

Cours 1 : Se familiariser avec Python

L'une des compétences essentielles dans le métier d'ingénieur est la maîtrise du langage informatique. Le langage retenu par les écoles est Python. Nous utiliserons le logiciel Pyzo.

Pyzo peut être utilisé de deux manières différentes (et complémentaires) :

1. On peut utiliser la **console interactive** et se servir de Pyzo comme d'une **calculatrice**
2. On peut écrire des **programmes** (c'est à dire une suite d'instructions) dans l'**éditeur de scripts**, les enregistrer puis les exécuter

I Utilisation de la console interactive de Python

I.1 Opérations

On peut utiliser la console pour effectuer des opérations. Que renvoient à votre avis les instructions suivantes ?

```
>>> 2 + 3 #somme
>>> 2 - 3.0
>>> 2 * 3
>>> 2**3
>>> 10 / 3
>>> 10 \% 3
>>> 10 // 3
```

⇒ Essayez !

```
>>> 1 / (2+2-4)
>>> 1 / 'a'
>>> 2 = 3
>>> a = 2
>>> 1 + a
```

I.2 Tests

On peut effectuer des tests pour comparer des nombres, des objets, des booléens

⇒ Essayez !

```
>>> 2 <= 3
>>> 2 == 2.0
>>> 'b' == 'ba'
>>> 2 == 3
>>> 2 != 3
>>> (2 < 3) or (2 > 3)
>>> (2 < 3) and (2 > 3)
>>> not 2<3
```

De manière générale, donnez l'interprétation des commandes ci-dessous :

commande Python	Interprétation	commande Python	Interprétation
<code>x == y</code>		<code>x <= y</code>	
<code>x != y</code>		<code>x in y</code>	
<code>x > y</code>		<code>x > y and y > z</code>	
<code>x < y</code>		<code>x > y or y > z</code>	
<code>x >= y</code>		<code>not x < y</code>	

II Ecriture de programmes (ou scripts)

L'éditeur de scripts est un espace dans lequel on peut rédiger une suite de lignes de codes afin de répondre à une problématique donnée. Ces lignes de codes constituent ce que lon appelle un programme.

Afin d'écrire un programme, il convient de suivre une certaine logique :

- (a) Initialisation du programme : les premières lignes de codes permettent d'importer toutes les bibliothèques et fonctions nécessaires au bon fonctionnement du programme, ainsi qu'initialiser les valeurs de certaines variables utiles au programme;
- (b) Déroulement du programme : dans le coeur du programme, des lignes de codes permettent deffectuer les opérations désirées, de créer les fonctions nécessaires, d'appeler des fonctions déjà existentes, d'utiliser des boucles...
- (c) Conclusion du programme : à la fin du programme, on peut renvoyer le résultat du problème initialement posé, afficher des courbes,...

III Variables

III.1 Affectations

Pour affecter une valeur à une variable, on utilise la commande : `nom_variable = valeur`

```
>>> a = 1
>>> b = True
>>> c = "bonjour"
```

III.2 Types

Pour connaître le type d'une variable a , on utilise la commande : `type(a)`

Quel est le type des variables suivantes ?

```
>>> a = 1
>>> b = 2.5 * a
>>> c = 'Hello World'
>>> d = 1/3
>>> e = b<a
```

On peut aussi forcer le type :

```
>>> int(2.4)
>>> int(-1.4)
>>> float(3)
>>> str(3)
```

IV Bibliothèques

IV.1 Préambule

Python comprend un ensemble de fonctions de base prédéfinies : par exemple les opérations élémentaires, l'échange de deux variables ...

Une **bibliothèque** (ou **package**) est un ensemble de fonctions spécifiques à un thème donné. Il ne faut pas reconstruire des fonctions qui existent déjà : il s'agit d'importer tel ou tel package en fonctions des besoins du moment.

Quelques exemples de bibliothèques :

Pour importer une bibliothèque, on utilise la commande : `import nom_package`
Exemples d'utilisation des fonctions appartenant à un package. ⇌ **Essayez !**

```
>>> import math
>>> math.cos(math.pi/2)

>>> import math as m
>>> m.cos(m.pi/2)

>>> from math import *
>>> cos(pi/2)

>>> from math import floor
>>> floor(pi/2)
```

Les commandes de **math** sont accessibles avec le préfixe **math**

Les commandes de **math** sont accessibles avec le préfixe *m*.

Les commandes de **math** sont accessibles sans préfixe! (*non conseillé*)

On importe seulement la fonction dont on a besoin.

Remarque : une fois que l'import est fait, il est valable pour toute la suite de votre programme.

Pour afficher toutes les fonctions de la bibliothèque **math**, on utilise la commande : **print(dir(math))**

⇒ Essayez !

Pour afficher de l'aide sur la fonction **ceil**, on utilise la commande : **help(math.ceil)**

IV.2 La bibliothèque math

Les constantes **pi** et **e** sont accessibles dans le package **math**. Voici une liste des fonctions accessibles dans **math** :

cos	sin	tan	acos	asin	atan	abs	sqrt	exp
log	log10	log(x,base)	factorial	floor	ceil	degrees	radians	