

Cours - Thème: Python - Programme encadrer la racine de 2

Lien vers une interface de programmation en ligne

<https://trinket.io/python3>

Lien vers un site permettant d'installer une interface de programmation

<https://thonny.org/>

QrCode pour la prise en main de Thonny →



Objectif

Créer un programme en python permettant d'encadrer la racine carrée de 2

$\sqrt{2} \sim 1.414\dots$

Lien vers l'article présentant toute la démarche:

<https://ipa-troulet.fr/cours/index.php/cours-dinformatique/2nde-snt/135-snt-2nde-python/481-2nde-snt-python-encadrement-d-une-racine-carree?start=1>

Table des matières

[1 Logiciels - application web / logiciel / application android](#)

[2 Rappel des commandes principales](#)

[3 Le principe de Héron d'Alexandrie](#)

[3.1 Application du du principe pour encadrer racine de 2](#)

[3.2 Questions](#)

[4 Mon premier programme en python](#)

[4.1 Ecrire un algorithme avant de programmer](#)

[4.2 Comprendre quelques principes en python](#)

[4.2.1 L'affectation de valeurs dans les variables et la pile d'enregistrement](#)

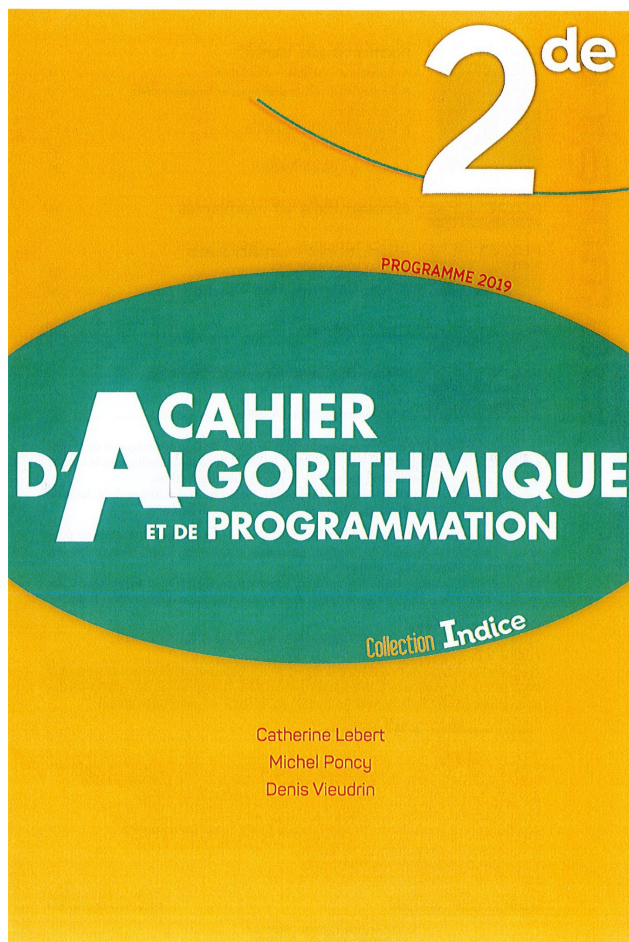
[4.2.2 La boucle bornée For in range\(...\):](#)

[4.2.3 De l'importance de la position](#)

[4.3 Écrire mon premier programme](#)

1 Logiciels - application web / logiciel / application android

- Tutos vidéo - Installation IDE Thonny + prise en main + affectation de variable
[Site Troulet](#)
- IDE en ligne
<https://trinket.io/python3>
(testé en cours sur les smartphones des élèves mais pas probant du tout - Par contre pour que les élèves qui n'installent pas Thonny... à tester)
- IDE pour android via le playstore QPython
- Utilisation de votre cahier d'algorithmique et de programmation



