



# Analyse de données Python

Cours par Nizar Ouarti  
(2022-2023)

# Plan du cours



**Les différentes parties détaillées tout au long du cours sont les suivantes:**  
Aucun prérequis n'est attendu pour ce cours



## **Variables**

Types de données (built-in), “mutable / immutable”

## **Built-in types**

Liste, dictionnaire, tuple, set , string, int, float, bool

## **Fonction**

Fonctions en python, paramètres, return, usage

## **Objet**

Création et utilisation d'un objet

## **Listes**

Création et manipulation des listes

## **If while for et Entrees / Sorties**

Structure de contrôle et entrée - sortie : input print

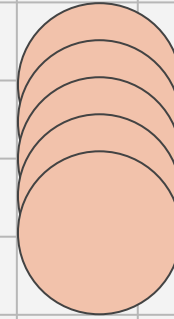


Plus d'informations:  
[nizar.ouarti@sorbonne.fr](mailto:nizar.ouarti@sorbonne.fr)

Page internet:  
**F page nizar**

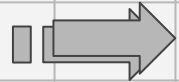
Qu'est ce qu'une

# Variable



Une variable en Python peut être décrite de la manière suivante:

- Une variable est une “**boîte**” contenue en mémoire
- Cette boîte possède : un **type**, une **adresse**, une **valeur**, une **taille**
- **type()** , **id()** , **print()**, **sys.getsizeof()**
- Une variable peut être **mutable** ou **immutable**
- Une variable peut être **locale** ou **globale**
- Les variables sont passées par **référence** dans une **fonction**: par **adresse**





# Variables

## IMMUTABLE

01

int, float, bool

Int: nombre entier, float:  
nombre à virgule, bool  
nombre binaire

02

str

str: chaîne de caractère

03

tuple

Liste d'items immuable

## MUTABLE



list

Liste d'items qui peut  
être modifiée et agrandi

04

set

Ensemble de données non  
redondantes

05

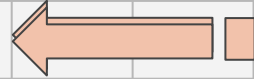
dictionnary

Association entre une clef  
et une données

06



# Exemple de création de variable



- **Entier**

```
>>> x=5
>>> type(x)
>>> <class 'int'>
>>> x=5
>>> type(x)
>>> <class 'int'>
```

- **Chaine de caractère (string)**

```
>>> x="bonjour"
>>> type(x)
>>> <class 'str'>
>>> import sys
>>> sys.getsizeof(x)
>>> 56
```

- **Tuple**

```
>>> montuple=(5.6, 3.4, 2.7)
>>> montuple[2]
>>> 2.7
>>> montuple[2]=4.5 -> error
```

- **List**

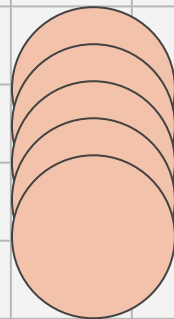
```
>>> maliste=[3.5, 2.3, 6.6]
>>> maliste[1]=2.9
>>> maliste
>>> [3.5, 2.9, 6.6]
```

- **Set et Dictionnaire**

```
>>> monset={3.5, 2.3, 6.6, 2.3}
>>> mondico={"name": "Freeman"}
```

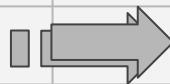
Qu'est ce qu'une

# Fonction

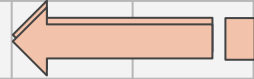


Une fonction permet:

- De réutiliser plusieurs fois du code
- De rendre le code plus intelligible
- De rendre le code **modulable**
- De modifier des paramètres
- D'isoler les actions faites dans la fonction: variable immutables ou copy

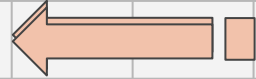


# Fonction en python



- **Mot clef def**
  - Def:sert à définit une fonction
  - `def affichage()`
- **Les entrées de la fonction**
  - Les paramètres sont les entrées de la fonction
  - Les paramètres peuvent être mutables ou immutables
  - La mutabilité influencera le changement de valeur des variables entrées en paramètre (locale ou non)
- **Les sorties de la fonction**
  - Les sorties de la fonction se font avec le mot clef **return**
  - Attention! En cas de mutabilité les paramètres seront aussi changés dans la fonction
  - Donc en théorie, il y à deux manières d'avoir des sorties

# Usage d'une fonction



- **Paramètre de fonction**

```
>>> def square(x):  
>>>     x=x**2 #important 4 espaces  
>>>     return(x)  
>>> x=2  
>>> square(x)  
4  
>>> print(x) #x n'est pas modifié  
2  
>>> def squarel(x):  
>>>     x[0]=x[0]**2 #important 4 espaces  
>>>     return(x)
```

