

# TP 1 informatique

BCPST 1 2017-2018

## Prise en main

---

### Points abordés

- Types, variables, expressions.
  - Programme, langage *Python*.
- 

### Introduction

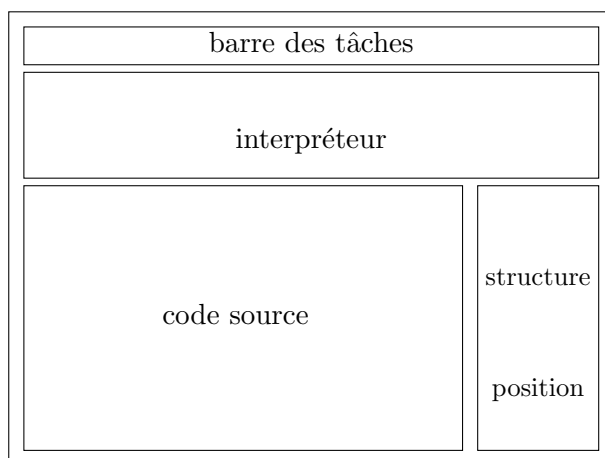
1. Créer un dossier **Informatique**.
2. Dans celui-ci créez un dossier **TP1**. Par la suite, vous enregistrerez vos fichiers correspondant à ce TP dans ce dossier.

Pour une bonne organisation, on créera un dossier **TPX** pour y placer les fichiers correspondant.

## 1 L'interface Pyzo

Il existe différentes manières de programmer en

1. Lancer l'application *Pyzo*. Constater que votre écran est maintenant divisé en plusieurs parties :
  - (a) la partie “**interpréteur**”, qui lit du code écrit en python et le traduit en langage machine, pour que celle-ci puisse l'exécuter, et vous donner le résultat.
  - (b) une partie “**code source**”, qui est un fichier comportant des programmes écrit en langage python. Il s'agit en fait d'un fichier texte qui peut être utilisé par l'interpréteur.
  - (c) Une partie comportant la structure du code, c'est-à-dire les différentes fonctions du code et leurs dépendance (cette partie ne nous intéresse pas pour le moment)
  - (d) la partie qui correspond à la position du fichier code source dans l'ordinateur.



2. Placez-vous à l'aide de la souris dans la partie **interpréteur**. Votre curseur maintenant y figure. **L'interpréteur** ne comprend que des commandes ou lignes de *Python*.
3. Saisissez : Bonjour. Constatez une erreur.
4. Saisissez : `print("Bonjour")`. Constatez que l'interpréteur affiche bien "Bonjour".
5. Afficher dans l'interpréteur "On débute en Python".
6. Saisissez `message="Bonjour"`. Cela signifie qu'à la variable `message`, on affecte la valeur "Bonjour".
7. Saisissez `message`. On constate que cette fois-ci, l'interpréteur affiche :

*Out[?]* : 'Bonjour'

*Out* correspond au fait que l'interpréteur a une valeur de retour, et il s'agit dans notre cas de la **chaîne de caractères** "Bonjour".

8. Dans la barre des tâches, cliquez sur **shell** et redémarrer.
9. Saisissez `message`. Constatez que l'on a maintenant une erreur.  
L'inconvénient de l'interpréteur, est que si on a écrit un long programme, que l'on a envie de le modifier, ce n'est pas évident. De plus, en redémarrant l'interpréteur ce qui a été fait aura été effacé.
10. Placez-vous dans le code source.
11. Saisissez `message="Bonjour"`.
12. Enregistrez votre fichier dans le bon dossier.
13. Exécutez le script.
14. Dans l'interpréteur, saisissez **message**.

**Remarque 1.** Les fichiers enregistrés sont des fichiers texte pouvant être ouvert avec n'importe quel éditeur de texte. Ils ont comme extension **.py**.

## 2 Les types en Python

Il existe de nombreux types en *Python*, les entiers, les flottants, les chaînes de caractères, les booléens, les tuples, les listes, les dictionnaires... Durant cette séance, on se concentre sur les entiers et les booléens. Un des points importants en informatique est de pouvoir affecter une valeur à une variable. Par exemple, précédemment, on a affecté à la variable **message** la valeur "Bonjour".

