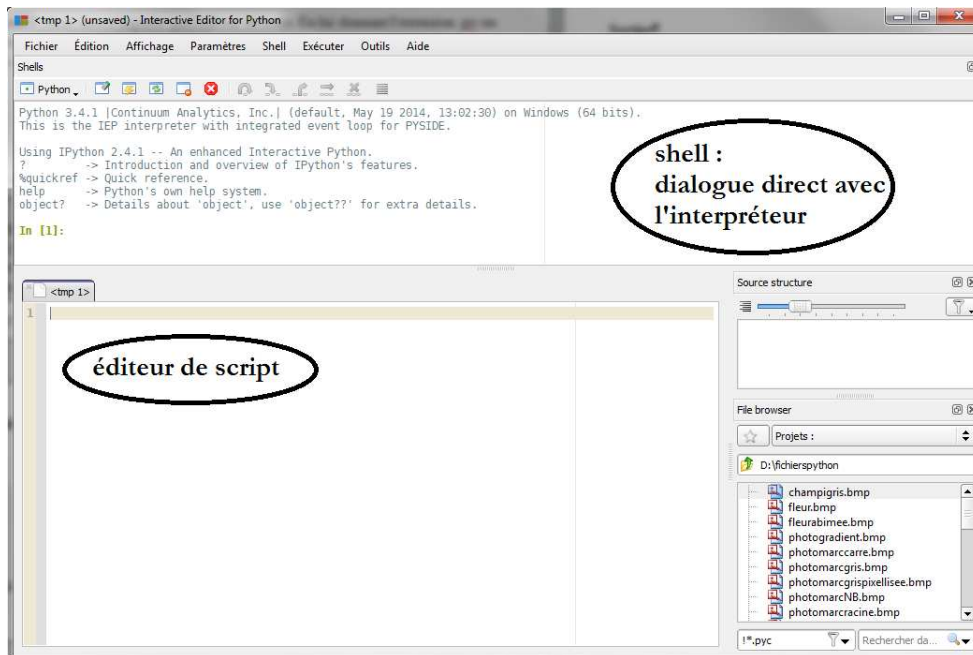


Nous verrons dans ce TP, les variables, entrées/sorties, organigrammes et séquences d'instructions.

On peut écrire un programme Python dans n'importe quel éditeur de texte. En lui donnant l'extension .py on peut ensuite exécuter le programme si Python est installé sur l'ordinateur.

Néanmoins, grâce à la distribution Pyzo, nous bénéficions d'un environnement de travail qui nous facilite l'écriture du code, son débogage. . .

Pour installer le langage python et le logiciel pyzo, utilisez le lien [ <http://www.pyzo.org/start.html> ].



## I Utilisation du shell

Dans cette partie, nous utiliserons le shell proposé par pyzo.

Au travers de premières commandes simples, nous allons découvrir les premiers types de variables que nous utiliserons. À savoir les nombres entiers, réels ou complexes, les chaînes de caractères et les tableaux.

### 1 Calculer avec Python

```
In[.]: 3+5
In[.]: 2    +    9    # l'espace change-t-il le resultat ?
In[.]: 7 + 3 * 4    # les priorites operatoires sont-elles respectees ?
In[.]: 20 / 3    # etait-ce ce que vous attendiez ?
```

Un peu de maths Comment lui fait-on calculer  $2^3$  ?

```
In[.]: from math import *
In[.]: 2**3
```

Et comment peut-on calculer  $\sqrt{2}$  ?

```
In[.]: from math import *
In[.]: sqrt(2)
In[.]: sqrt(2)**2
```

Que constatez-vous ?

### 2 Les affectations

```
In[.]: n = 7
In[.]: msg = "Hello world"
In[.]: pi = 3.14 # Important : on utilise la notation anglo-saxonne (, <-> .)
```

Entrer n, msg puis pi. Que se passe-t-il ?

### 3 Les sorties

Pour afficher une valeur ou un message à l'écran on utilise la fonction `print()`.

```
In[:]: print(n)
In[:]: print(msg)
In[:]: print(pi)
# et meme...
In[:]: print(n, msg, pi, "vive moi")
```

Que se passe-t-il ?

**Exo 1 :** transformation des types d'une variable

Sous Python, définir `n="10"`, `m=10` et `p=0,1`.

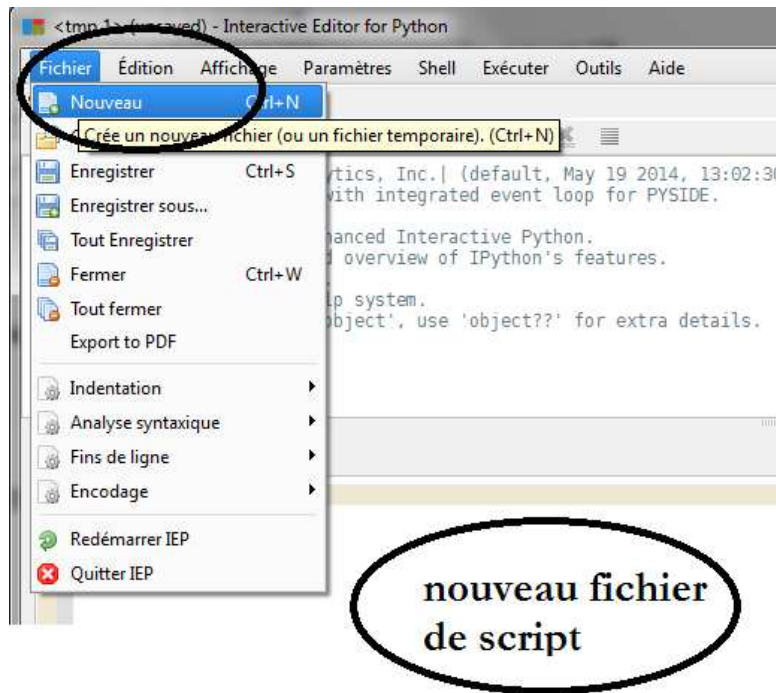
1. taper `type(n)` ; puis `type(m)` ; puis `type(p)` et enfin `print(n, m)`. Que remarquez-vous ?
2. Rappelez l'inclusion des ensembles entiers naturels, entiers relatif, quotient et réels.
3. Taper `type(0.01*m)` et `type(10*p)`. Que remarquez-vous ? Comment l'expliquez-vous ?

