une seule réponse possible.

- Vrai
 Faux

3. 1.3- Question pour que ce soit bien compris

$$2^{2^{2^{2^2}}} = 4^{2^{2^2}} = 16^{2^2} = 256^2$$

Une seule réponse possible.

- Vrai
 Faux

4. 1.4- Pour répondre à la 12 et la 13 j'ai utilisé la règle des priorités des opérations en maths qui est :

https://www.youtube.com/watch?v=d_Qu4bWSuA8

Une seule réponse possible.

- je commence par parenthèse /puissance / produit / somme
 je commence par produit / somme / puissance / parenthèse
 je regarde mon horoscope puis je me lance au hasard si Saturne est dans le Sagitaire ;))

5. Bilan de cette première section

Développer ce que vous avez compris sur les valeurs et les variables, l'affichage et les opérateurs de base (+, *, / et **). Ce bilan n'apporte aucun point! Au fait

La boucle bornée for ... in range(...)

Bornée car elle s'arrête à un moment...

E Boucle bornée (For)

1 Structure de la boucle « POUR »

On peut répéter un bloc d'instructions un certain nombre de fois fixé au départ : on utilise alors la boucle « POUR ».

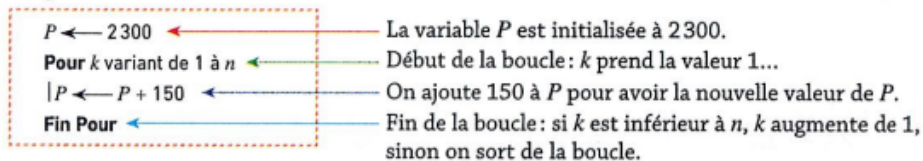
Exemple : croissance d'un village

Un village compte aujourd'hui 2 300 habitants. Le village étant en pleine croissance, sa population augmente chaque année de 150 habitants.

On souhaite élaborer un algorithme donnant le nombre d'habitants de ce village dans n années.

Pour cela, on définit une variable P qu'on initialise à 2 300, puis on répète n fois l'opération qui consiste à ajouter 150 à P : le nombre de répétitions (ou « itérations ») est *connu au départ* puisque c'est le nombre n d'années.

On parle de « calcul itératif ».

