



## I) Découverte du logiciel

Si vous voulez utiliser Edupython chez vous :

Installer Edupython en allant directement sur le site officiel : <https://edupython.tuxfamily.org/>

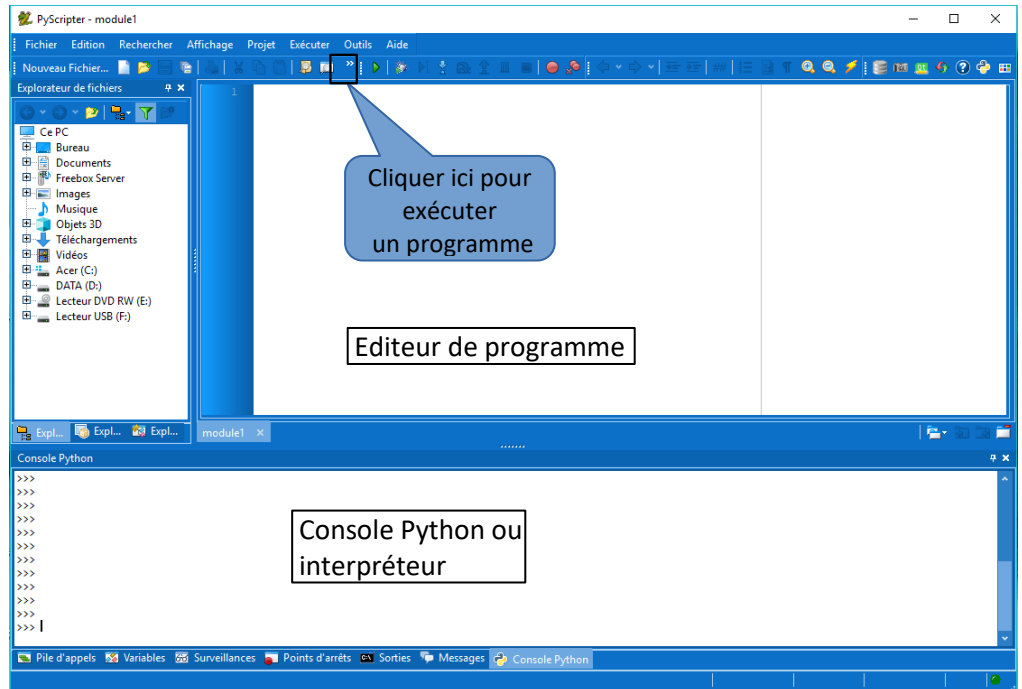
Il est aussi possible de travailler directement en ligne : <https://repl.it/languages/python3>

Au lycée, lancer Edupython depuis le dossier : *Ordinateur/Applications/SII/Edupython*

### Organisation de Pyscripter :

**L'éditeur** : zone de saisie du programme

**La console Python** : zone où le programme s'exécute ; cette zone permet aussi de tester directement des instructions en langage Python.



## II) Activité 1 : Premières approches.

Nous allons d'abord l'utiliser en mode interactif, appelé aussi le mode "console" : on dialoguera avec l'interpréteur directement depuis le clavier.

### a) Une calculatrice basique !

Testez en entrant les instructions suivantes dans la console ; valider en appuyant sur « Entrée » :

```
>>> 13 + 20
>>> 8 + 4 * 5
>>> 20 / 5
>>> 20 // 3
>>> 3 ** 2
```

Le symbole représentant le signe de multiplication est \* .

L'opérateur // effectue une division entière ...

Le symbole représentant la mise à la puissance est \*\* .

### Exercice 1 :

Effectuer à l'aide du logiciel les calculs suivants puis recopier les instructions que vous avez tapées :

1-)  $10 + 4 \times 5$       2-)  $3 \times (123 + 377)$       3-)  $20 \div 4 + 2$       4-)  $20^3$       5-)  $(10 - 4)^2$



## b) Nos premières variables, les affectations.

En Python l'**affectation** (ou l'assignation) à une variable se fait par le signe égal : "="

Testez :

<pre>&gt;&gt;&gt;r=5 &gt;&gt;&gt;r &gt;&gt;&gt;pi &gt;&gt;&gt;pi=3.14159 &gt;&gt;&gt;pi &gt;&gt;&gt;2*pi*r &gt;&gt;&gt;perimetre=2*pi*r &gt;&gt;&gt;perimetre &gt;&gt;&gt;message="Le résultat est"</pre>	<p>On <b>déclare</b> une <b>variable</b> nommée <b>r</b> et on lui <b>affecte</b> la valeur entière <b>5</b> .</p> <p>Cette instruction permet d'<b>afficher</b> dans la console la valeur de <b>r</b>.</p> <p><b>ERREUR</b> : on n'a pas déclaré de variable <b>pi</b> dans les instructions précédentes</p> <p>On <b>déclare</b> une <b>variable</b> nommée <b>pi</b> et on lui <b>affecte</b> la valeur 3.14159</p> <p>Dans le nombre décimal 3.14159, le <b>séparateur décimal</b> est <b>un point</b> et non une virgule.</p> <p>Cette fois <b>pas d'erreur</b> !</p> <p>On peut faire des calculs avec les nombres stockés dans les variables !</p> <p>On affecte à la variable périmètre le résultat du calcul situé à droite du « = ».</p> <p>La variable message n'est pas un nombre mais une chaîne de caractères !</p>
---	---