## II Utilisation de script

Jusqu'à présent nous avons travaillé dans l'interpréteur (Python shell). Chaque fois que nous faisons Entrée, la ligne que nous venions d'écrire était immédiatement interprétée et exécutée. Impossible donc d'écrire un programme sur plusieurs lignes.

Dorénavant, chaque fois que vous voudrez programmer en Python, vous le ferez dans un fichier de script. Attention, Pyzo vous ouvre toujours par défaut les derniers fichiers Python utilisés sur l'ordinateur (probablement pas les vôtres!). Il vous faudra donc en créer de nouveau (File -> New) ou fermer ceux proposés et ouvrir les vôtres.

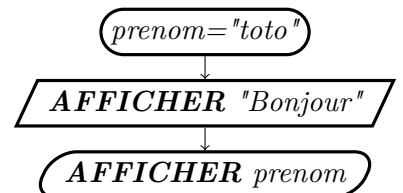
Vous devrez l'enregistrer à une place adéquate dans vos documents, l'idéal étant de vous créer un répertoire avec votre nom sur la machine.



expl : saisir dans la fenêtre du script

```
prenom = "toto"
print("Bonjour", prenom)
```

Ce qui correspond à l'organigramme suivant :



Enregistrez-le dans votre répertoire personnel et lancez-le avec **F5**.

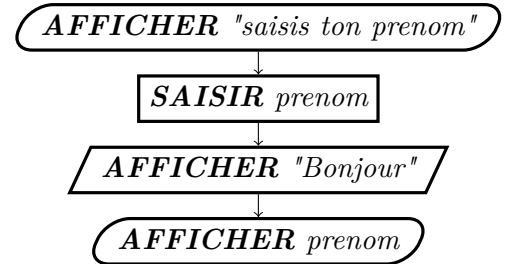
## 1 Les entrées

Pour que la valeur de la variable ne soit pas fixée dans la programme mais soit donnée par l'utilisateur, on peut utiliser l'instruction `input`.

```
prenom = input (" saisis ton prenom ")
print (" Bonjour ", prenom)
```

Que se passe-t-il ?

rq : `pre nom=input("")` est en réalité la succession de ... instructions.



## 2 Influence du typage

```
age = int(input("quel est ton age ?"))
print("tu as ", age, " ans")
```

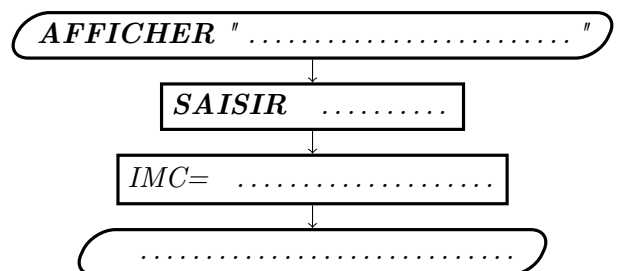
Essayez en entrant des lettres, un entier ou un réel et comparez les résultats.

Exo 2 :

L'algorithme suivant permet de calculer l'IMC d'une personne mesurant 1,60m.

1. Quel(s) type(s) vous semble(nt) adapté(s) pour les variables *poids* et *IMC* ?
2. Compléter l'organigramme de cet algorithme.
3. Implémenter l'algorithme en Python dans un script (attention au type de *poids*)
4. (facultatif) Modifier l'organigramme puis l'algorithme afin que l'utilisateur puisse donner sa taille et que son IMC soit calculé.
5. (facultatif) Implémenter votre nouvelle solution.

Variables	
<i>poids</i> :	.....
<i>IMC</i> :	.....
Algorithme	
	AFFICHER "Combien pesez-vous en kg?"
	SAISIR <i>poids</i>
	$IMC \leftarrow poids / (1.60)^2$
	AFFICHER IMC
Fin Algorithme	



## Travail à faire pour la prochaine séance

Exo 3 : Faire les questions facultatives de l'exo en fin de TD.

Exo 4 :

1. Proposer l'organigramme de l'algorithme suivant.
2. Que donne l'exécution de la séquence suivante ?
3. Quelles sont les valeurs de *x* et de *y* ?

Variables	
<i>x</i> :	entier
<i>y</i> :	entier
Algorithme	
	$x \leftarrow 3$
	$y \leftarrow x + 1$
	$x \leftarrow 5$
Fin Algorithme	

Exo 5 : En s'inspirant de la fin du TD, on veut créer un petit programme qui demande l'âge de l'utilisateur, puis qui lui donne son âge en 2050 :

1. Proposer un algorithme et son organigramme répondant à votre problématique.
2. Quel type vous semble le plus adapté pour la variable *age* ?
3. Implémenter l'algorithme en Python dans un script.