2 Rappel des commandes principales

Programmation en PYTHON



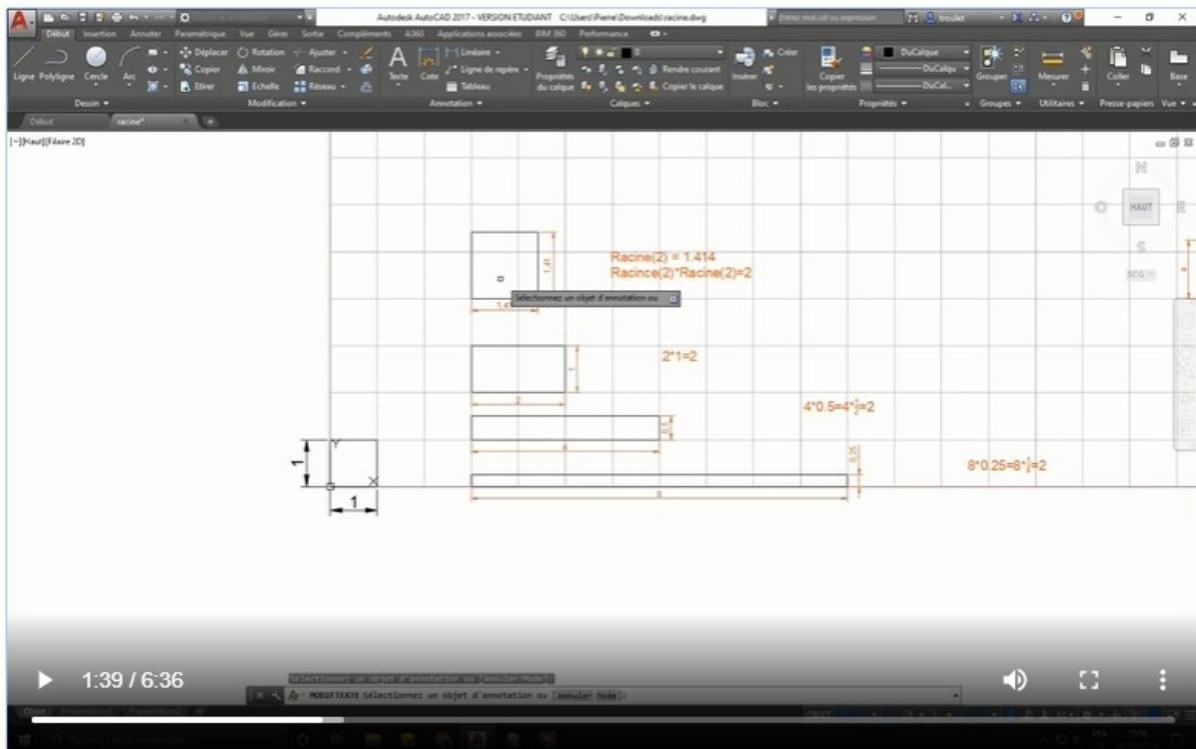
Créer un nouveau programme	On saisit le programme dans l'éditeur.	
	On va à la ligne après chaque instruction, mais on peut taper plusieurs instructions sur la même ligne en les séparant par un point-virgule.	
Importer un module	<code>from nom_du_module import*</code>	<code>from lycee import*</code> avec EduPython pour la bibliothèque lycée
Afficher <i>A</i>	<code>print(A)</code>	
Affecter à <i>B</i> la valeur de <i>A</i>	<code>B=A</code>	
Écrire des commentaires	On écrit les commentaires sur une ligne, précédés de #	
Tester si $A = B$ / si $A \neq B$	<code>A==B</code>	<code>A!=B</code>
Tester si $A \geq B$ / si $A \leq B$	<code>A>=B</code>	<code>A<=B</code>
<i>A</i> et <i>B</i> / <i>A</i> ou <i>B</i>	<code>A and B</code>	<code>A or B</code>
Si {condition C} Alors {instructions A} Sinon {instructions B} Fin Si	<code>if</code> {condition C}: {instructions A} <code>else</code> : {instructions B}	Il n'y a pas d'instruction de fin : c'est l'indentation (le décalage vers la droite) qui indique les instructions faisant partie de la structure conditionnelle.
Pour <i>i</i> variant de 1 à <i>n</i> {instructions} Fin Pour	<code>for i in range(1,n+1):</code> {instructions}	<ul style="list-style-type: none"> L'instruction <code>for i in range(n)</code> fait parcourir à la variable <i>i</i> tous les entiers de 0 à <i>n</i> - 1. L'instruction <code>for i in range(n₀,n)</code> fait parcourir à la variable <i>i</i> tous les entiers de <i>n</i>₀ à <i>n</i> - 1. L'instruction <code>for i in range(n₀,n,p)</code> fait parcourir à