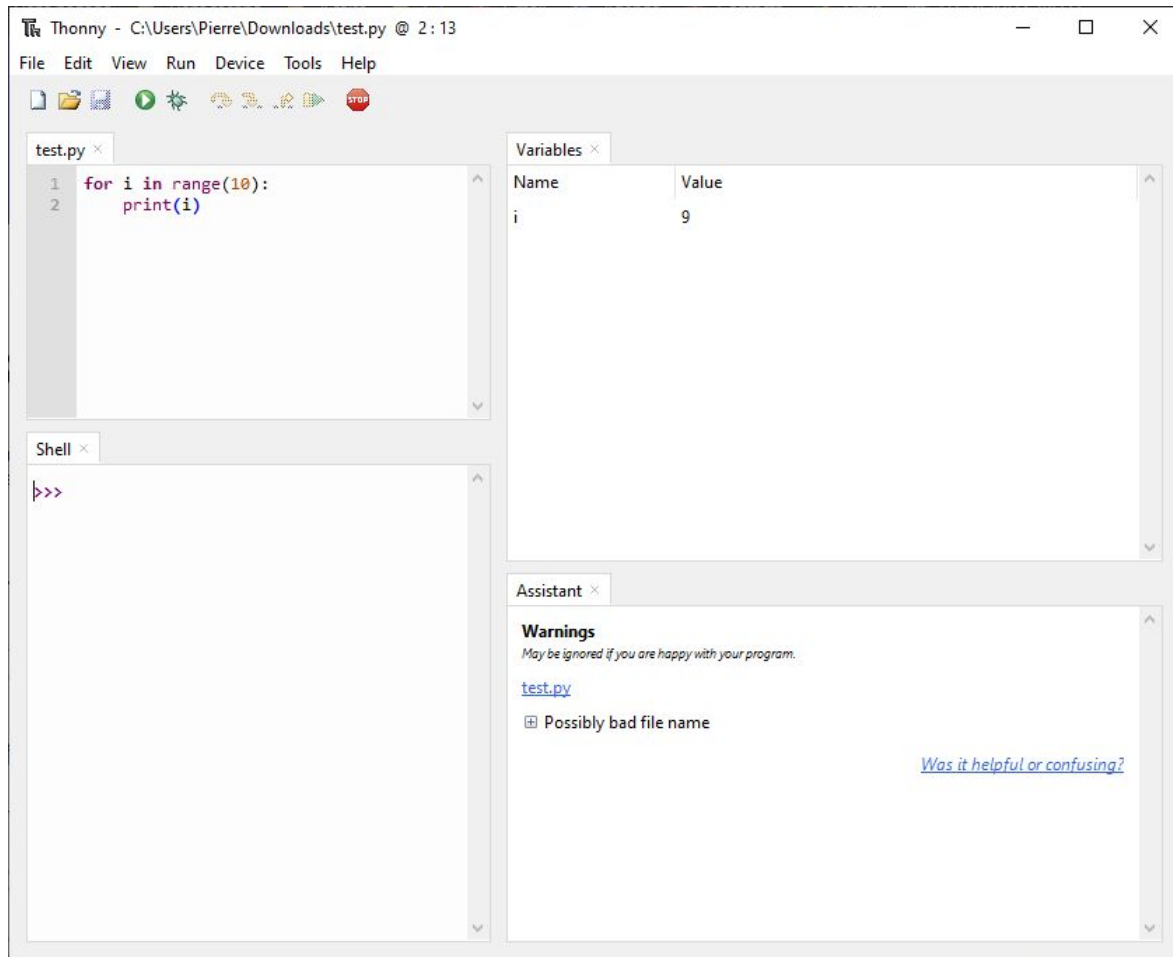


On suppose que $n = 3$.

	k	P	
Avant le début de la boucle		2 300	← Au début, la variable P est initialisée à 2 300.
Fin de la 1 ^{re} itération	1	2 450	← $k = 1$ et on ajoute 150 à P donc $P = 2\ 300 + 150 = 2\ 450$.
Fin de la 2 ^e itération	2	2 600	← $k = 2$ et on ajoute 150 à P donc $P = 2\ 450 + 150 = 2\ 600$.
Fin de la 3 ^e itération	3	2 750	← $k = 3$ et on ajoute 150 à P donc $P = 2\ 600 + 150 = 2\ 750$.

La variable k contrôle le nombre de répétitions. Dans l'exemple ci-dessus, k prend pour valeur initiale 1 et pour valeur finale n en augmentant de 1 à chaque fois (cette augmentation constante est appelée « pas »).

2- Saisie à faire dans Tonny



6. 2.1- Boucle for

Une seule réponse possible par ligne.

	Vrai	Faux	Peut-être, mais il faut avant que je sorte mon chien
Par défaut, la boucle for part de 0 et attend 10 en augmentant la variable de + 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Le print(i) est exécuté 10 fois et j'ai 10 lignes dans le shell	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La variable i dans cette boucle va prendre la valeur 0 puis 1 puis ...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Le décalage de 4 espaces au début de la ligne 2 c'est pour faire joli	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Si j'ajoute dans la boucle en ligne 3 print(2*i) alors je vois une liste de nombres pairs	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Le mieux serait en fait pour voir la liste des nombres pair de remplacer print(i) de la ligne 2 par print(i,"ième pair ",2i)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Avec un for, rien d'impossible

2.2.1- Une copie d'écran pour vous aider... au départ i ne vaut rien...

```

test.py ×
1  i=""
2  print(i)
3  i=i+"*"
4  print(i)
5  i=i+"*"
6  print(i)
7  i=i+"*"
8  print(i)

Shell ×
>>> %Run test.py

*
**
***

>>> |

```

7. 2.2.2- Avec un for et deux variables écrivez un script qui affiche 10 lignes d'étoiles

Votre réponse c'est le coller de votre script!

```

>>> %Run test.py

*
**
***
****
*****
*****
*****
*****
*****
*****
*****
*****
*****

>>> |

```

Je veux savoir...

combien d = 2**2**2**2**2 à de chiffre, pas vous?

2.2.3- A saisir

The screenshot shows a Python IDE with three main panels:

- Code Editor:** Contains a Python script:

```
1 i=2
2 d=i**i**i**i**i
3
4 # je transforme d en lettre
5 lettre=str(d)
6 exp=len(lettre)
7 print ("d est un nombre contenant ",exp," chiffres")
```
- Shell:** Shows the prompt `>>> |`.
- Variables:** A table showing the current state of variables:

Name	Value
d	2003529930406846464979072351560255750
exp	19729
i	2
lettre	'2003529930406846464979072351560255750'
- Assistant:** Shows a warning:

Warnings
May be ignored if you are happy with your program.
[test.py](#)
Possibly bad file name
[Was it helpful or confusing?](#)

8. 2.3.2- d est un nombre de :

Une seule réponse possible.

