

# A retenir du chapitre 1 : Prise en main de Python



## I Raccourcis

- Ctrl + E : exécution du fichier,
- Alt+Return : exécution de la sélection,
- Tab : indentation,
- Maj+Tab désindentation,
- Flèches de défilement : historique de la console,

## II Opérations basiques

### 2.1 Affectation

- $x=...$  : affectation de la variable  $x$ ,
- $x,y=...$  : affectation multiple des variables  $x$  et  $y$ ,

### 2.2 Incrémentation

- $x+=a$  :  $x=x+a$ ,
- $x-=a$  :  $x=x-a$ ,
- $x*=a$  :  $x=x*a$ ,
- $x/=a$  :  $x=x/a$ ,

### 2.3 Calcul algébrique

- $**$  : puissance,
- $abs(...)$  : valeur absolue,
- $//$  : division entière (quotient de la division euclidienne),
- $%$  : reste de la division euclidienne,

### 2.4 Opérations sur les booléens

- **True** : vrai,
- **False** : faux,
- **and** : et,
- **or** : ou,
- **not** : négation,

### 2.5 Tests

- $==$  : test d'égalité,
- $!=$  : test de différence,
- $<=$  et  $>=$  : test d'inégalité large,
- $<$  et  $>$  : test d'inégalité stricte,

## 2.6 Types

- **type(...)** : obtention du type,
- **int** : entier,
- **float** : flottant,
- **bool** : booléen,

## III Modules

### 3.1 Importation de module

- **import ...** : importation du module avec nécessité de préfixer,
- **import ...as ...** : importation du module avec nécessité de préfixer avec l'abréviation,
- **from ...import \*** : importation du module sans nécessité de préfixer,

### 3.2 Commandes du module math

- **sqrt(...)** : racine carrée,
- **cos(...),sin(...),tan(...)** : fonctions trigonométriques,
- **pi,e** : nombre pi, nombre e
- **exp(...),log(...)** : exponentielle, logarithme népérien,

## IV Entrées et sorties

- **print(...)** : affichage,
- **input(...)** : affichage demandant une réponse de l'utilisateur (format chaîne de caractères),
- **eval(input(...))** : affichage demandant une réponse de l'utilisateur (format numérique),
- **def** : définition d'une fonction,
- **return** : renvoi d'une fonction.



- Créer un fichier pour sauvegarder le travail de chaque séance.
- Choisir un nom de variable qui a un sens.
- Bien lire les messages d'erreur.
- Importer les modules une seule fois, en début de fichier.
- Une fonction doit être exécutée après chaque modification et testée sur un ou des exemples.
- Une fonction doit être commentée.



- L'affectation n'est pas commutative.
- On n'utilisera pas de test d'égalité entre flottants.
- L'indentation (le bloc de quatre espaces précédant les instructions) est obligatoire : cela signifie que l'on se situe à l'intérieur de la fonction.
- Les commandes qui suivent un return ne sont pas lues (code mort).



- L'évaluation booléenne est paresseuse : les booléens sont évalués dans l'ordre d'écriture et la réponse est donnée avant la fin de l'évaluation de tous les booléens.
- Les variables locales sont définies à l'intérieur d'une fonction. Elles peuvent être modifiées au cours de l'exécution et seront automatiquement effacées à la fin de l'exécution.
- Les variables globales sont définies à l'extérieur d'une fonction. Elles ne peuvent pas être modifiées par la fonction (à moins de les avoir déclarer). Il n'y a pas de conflit de nom entre les variables locales et globales.