la variable <i>i</i> les entiers de <i>n</i>₀ jusqu'à l'entier immédiatement inférieur ou égal à <i>n</i> - 1 avec un pas de <i>p</i>. Comme pour <code>if</code> , on utilise l'indentation pour indiquer les instructions faisant partie de la boucle.
Tant que {condition C} {instructions} Fin Tant que	<code>while</code> {condition C}: {instructions}	Comme pour <code>if</code> et <code>for</code> , on utilise l'indentation pour indiquer les instructions faisant partie de la boucle.
Définition d'une fonction <i>f</i>	<code>def f(a,b,c,...):</code> <code>y=</code> <code>return(y)</code>	On peut aussi écrire : <code>return y</code> au lieu de <code>return(y)</code>
Racine carrée de <i>x</i>	<code>sqrt(x)</code> dans la bibliothèque lycée ou le module math	
<i>x</i> à la puissance <i>n</i>	<code>x**n</code>	
π	<code>pi</code> dans la bibliothèque lycée ou le module math	
Reste de la division de <i>a</i> par <i>b</i>	<code>a%b</code> avec la bibliothèque lycée	<code>fmod(a,b)</code> pour la calculatrice NumWorks
Nombre décimal aléatoire compris entre <i>a</i> et <i>b</i>	<code>uniform(a,b)</code> avec la bibliothèque lycée ou le module random	
Entier aléatoire compris entre <i>a</i> et <i>b</i>	<code>randint(a,b)</code> avec la bibliothèque lycée ou le module random	
Longueur d'une chaîne de caractères <i>C</i> / Extraire un caractère de <i>C</i>	<code>len(C)</code>	<code>C[k]</code> renvoie le (k + 1) ^e élément de <i>C</i>
Concaténation de deux chaînes de caractères	"abcd" + "efg" donne "abcdefg"	
Booléens	<code>True</code> (Vrai)	<code>False</code> (Faux)
Exécuter un programme	Onglet Exécuter	EduPython : cliquer sur ►