- 19730 chiffres
- 19729 chiffres
- 19728 chiffres
- j'ai pas eu le temps de les compter

Les fonctions

C Les fonctions

On peut simplifier l'écriture des programmes en utilisant des **fonctions**, sur le modèle des fonctions numériques étudiées en mathématiques.

1 Exemple de fonctions

Exemple 1: calcul d'une distance d'arrêt

La distance d'arrêt d (en mètres) d'un véhicule roulant à une vitesse v (exprimée en km.h^{-1}) sur une route sèche peut être évaluée par la formule : $d = 0,005v^2 + 0,278v$.

La distance d dépend de la variable v . Pour programmer ce calcul, on va ainsi définir une fonction, comme en mathématiques : on lui donne un nom, **arrêt** et on dit qu'elle a pour **argument** v .

En langage Python, on définit cette fonction par la commande **def** suivi du nom de la fonction, puis de son argument v , suivi de deux points.

Les autres instructions sont **indentées** par rapport à la première ligne.

Après le calcul de d , on renvoie le résultat par la commande **return**.

Pour calculer la distance d'arrêt d'un véhicule roulant à 90 km.h^{-1} , on saisit **arrêt** (90) dans la console : on trouve environ 65,5 m.

```
def arrêt(v):
    d=0.005*v**2+0.278*v
    return(d)
```

↑ indentation

deux points

Exemple 2: calcul de la surface de la peau

La surface S en mètres carrés de la peau d'un adulte est donnée approximativement par la formule :

$S = \frac{\sqrt{L \times M}}{6}$, où L est la taille de la personne exprimée en mètres et M sa masse exprimée en kilogrammes.

Pour programmer ce calcul, on définit une fonction appelée **surf** qui a ici deux **arguments** L et M et qui retourne S , celle-ci sera écrite **surf(L,M)** dans le programme.

Pour calculer la surface corporelle d'un adulte de 1,80 m et 75 kg, on saisit **surf**(1.8,75) dans la console : on trouve environ 1,94 m^2 .

```
def surf(L,M):
    return(sqrt(L*M)/6)
```

2 Définition – Programmation

Une fonction est un bloc d'instructions qui a reçu un nom et dont le fonctionnement dépend d'un certain nombre de paramètres (les **arguments** de la fonction). La fonction renvoie un résultat (au moyen de la commande **return**) ; le programme s'arrête après **return**.

Programmation d'une fonction en langage Python

• La programmation d'une fonction commence toujours par **def**, suivi du nom que l'on donne à la fonction, suivi des arguments de la fonction : **cette ligne se termine par deux points (;)**.

• Les deux points marquent le démarrage du bloc d'instructions définissant la fonction : **toutes ces instructions sont indentées**, c'est-à-dire décalées vers la droite par rapport à la première ligne.

On ajoute en tête de chaque ligne du bloc le même nombre d'espaces.

• Pour renvoyer la valeur de la variable d , on peut écrire : **return(d)** ou **return d**.

• On peut utiliser une fonction dans la console en donnant des valeurs aux arguments (dans le bon ordre).

Propriétés

• Une fonction ne renvoie qu'un seul résultat (ce peut être une liste de résultats).

• Une fonction peut n'avoir aucun argument.

Par exemple, la fonction **essai()** ci-contre peut être utilisée dans un programme où on résout des équations.

• Une fonction peut être appelée dans un autre programme : il suffit pour cela de l'insérer dans une instruction en saisissant son nom et les valeurs des arguments.

```
def nom_fonction(liste des arguments):
    bloc d'instructions
    return(résultat)
```

```
def essai():
    return("il y a une unique solution")
```

3- A saisir sous Thonny


```

test.py x
1  def racine(x):
2      f=x**0.5
3      return(f)
4  for x in range (10):
5      print(x,"-",racine(x))

Shell x
|>>>

```

9. 3.1- Répondre

Une seule réponse possible par ligne.

	Vrai	Faux	A zut, il faut que je donne à manger à mon chien car il a faim
$x^{**0.5}$ quand $x=4$ donne la racine de 4 qui est égale à 2.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
L'argument (variable) de la fonction "racine" est x	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La boucle for fait varier x de 0 à 9 tous les 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La ligne 5 me permet grâce au for de calculer 10 couples de valeurs (x,racine(x)) comme en maths	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10. 3.2- Ai-je un chien?

Une copie de vos réponses sera envoyée par e-mail à pierre.troulet@nantes-terre-atlantique.com.

Fourni par