La virgule sert en langage Python à représenter autre chose que le séparateur décimal... **Testez** :

<pre>&gt;&gt;&gt; s,t=7,5 &gt;&gt;&gt; s &gt;&gt;&gt; t &gt;&gt;&gt; u,v,w=0.1,0.2,0.3 &gt;&gt;&gt; u</pre>	<p>On <b>déclare deux variables</b> nommées <b>s</b> et <b>t</b> et on leur <b>affecte</b> des valeurs entières.</p> <p>On <b>déclare trois variables</b> nommées <b>u</b> , <b>v</b> et <b>w</b> et on leur affecte des valeurs décimales.</p>
---	---

### Exercice 2 :

Voici trois séquences d'instruction :

1-) Donner pour chacune des séquences la valeur de **a** et **b** à la fin de la séquence.

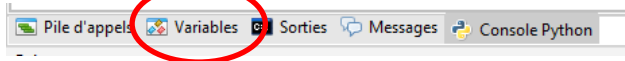
2-) Expliquer en une ou deux phrases en quoi la séquence 3 est différente des séquences 1 et 2.

Séquence 1	Séquence 2	Séquence 3
<pre>&gt;&gt;&gt; a=4 &gt;&gt;&gt; b=5 &gt;&gt;&gt; a=b</pre>	<pre>&gt;&gt;&gt; a=4 &gt;&gt;&gt; b=5 &gt;&gt;&gt; b=a</pre>	<pre>&gt;&gt;&gt; a=4 &gt;&gt;&gt; b=5 &gt;&gt;&gt; b=a &gt;&gt;&gt; b</pre>



### c) Les types.

Mais où en sommes-nous ? Pour le savoir, il suffit de cliquer sur :



Et on obtient un état des lieux :

Variables		
Nom	Type	Valeur
globals	dict	{'__builtins__': {'Arithme
message	str	'Le résultat est'
perimetre	float	31.4159
pi	float	3.14159
r	int	5

- **message** est du type **str** qui signifie **string** (une chaîne de caractères).
- **r** est du type **int** : qui signifie **integer** (un entier).
- **pi** et **perimetre** sont des **float** : des nombres décimaux.

**Le typage est automatique (décidé par l'interpréteur) et dynamique** : il peut évoluer au cours du programme .

On peut obtenir le type d'une variable à l'aide d'une instruction. **Testez :**

```
>>> type(message)
>>> type(perimetre)
>>> type(pi)
>>> type(r)
```

Le type d'une variable peut changer... **Testez :**

```
>>> m=7
>>> type(m)
>>> m=1.3
>>> type(m)
```

← **m** contient maintenant la valeur 1.3

**Testez :**

```
>>>a="5"
>>>b="4"
>>>a+b
>>>a= 5
>>>a+b
>>>b= 4
>>>a+b
```

← **a** et **b** sont tous de type **string** (chaîne de caractère).

← L'opération réalisée par cette instruction n'est pas l'addition de 4 à 5...

← **ERREUR !**

← On trouve le résultat auquel on s'attendait !

### Exercice 3 :

Quel est le type de chaque variable x,y,z et t ?

```
x=2.5
y="Bonjour"
z=7
t="18"
```

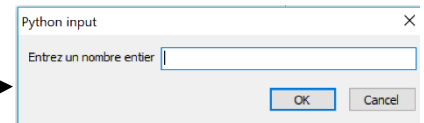


**d) Des fonctions : Input() , int(), float(), str() :**

**Testez :**

```
>>> m=input("Entrez un nombre entier")
>>> type(m)
>>> m+4
>>> m=int(m)
>>> type(m)
>>> m+4
```

Une fenêtre s'ouvre



**m** est de type **string**

**ERREUR** : on ne peut pas additionner un objet de type string avec un objet de type nombre entier.

Pour résoudre ce problème, on va « convertir » **m** en nombre entier à l'aide de la fonction **int()**.

Cette fois pas d'erreur !

On peut faire plus rapidement. **Testez :**

```
>>>m=int(input("Entrez un nombre entier"))
>>> type(m)
>>> m+4
```

On peut au lieu de convertir en un nombre entier, convertir en un nombre de type **float** . **Testez :**

```
>>>m=float(input("Entrez un nombre"))
>>> type(m)
>>> m+4
```

Les fonctions **int()**, **float()** et **str()** servent à « convertir » :

```
>>> m=3.54
>>> int(m)
>>> float(m)
>>> str(m)
```

Le résultat est un nombre entier (on a tronqué le nombre à l'unité).

m est un nombre décimal donc pas de changement ...

Le résultat est une chaîne de caractère (donc de type **string**).

**Exercice 4 :**

On donne le programme suivant :

```
>>> x= int(input("Entrez x"))
>>> y=8*x
>>> y=y+2
>>> y=y**2
>>>y
```

**1-) Quel est le type de x ?**

**2-) Pour chacune des valeurs de x saisie en entrée, donner la valeur y affichée en sortie .**

**a) x=4**

**b) x=-1**

**c) x=11**