3 Le principe de Héron d'Alexandrie

1>> Sur votre blog créer un article présentant Héron d'Alexandrie (un paragraphe et une image)

Suivre le lien du dessous pour visualiser l'idée de Héron

Thème Transversal - Python - Programmation - TD racine de 2



<https://ipa-troulet.fr/cours/attachments/article/481/01%20encadrement%20racine.mp4>

3.1 Application du du principe pour encadrer racine de 2

Soit un rectangle de surface $s = 2$

$$s = a * b$$

Etape 1:

Choisir une valeur de a arbitraire grande (par exemple 8)

$$a_1 = 8$$

En déduire b

$$b_1 = s / a_1 = \dots\dots\dots$$

Dessiner le rectangle avec a_1 et b_1

Etape 2:

$$a_2 = (a_1 + b_1) / 2 = \dots\dots\dots$$

En déduire b_2

$$b_2 = s / a_2 = \dots\dots\dots$$

Dessiner le rectangle avec a_2 et b_2

Etape 3:

Compléter et calculer

$$a_3 = \dots\dots\dots$$

En déduire b_3

$$b_3 = \dots\dots\dots$$

Dessiner le rectangle avec a_3 et b_3

Thème Transversal - Python - Programmation - TD racine de 2

Etape 4:

Compléter et calculer

$a_4 = \dots\dots\dots$

En déduire b_4

$b_4 = \dots\dots\dots$

Dessiner le rectangle avec a_4 et b_4

On s'arrêtera là car je nous n'avons pas de microscope et de crayon assez fin

3.2 Questions

Compléter le tableau suivant :

2>

Etape i	Valeur de b_i	valeur de a_i
Etape 1	$b_1 =$	$a_1 =$
Etape 2		
Etape 3		
Etape 4		

Classer les valeurs des b_i par ordre croissant comme ceci ... $< b_1 < b_2 < \dots$

3>

Classer les valeurs des a_i par ordre croissant comme ceci ... $< a_1 < a_2 < \dots$

4>

A votre avis ai-je raison de penser que pour i étant de plus en plus grand alors les valeurs de a_i se rapprochent de plus en plus de celles de b_i ?

5>

A votre avis ai-je raison de penser que pour i étant de plus en plus grand alors la forme géométrique ressemble de plus en plus à un carré ?

6>

Le carré vous connaissez ?

Si la surface d'un carré vaut 2, que vaut son côté ?

