

TP1 : prise en main de Python

```
In [ ]: # Les bonnes pratiques :
# ajouter des commentaires
# donner un nom significatif pour les variables, les fonctions, les fichiers, etc.
```

```
In [ ]: # **Généralités**
```

```
In [ ]: # la déclaration de variables est simple sans initialisation
a=10
print(a)
b=2
print(a)
c,d=3,4
print(c)
print(d)
```

```
In [ ]: #Les opérations standards
c=a+b
print(c)
c=a-b
print(c)
d=a*b
print(d)
e=a/b
print(e)
```

```
In [ ]: # affichage avec print
euro=1
print(euro)

print('euro : ',euro)
print('euro1 : ',euro,'et', 'euro2',2*euro)

print('\n')
print('-----')
chaine='Bonjour' + ' Mr ' + 'Toto'
print(chaine)
print('-----')
```

```
In [ ]: #types
a=1
print(type(a))

a=1.0
print(type(a))

a='coucou'
print(type(a))
```

```
In [ ]: #des commandes système
# ls, pwd, who
# !python --version
```

Structures de données

Les listes

```
In [ ]: tab=[2*3, 5, 7, 1/3, 0]
print(tab)
type(tab)
```

```
In [ ]: listel = [3.14000, 'Hello']
print(listel)
print(listel[0])
print(listel[1])
```

```
In [ ]: liste2 = [2, 4.1, "avec ... le soutien de", listel]
print(liste2)
```

```
In [ ]: print(liste2[2])
```

```
In [ ]: print(liste2[3][1])
```

```
In [ ]: #longueur d'une liste
len1=len(listel)
print(len1)
len2=len(liste2)
print(len2)
```

```
In [ ]: #le dernier élément d'une liste
print(listel[-1])
```

```
In [ ]: #sélection d'une partie
liste=liste2[0:3]
print(liste)
```

```
In [ ]: #création automatique de quelques listes
listel=range(0,20,5)
print(listel[0],listel[1],listel[-1])
```

```
In [ ]: type(listel)
```

```
In [ ]: # opération entre les listes
a=[1, 2, 3]
b=[0, 2, 4]
print(a+b)
```

```
In [ ]: # ajouter des éléments à la fin d'une liste
c=a+b
c.append(10)
print(c)
```

```
In [ ]: # insérer un élément
c.insert(0,100)
c.insert(len(c),1000)
print(c)
```

```
In [ ]: #enlever un élément
c.remove(1000)
print(c)

#autres opérations : file, pile
# extend, pop, clear,del, copy, count, sort ....
```

Outils de contrôle

```
In [ ]: # instruction if
x = int(input("Veuillez entrer un entier : "))
print(x)
if x>5:
    print('bien')
elif x==5:
    print('ok')
else:
    print('pas bien')
```

```
In [ ]: # instruction for
iter=1
liste=['chat', 'chien','ours','cheval']
for chaine in liste:
    print(iter, 'eme élément',chaine, 'de longueur',len(chaine))
    iter+=1
```

```
In [ ]: # la fonction range
for i in range(5):
    print(i**2)
```

```
In [ ]: liste = ['Alice', 'a', 'plus', 'de chance', 'que', 'Bob']
for i in range(len(liste)):
    print(i, 'eme élément : ',liste[i])
```

```
In [ ]: print(sum(range(5)))#0+1+..4
```

```
In [ ]: # Les instructions break, continue
#et les clauses else au sein des boucles
for n in range(1,10,3):
    for x in range(2, n):
        if n % x == 0:
            print(n, 'égale', x, '*', n//x)
            break
    else:
        print(n, 'est un nombre premier')
```

Les fonctions

```
In [ ]: #définition d'une fonction
def fct_somme(a,b):
    return(a+b)

#Appel de la fonction
print(fct_somme(2,3))
c=fct_somme(2,3)
print(c)
f=fct_somme
f(1,1)

#Les fonctions sont très utiles. Une partie plus détaillée sera
#présentée la semaine prochaine
```

Tuples et séquences

```
In [ ]: t = 12345, 54321, 'miroir!'
print(t[0])
type(t)
```

```
In [ ]: #t[0]=1000
```