1. Affectez à la variable **Proposition** la valeur **True**.
2. Il est possible de vérifier le **type** d'une variable  $x$  ainsi :  
 $type(x)$ .
3. Quel est le type de  $x$  ?
4. Essayez d'affecter à la variable **False** la valeur **True**. Il existe en fait des mots qui sont réservés par le langage. Il s'agit des mots-clés (ex : **if**, **and**,...) et des constantes (entiers, chaînes de caractère,...)  
Un autre point important en *Python* est de pouvoir tester si deux expressions ont même valeur. Cette comparaison est effectuée à l'aide de l'opérateur `==`.
5. Dans l'interpréteur, saisissez  $(2 + 2) == 4$ . Quelle est la valeur de retour ?
6. Quel est le type de **True** ? de **False** ?

## 2.1 Les entiers

En python, les opérateurs suivant sur les entiers sont déjà présents :

- l'addition `+` ;
- la multiplication `*` ;
- la soustraction `-` ;
- la division entière `//` ;
- le modulo `%` ;
- l'exponentiation `**`.

Saisissez le programme :

```
print("saisissez deux entiers : \n")
m=int(input())
n=int(input())
n=n+m
print("le resultat est : ", n)
```

Quel est le résultat affiché par ce programme ?

1. Dans l'interpréteur, saisissez  $z=input()$ . Que se passe t-il ?
2. Saisissez 36.
3. Quel est le type de  $z$  ? La fonction *input* renvoie toujours une chaîne de caractères. Cependant, il est possible de convertir les chaînes de caractères en entiers. Pour cela, il suffit d'utiliser la fonction **int**.
4. Sur le même modèle que précédemment, écrire un programme qui demande à l'utilisateur de saisir deux entiers  $a$  et  $b$  et qui affiche le reste de la division euclidienne de  $a$  par  $b$ .

## 2.2 Chaînes de caractères

Le type chaînes de caractères correspond aux suites finies de caractères, qui peuvent être des lettres, des signes de ponctuations... Différentes fonctions existent également sur les chaînes de caractères.

1. Affectez à la variable **souris** la chaîne de caractères "**Minus et Cortex**".
2. Saisissez  $len(souris)$ . Quelle est la valeur de retour ?
3. Affectez respectivement à la variable **oiseau** "titi " et à la variable **chat** "et gros minet". Quel est la valeur de retour de **oiseau + chat** ?

4. Saisissez “**i**” in “**titi**”. Quelle est la valeur de retour ?
5. Essayez “**Minus**” in **souris**.
6. Essayez **Souris**[0].
7. Essayez **Souris**[-1].
8. À l’aide de crochets, trouver la 6-ième lettre de la chaîne contenue dans **souris**.

Déduire des exemples précédents à quoi correspondent les valeurs de retour des fonctions précédentes sur les chaînes de caractères.

**Un point très important est que les valeurs de retour peuvent être stocker dans des variables.** Par exemple, “**Anime**=oiseau+chat” affecte à la variable **Anime** la chaîne de caractères “titi et gros minet”.

1. Saisissez chat=oiseau+chat.
2. Quel est la valeur de **chat** ?

### 3 Exercices

**Exercice 1.** Écrivez un programme **prenom.py** qui demande à l’utilisateur de saisir son prénom puis qui affiche :

```
"Bonjour Prenom !"
```

où **Prenom** correspond à la chaîne de caractère entrée.

**Exercice 2.** 1. Écrivez et exécutez le programme **age.py** qui à l’aide d’une saisie au clavier, affiche sur une ligne le message suivant :

```
Le capitaine a C ans. Dans deux ans, il en aura C'.
```

2. Sachant que dans cinq ans le perroquet aura le tiers de l’âge du capitaine (arrondi à l’inférieur), compléter votre programme pour qu’il affiche

```
le perroquet du capitaine a P ans.
```

**Exercice 3.** Écrire un programme **echange.py** qui lit deux variables de type entière, les affiche, échange leur contenu et les affiche de nouveau.

**Exercice 4.** Écrivez un programme **conversion.py** qui lit un nombre entier correspondant aux nombres de secondes et convertit celui-ci en heures minutes, secondes. Par exemple, si l’utilisateur saisit 3698, on obtient l’affichage suivant :

```
3698 s correspondent à 1 h 1 mn 38 s.
```

**Exercice 5.** Écrire un programme **contientVoyelles.py** qui lit une chaîne de caractères et qui affiche “contient au moins une voyelles” si la chaîne saisie en contient et affiche “ne contient que des consonnes” sinon.