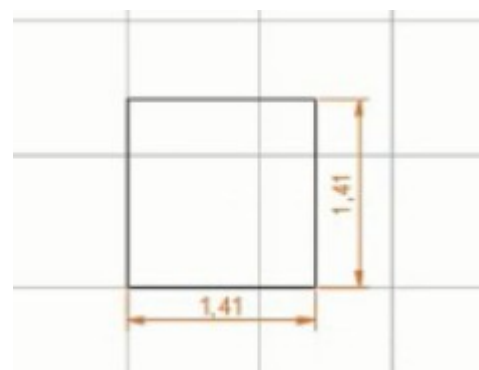
(Symbole à copier dans votre formule si besoin $\sqrt{\quad}$)

7>

A cette étape du cours, je suis entrain de penser à l'enseignement des suites en mathématiques, pourquoi ?

Si vous n'avez pas encore commencé les suites, alors laissez tomber.

8>



4 Mon premier programme en python

4.1 Ecrire un algorithme avant de programmer

L'idée est d'écrire une ligne pour chacune des étapes dans le genre:

Mettre la valeur 3 dans la variable a	$a \leftarrow 3$ ou $a = 3$
Afficher la valeur de a	affiche a ou <code>print (a)</code>
Prendre la valeur de i et lui ajouter 1 puis mettre le résultat dans i	$i \leftarrow i + 1$ ou $i = i + 1$
Mettre le résultat de $a / 5$ dans c	$c \leftarrow a / 5$ ou $c = a / 5$
Pour i allant de 0 à 10 exclu	Boucle i de [0 à 10[: Fin de la boucle
Commentaire qui n'est pas une ligne de programme	#

Vous venez de calculer des valeurs successives de a_i et b_i . Présenter ligne par ligne chacune de vos instructions en insérant quelques commentaires.

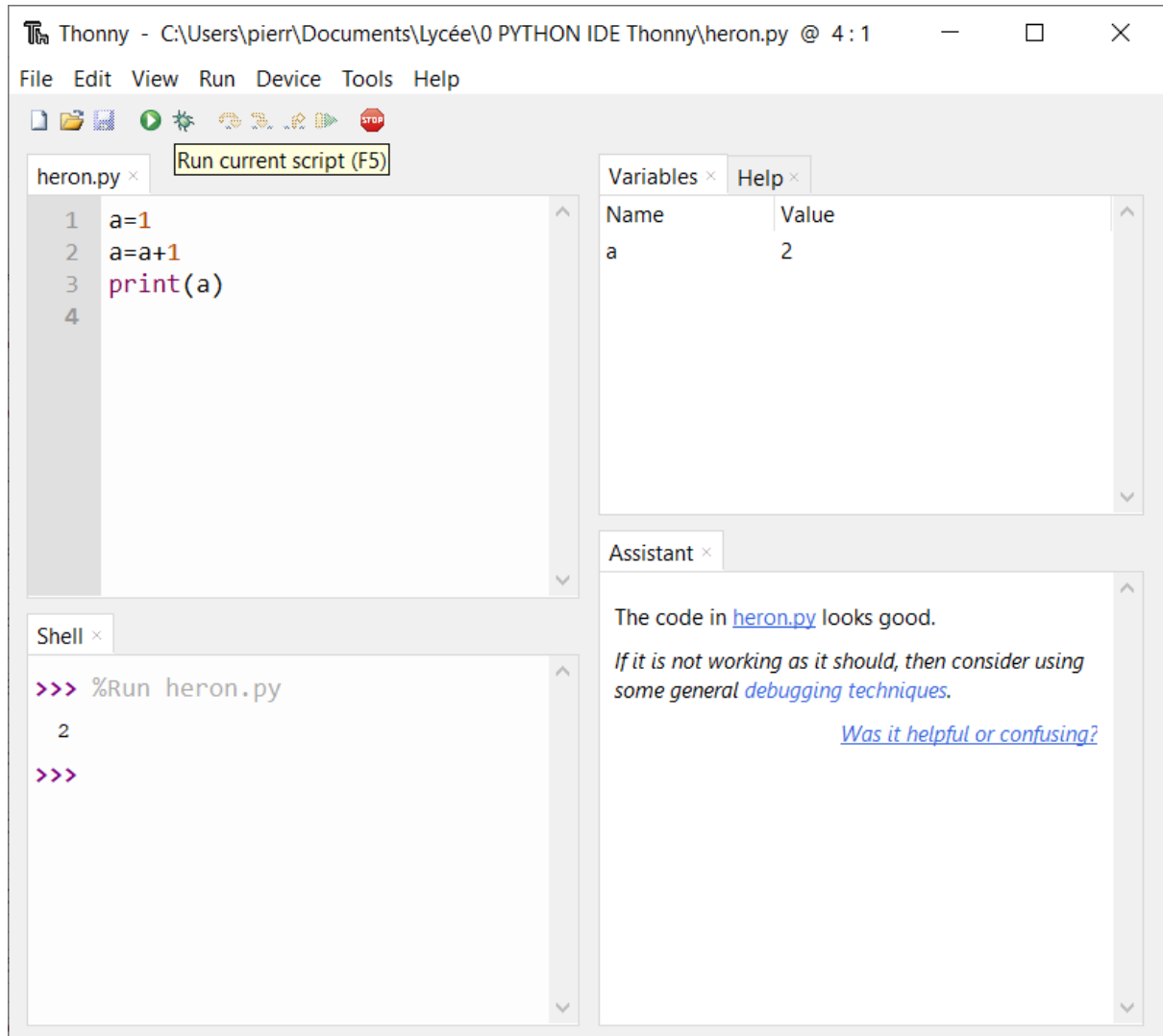
9>

```
# constante
Mettre 2 dans s
# initialisation des variables – etape 1
# j'ai besoin de a1, b1
a1 = 8
b1 = s / a1
# etape 2
...
...
# etape 3
...
...
```