Les ensembles

```
In [ ]: Panier={'orange', 'poire', 'banane', 'kiwi'}
print(Panier)
#print(Panier[0])
```

```
In [ ]: 'poire' in Panier
```

```
In [ ]: chainel=set('abracadabra')
print(chainel)
chaine2 = set('alacazam')
print(chainel & chaine2)
print(chainel-chain2)
```

```
In [ ]: a = {x for x in 'abracadabra' if x not in 'abcy'}
print(a)
```

Les dictionnaires

```
In [ ]: data={'Alice': 2010, 'Bob': 2011}
print(data)
print(data['Alice'])
```

```
In [ ]: list(data)
```

```
In [ ]: 'Alice' in data
```

```
In [ ]: {x: x**2 for x in (2, 4, 6)}
z={y: 1/y for y in (1, 2, 3)}
print(type(z))
print(z)
```

```
In [ ]: z={y for y in range(5)}
print(type(z))
print(z)
```

```
In [ ]: prenom=['Alice', 'Bob', 'autre']
age=[19,18,21]
data={prenom[x]: age[x] for x in range(len(prenoms))}
print(data)
```

```
In [ ]: for i,j in data.items():
    print(i, 'a',j, 'ans')
```

```
In [ ]: questions = ['nom', 'prenom', 'age']
reponses = ['Marcelo', 'Alice', '19']
for i,j in zip(questions,reponses):
    print(i,':',j)
```

Pratique

Exercice 1 : manipulation des listes

- 1- Créer une liste **liste1** qui contient les entiers entre 1 et 100
- 2- Inverser l'ordre des éléments de **listes1**
- 3- Créer une liste **listes2** qui contient autant d'éléments que **listes1**
- 4- Créer une liste **listes3** qui contient la somme de **listes1** et **listes2**
- 5- Calculer le maximum de **listes3**
- 5- Définir une fonction qui calcule le maximum d'une liste

Exercice 2 : manipulation des tuples

- 1- Créer un tuple **tuple1** qui contient les entiers entre 1 et 5 inclus
- 2- Créer un tuple **tuple2** de chaînes de caractères qui contient autant d'éléments que **tuple1**
- 3- Créer un tuple **tuple3** qui contient **tuple1** et **tuple2**
- 4- Définir une fonction qui calcule le minimum de **tuple1**
- 5- Expliquer la différence entre un tuple et une liste

Exercice 3 : manipulation des ensembles

- 1- Créer un ensemble **ensemble1** qui contient les alphabets entre b et h inclus
- 2- Créer un ensemble **ensemble2** d'entiers qui contient autant d'éléments que **ensemble1**
- 3- Enlever les alphabets d'indices paires dans **ensemble1**
- 4- Enlever les entiers impaires dans **ensemble1**
- 5- Ecrire une fonction qui prend en entrée un ensemble et enlève toutes les voyelles

Exercice 4 : manipulation des chaînes de caractères

- 1- Créer une chaîne de caractères **chaine1** qui contient la phrase **c'est que j'aime Python et le module aussi**
- 2- Afficher **chaine1** avec un retour à la ligne
- 3- Afficher **chaine1** avec un mot par ligne en utilisant la méthode split
- 4- Enlever les **e** dans chaine1 et afficher le résultat
- 5- Ecrire une fonction qui prend en entrée une chaîne de caractères et compte le nombre des alphabets

Exercice 5 : manipulation des dictionnaires

- 1- Créer un tuple **tuple1** qui contient les noms différents de 5 personnes minimum. Les noms seront entrés à l'aide de input.
- 2- Créer un tuple **tuple2** qui contient les prénoms différents d'autant de personnes que dans **tuple1**. Les prénoms seront entrés à l'aide de input.
- 3- Former un dictionnaire **dict1** de noms et prénoms à partir de **tuple1** et **tuple2**
- 4- Garder l'ordre des noms et changer l'ordre des prénoms
- 5- Créer une copie **dict2** indépendante de **dict1**
- 6- Ajouter une personne de nom **Toto** et de prénom **Hello** dans dict2
- 7- Ecrire une fonction qui prend en entrée deux dictionnaire et un booléen verbose pour afficher :
 - * verbose est vrai : ce qui est commun
 - * verbose est faux : ce qui strictement différent