- **Paramètre de fonction**

```
>>> x=[2] #ici je crée une liste  
>>> squarel(x)  
[4]  
>>> print(x) #x est modifié  
[4]
```

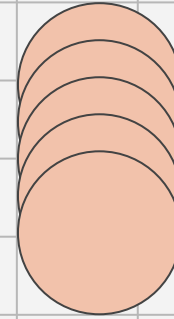
- **Plusieurs variables de sortie**

```
>>> def getLivre(numero,librairie):  
>>>     auteur=librairie[numero].auteur  
>>>     titre=librairie[numero].titre  
>>>     annee=librairie[numero].annee  
>>>     return (auteur, titre, annee)  
>>> (aut, tit, an) = getLivre(35,librairie)
```



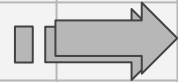
Qu'est ce qu'un

Objet

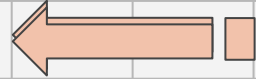


Objet en Python:

- En Python **toute variable est un Objet**
- Un **Objet** contient des **attributs** et des **méthodes**
- Les attributs sont des variables et les méthodes sont des fonctions
- Un objet défini par l'utilisateur est créé avec le mot clef **class**
- Le constructeur est la fonction appelé `__init__(self)`

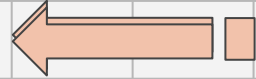


# Création d'un objet



- **définition d'objet**  
**class Livre () :**  
    **def \_\_init\_\_(self, auteur, titre, annee):**  
        **self.auteur=**  
        **self.titre=titre**  
        **self.annee=annee**  
  
**librairie=[]**  
**livre1 = Livre("Aristote", "La métaphysique", "-400")**  
**librairie.append(livre1)**  
**livre2 = Livre("Ovide", "Les Métamorphoses", "1")**  
**librairie.append(livre2)**  
**livre3 = Livre("Pascal", "Les Pensées", "1670")**  
**librairie.append(livre3)**

# Listes

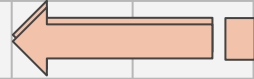


- **Une liste peut être définie ainsi:** `liste=[2,5,1,4]`
- **Plusieurs fonctions agissent sur les listes**
  - `len` : connaître le nombre d'éléments
  - `min`: trouver la valeur min de la liste
  - `max`: trouver la valeur de la liste
  - `index`: retrouve l'index pour une valeur
  - `append`: sert à ajouter un élément
  - Et plus encore `>>> liste. ## appuyer sur tab`
  - `dir(liste)` # va liste les attributs et méthodes
- **Pour accéder à un élément d'une liste**
  - Récupérer la valeur: `val=liste[index]`
  - Pour modifier la valeur `liste[index]=val`

**Pour ajouter un élément dans une liste**

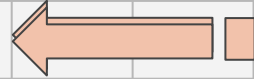
```
>>> liste=[]
>>> liste.append(1)
>>> [1]
>>> liste.append(3)
[1,3]
>>> len(liste)
2
>>> min(liste)
1
>>> list.index(3)
1
>>> dir(list)
```

# Structures de contrôle: if while for



- **Un bloc en python commence toujours avec “:”**
- **Un bloc est délimité par le même niveau d'espaces ou de tabulation**
- **Une instruction va s'appliquer sur un bloc**
  - On a déjà vu les blocs pour les fonctions et les classes. Le bloc délimite leur fin
- **If elif else**  
If latte== “petit”:  
    print(“Le choix est un latte petit”)  
elif latte == “moyen”:  
    print(“Le choix est un latte moyen”)  
elif latte == “grand” or latte==“large”:  
    print(“Le choix est un latte grand”)  
Else:  
    print(“Erreur de saisie”)
- **While (tant que) : boucle indéfinie**  
condition=True  
while(condition):  
    nbr=input(“entrer un nombre en 1 et 10”)  
    If nbr>=1 and nbr<=10:  
        condition=False
- **For : boucle définie**
  - for i in range(5): #i : valeur de 0 à 4
  - For el in element: # el: element de la liste

# Entrées - Sortie



- **Print : sert à afficher des messages**
- **Le caractère b permet d'écrire une suite de bytes**
  - `print(b"Envoie de bytes")`
  - ceci est utilisé lorsque des programmes communiquent
  - `sys.getsizeof(b'envoie de bytes')`
  - `sys.getsizeof("envoie de bytes")`
- **Le caractère f permet de formater les informations introduites dans le print**
  - `print (f"Ce texte est écrit par {name}")`
  - Ici name est une variable qui va être interprétée grâce aux accolades
- **Input: permet à l'utilisateur de rentrer des messages**
- `>>> message=input("Entrer le nom de la personne")`
- **La variable récupérée est une chaîne de caractères.**