Thème Transversal - Python - Programmation - TD racine de 2

4.2 Comprendre quelques principes en python

4.2.1 L'affectation de valeurs dans les variables et la pile d'enregistrement



Expliquez ligne par ligne ce qu'il se passe:

10> ligne 1

11> ligne 2

12> ligne 3

13> ligne 4 ;)

Expliquez pourquoi un prof de maths s'arrache les cheveux en lisant $a = a + 1$?

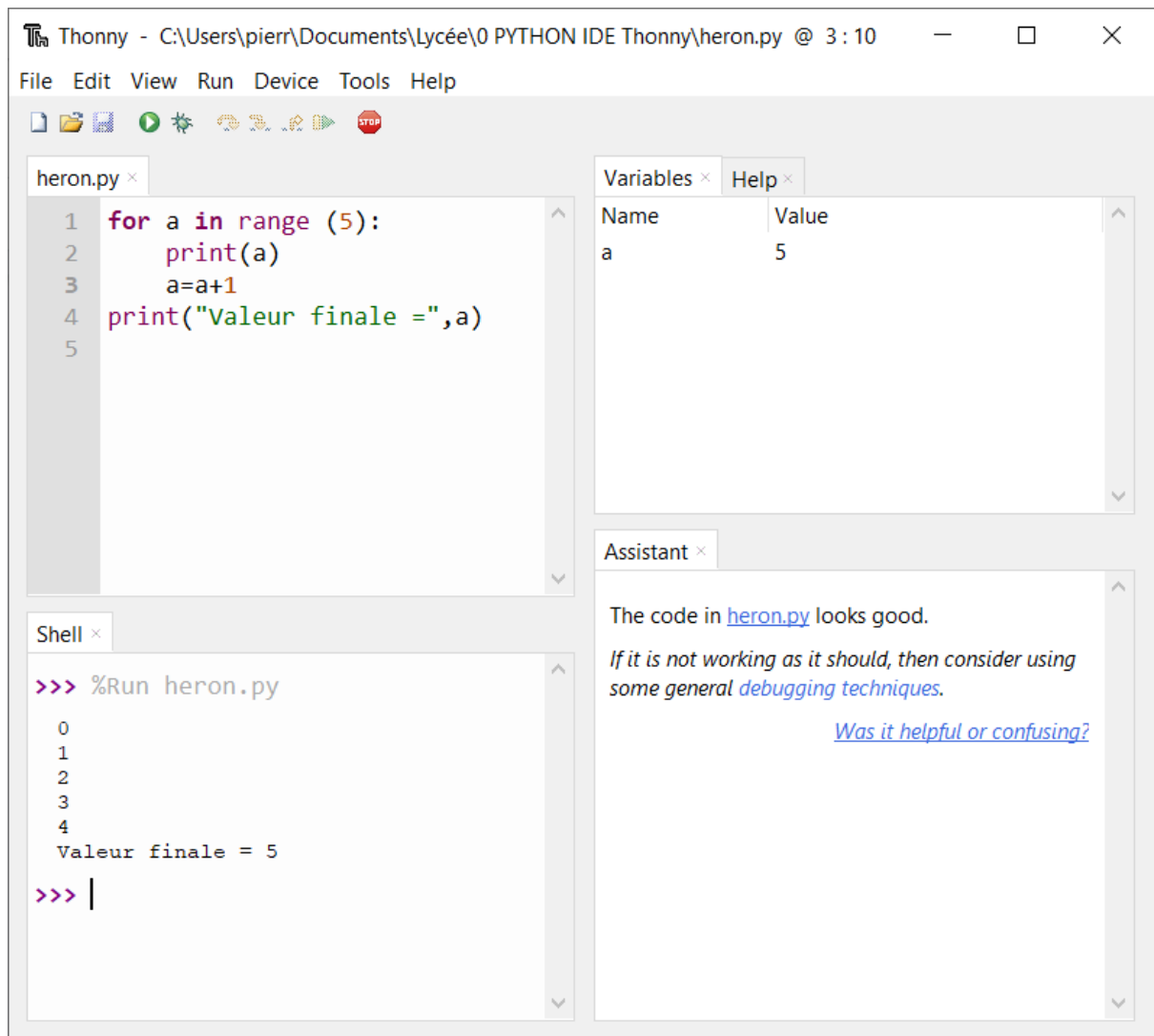
14>

Expliquez pourquoi dans un algorithme on préfère noter $a \leftarrow a + 1$

15>

Thème Transversal - Python - Programmation - TD racine de 2

4.2.2 La boucle bornée For in range(...) :



The screenshot shows the Thonny Python IDE interface. The main editor window displays the following Python code in `heron.py`:

```
1 for a in range (5):
2     print(a)
3     a=a+1
4 print("Valeur finale =",a)
5
```

The Shell window shows the execution output:

```
>>> %Run heron.py
0
1
2
3
4
Valeur finale = 5
>>> |
```

The Assistant panel provides feedback:

The code in [heron.py](#) looks good.
If it is not working as it should, then consider using some general [debugging techniques](#).
[Was it helpful or confusing?](#)

The Variables panel shows the current state of the program:

Name	Value
a	5

Dans la boucle for a in range (5)...

Quelle est la première valeur de a?

16>

Combien de boucles fait la boucle?

