NOTE 2
Principaux types de données

ESIEE[it]
L'école de l'expertise numérique
une école de la
CCI PARIS ILE-DE-FRANCE

Types de données

- **Basiques**
  - Booléens : *True* ou *False*
  - Numériques : 3, *3.13e10, 0.12*
  - Chaines de caractères ‘Bonjour’
  - Nombres complexes *3.2+2.5j*

- **Complexes**
  - Tuples
  - Listes
  - Dictionnaires
  - Ensembles (sets)
  - Fonctions et méthodes
  - Classes

- **type(v)**

Pour connaître le type d'une variable v, on utilise la fonction *type(v)*

```
>>> a=3
>>> type(a)
<class 'int'>
>>> b='bonjour'
>>> type(b)
<class 'str'>
>>>
```

- **Changer de types**

Le transtypage ou le cast permet à une variable de changer de type grâce par exemple à des fonctions

```
>>> a=3
>>> type(a)
<class 'int'>
>>> a=str(a)
>>> type(a)
<class 'str'>
>>>
```

Opération	Résultat
float(x)	Conversion de *x* en flottant
long(x)	Conversion de *x* en entier long
int(x)	Conversion de *x* en entier
str(x)	Conversion de *x* en chaîne de caractère

Opérateurs arithmétiques

- Un opérateur est un symbole qui va être utilisé pour effectuer certaines actions notamment sur les variables et leurs valeurs.

- Les opérateurs arithmétiques vont nous permettre d'effectuer des opérations mathématiques entre les valeurs de type Number contenues dans des variables.

Opérateur	Description	Exemple	Résultat avec y=5
+	Addition	x=y+7	x=12
-	Soustraction	x=y-4	x=1
*	Multiplication	x=y*2	x=10
/	Division	x=y/2	x=2.5
%	Module (reste de division)	x=y%2	x=1
++	Incrémentation	x=++y	x=6
--	Décrémentation	x=--y	x=4

Opérateurs d'affectation

- Les opérateurs d'affectation permettent d'affecter (de donner) une valeur à des variables.

Opérateur	Exemple	Mêmes que	Résultat
=	x=y	Affecte la valeur de y à x	x=5
+=	x+=y	x=x+y	x=15
- =	x-=y	x=x-y	x=5
*=	x*=y	x=x*y	x=50
/=	x/=y	x=x/y	x=2
%=	x%=y	x=x%y	x=

Il existe d'autres opérateurs comme :

- Les opérateurs de comparaison
- Les opérateurs logiques
- L'opérateur ternaire
- l'opérateur virgule

```
>>> a,b=3,4
>>> print(a)
3
>>> print(b)
4
>>>
```

Il est possible de faire de la multi-affectation en Python On donne une valeur simultanément à plusieurs valeurs

Opérations numériques

Il existe de nombreuses fonctions pour manipuler les variables numériques, on distingue deux types

- **Les fonctions procédurales** écrites comme en mathématiques, les paramètres sont entre parenthèses dans la fonction :

Opération	Résultat	
pow(x,y)	*x* puissance *y*	*>> pow(3,2)* *>>9 #3²*
divmod(x,y)	La paire (*x//y,x%y*)	*>> divmod(11,2)* *>>(5, 1) #5 est le diviseur et 1 le reste*

- **Les fonctions /méthodes objet** , le paramètre qui se lequel va opérer la fonction est mis devant la fonction

Opération	Résultat	
c.conjugate()	Le conjugué de *c* (si *c* ℂ)	*>>a=3.2+2.5j* *>> b=a.conjugate()* *>>print(b)* *>>3.2-2.5j*

Chaînes de caractères

Une chaîne de caractères (string) est une suite de caractères (alphabétiques, chiffres, spéciaux) considéré par l'ordinateur comme du texte.

```
>>> a='chaine1'
>>> b="chaine2"
>>> c='''Chaine3'''
>>> print(a)
chaine1
>>> print(b)
chaine2
>>> print(c)
Chaine3
>>>
```

1 caractère = une chaîne d'1 caractère

Les opérations sur les chaines de caractères sont les suivantes:

- L'addition (+) permet de concaténer deux string

```
>>> a='debut de chaine'
>>> b='fin de chaine'
>>> c=a+b
>>> print(c)
debut de chainefin de chaine
```

- La multiplication (*) permet de multiplier un string par un nombre entier

```
>>> a="ab"
>>> a*5
'abababababab'
```

ESIEE[it]
L'école de l'expertise numérique

Formatage des chaînes de caractères

Il est possible d'ajouter des variables dans une chaine de caractères. Pour cela, il faut ajouter les balises suivantes:

« %s » : Pour ajouter un string

« %i » : Pour ajouter un nombre entier

« %.5f » : Pour ajouter un nombre à virgule (ici, 5 correspond au nombre de décimales voulues)

Format	Signification
%%	Le caractère '%'
%s	Chaîne
%c	Caractère
%f	Flottant
%e	Flottant (notation expo)
%d	Entier décimal
%x	Entier hexadécimal
%o	Entier octal

```
import numpy

pi = numpy.pi
nom = "Pi"

sortie = "La valeur de %s est de %f mais on peut l'arrondir à %.2f."%(nom, pi, pi)
print(sortie)
```

```
>>>
La valeur de Pi est de 3.141593 mais on peut l'arrondir à 3.14.
>>>
```

ESIEE[it]
L'école de l'expertise numérique

Interaction avec l'utilisateur

print pour faire afficher quelque chose

- Affiche les arguments (séparés par des virgules) sur la sortie standard
- Termine automatiquement par un saut de ligne (sauf si le dernier caractère est ,)
- Arguments convertis en chaine de caractères avec *str()*

input pour demander à l'utilisateur la saisie d'une valeur

- Interrompt exécution du programme et attend une entrée
- Valeur renvoyée par la fonction sous forme de **chaine de caractères**

```
age = input("quel age as tu ?")
print (age)
print ("ton age est "+age)
print "ton age est %s", %(age)
```

ESIEE[it]
L'école de l'expertise numérique

une école de la
CCI PARIS ILE-DE-FRANCE

www.esiee-it.fr