17>

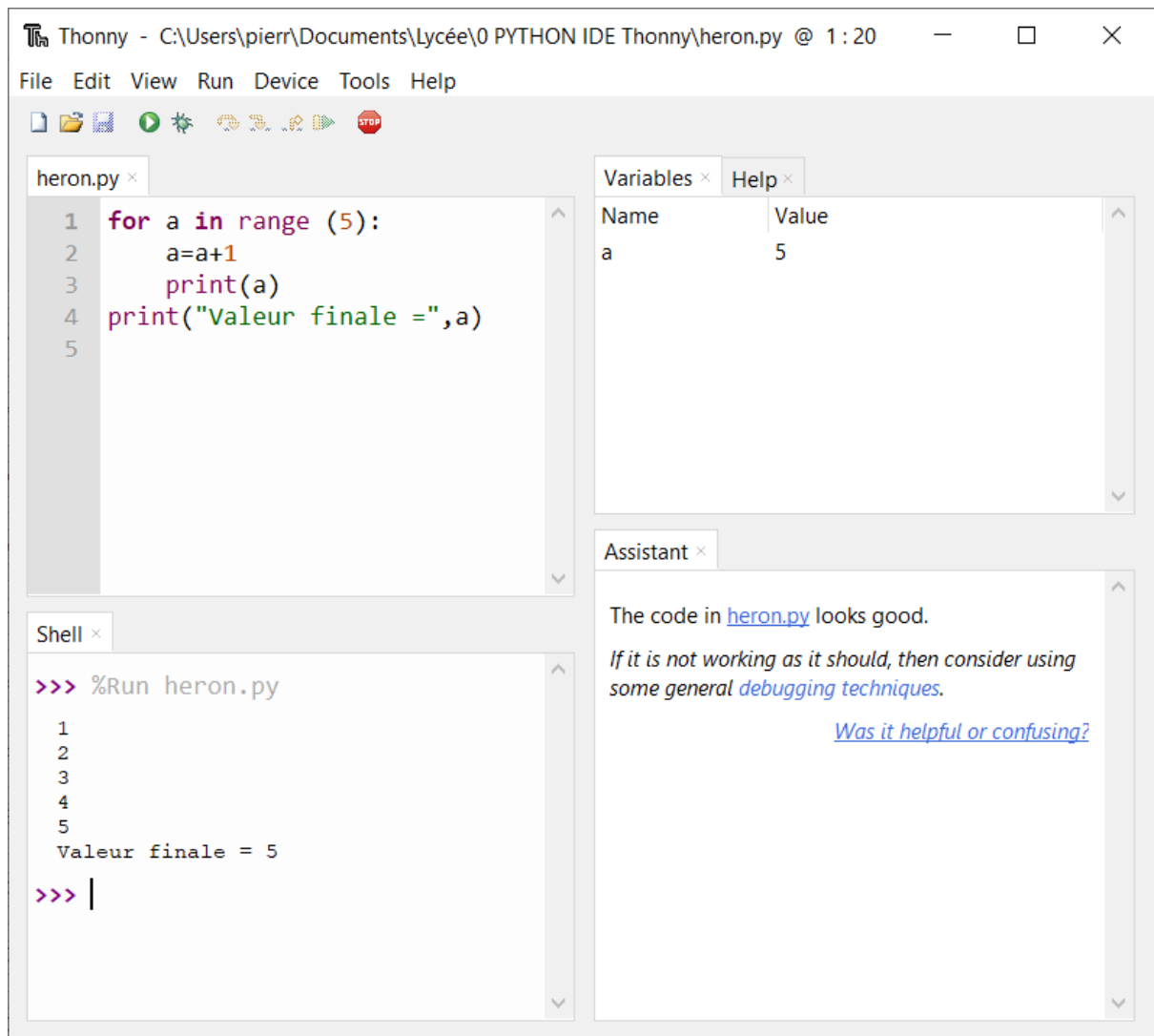
Python compte bien 5 boucles car a ira de 0 à ???

18>

Le `print("Valeur finale =",a)` de la ligne 4 n'est pas dans la boucle. Pourquoi ici il me dit que la valeur de a = 5?

19>

4.2.3 De l'importance de la position



Comparez les deux copies d'écran et comprenez ;)

4.3 Écrire mon premier programme

Reprendre votre premier algorithme et regrouper dans une boucle les calculs successifs des a et des b...

20>

Ecrire votre programme en python sous Thonny sur un ordi ou sur votre calculatrice (vous faites une photo de l'écran) ou en ligne sur trinket ou sur votre téléphone avec qpython (vous en ferez une photo avec un autre téléphone - c'est dingue ce qu'on écrit au XXI ème siècle hein ;) puis vous insérerez cette preuve en dessous.

Thème Transversal - Python - Programmation - TD racine de 2

The screenshot shows the Thonny Python IDE interface. The main editor window displays a Python script named `heron.py` with the following code:

```
1 #initialisation
2 a=4
3 s=2
4 b=s/a
5
6 #boucle étape successive
7 for i in range (5):
8     a=(a+b)/2
9     b=s/a
10    print ("étape ",i," val a=",a," val b=",b)
11
12
```

The Shell window at the bottom left shows the prompt `>>>`. The Variables window on the right shows the current state of variables:

Name	Value
a	1.414244175296705
b	1.4141829501121366
i	4
s	2

The Assistant window on the right provides feedback: "The code in `heron.py` looks good. If it is not working as it should, then consider using some general debugging techniques. [Was it helpful or confusing?](#)"

The screenshot shows a terminal window with the following output:

```
>>> %Run test.py
étape 0 val a= 4.125 val b= 0.48484848484848486
étape 1 val a= 2.3049242424242424 val b= 0.8677074774034512
étape 2 val a= 1.5863158699138467 val b= 1.260782956622914
étape 3 val a= 1.4235494082683804 val b= 1.4049389423250296
étape 4 val a= 1.414244175296705 val b= 1.4141829501121366
>>>
```