

```
#####
#####TP-TD2 :Opérateurs/Types numériques/fonctions E_S#####
#####
```

On va travailler sur "python shell" (ou "mode calculatrice" ou "mode interactif")

```
#####
#####Les types#####
#####
```

```
>>> type(10)
```

```
-----
```

```
>>> type(10.5)
```

```
-----
```

```
>>> type('jhkhkjkhkj')
```

```
-----
```

```
>>> type([1,2,3])
```

```
-----
```

```
>>> type(True)
```

```
-----
```

```
>>> type(1+1j)
```

```
-----
```

```
>>> type(10)
```

```
-----
```

```
>>> type('10')
```

```
-----
```

```
#la fonction "int"
```

```
>>> int('10')
```

```
-----
```

```
>>> int('lk1k')
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
ValueError: -----
```

```
#la fonction "float"
```

```
>>> float('10dsffsd')
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
ValueError: -----
```

```
>>> float('10')
```

```
-----
```

```
>>> float(10)
```

```
-----
```

```
>>> float('10.2')
```

```
-----
```

```
#la fonction "str"
```

```
>>> str(10)
```

```
-----
```

```
>>> str(10.5)
```

```
-----
```

```
>>> str([1,2,3])
```

```
-----
```

```
>>> str([1,2,3])
```

```
-----
```

```
#la fonction "bool"
```

```
>>> bool(1)
```

```

-----
>>> bool(0)
-----
>>> bool('jhkjkhkjkj')
-----
>>> bool(120)
-----
>>> bool([1,2,3])
-----
>>> bool('jlk1k')
-----
>>> bool('')
-----
>>> e={1,2,3}

```

Déf: 'un ensemble' est une collection d éléments (de même type) et (distincts 2 à 2)

```

>>> {1,2,3,3}
-----
>>> {}
-----

>>> 'hjkjjkhkj'
-----
>>> ''
-----
>>> [1,2,'bjr']
-----
>>> []
-----
>>> 10.5
-----
>>> None
>>> bool('')
-----
>>> bool([])
-----
>>> bool(None)
-----
>>> type([])
-----
>>> type(None)
-----

```

```

#la fonction "complex"
>>> complex('1')
-----
>>> complex('j')
-----
>>> 1j
-----
>>> j
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: -----
>>> j=10
>>> j+1j*j

```

```

-----
>>> complex(1)
-----
>>> complex(10.5)
-----
>>> complex(10.5)
-----

#la fonction "list"
>>> list('vcxcvcxv')
-----
>>> l=[1,2,10,True,'bjr',[0,'a']]
>>> l[0]
-----
>>> l[1]
-----
>>> l[5]
-----
>>> len(l)
-----
>>> l[5]
-----
>>> l[len(l)-1]
-----
>>> l[len(l)-2]
-----
>>> l[-1]
-----
>>> l[-2]
-----
>>> l[len(l)-2]
-----
>>> l
-----

>>> l
-----
>>> l[0]
-----
>>> l[1]
-----
# indexée=ordonnée
# parcourir une liste= balayer une liste=itérer une list cad on peut y appliquer
boucle "for"
# Donc une liste est dite "ITERABLE"
#les chaines de caractères ont les même ppts

>>> s='jhhjghjhjggjhgh'
>>> s='bonjour tt le monde'
>>> s[0]
-----
>>> s[-1]
-----
>>> len(s)
-----
>>> s[-1]
-----
>>> s[len(s)-1]
-----

```

```

>>> s[-2]
-----
>>> #s est ordonnée
>>> #s itérable
>>> list(1)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: -----
>>> list('hkhjhkhkhkh')
-----
>>> [1,2,3,5]
-----
#####
#####Les opérateurs#####
#####

##### Opérateurs arithmétiques:+,*,/,//,% #####
Exemple:
10+5 # c'est une opération. le '+' est un opérateur arithmétique dit: "l'addition".
10 et 5 sont les opérants

>>> 10+5 # c'est une opération. le '+' est un opérateur arithmétique dit:
"l'addition".
10 et 5 sont les opérants
15
>>> 10*5
-----
>>> 15/20
-----
>>> 10//3
-----
>>> 10%3
-----

##### Opérateurs de comparaisons: <,>,<=,>=,==,!= #####
1<3 est dite "une comparaison". Le resultat est de type booléen (True/False)

>>> 1<3
-----
>>> 10+5
-----

>>> True+True
-----
>>> True+True*10
-----
>>> False*1111
-----
>>> True==1
-----
>>> False==0
-----
>>> (1<=5)
-----

>>> (1<=5) or (10>8)
-----
>>> (1<=5) and (10>8)
-----

```

```

>>> 3+5.5
-----

#### '+' , selon le type, est une concaténation (pour deux chaines ou 2 listes ou 2
tuples)
>>> 'bonjour'+' tt le monde'
'bonjour tt le monde'
>>> 'bonjour'+10
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: -----
>>> 10.2+15.5
-----
>>> 'bonjour'+' tt le monde'
-----
>>> t=[10,15,'bjr']

>>> type(t)
-----
>>> t
-----
>>> print(t)    #ecrire(t)
-----

>>> [1,2,'bbb']+[10,20,'uu']
-----

#####
#####Les fonctions
d'entrée/sortie#####
#####

####la fonction print

>>> print(t)
-----
>>> print(l)
-----
>>> print(s)
-----
>>> print(s,t,l)
-----
>>> print('salut')
-----
>>> print(100)
-----
>>> print([1,2,3])
-----
>>> print(True)
-----
>>> print('je m appelle=',t)
-----
>>> print('je m appelle=',t,s,10)
-----
>>> print('je m appelle=',t,s,10)
-----
>>> print(t)
-----
>>> print(s)

```

```

-----
>>> print(s,t,l)
-----
>>> print(s,t,l,sep="*")
-----
>>> print(s,t,l,sep="$$$$")
-----
>>> print(s,t,l,sep="/n")
-----
>>> print(s,t,l,sep="\n")
-----
-----
-----
>>> print(s,'\n',t,'\n',l)
-----
-----
-----
-----
-----
##### la fonction "input"

>>> #lire(x) <----> entrer (qq chose par le clavier) ET (hak el valeur 7ottha fi x)
>>> lire()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: -----

lire() #entrer (qq chose par le clavier)

>>> input() #lire()
-----
-----
>>> x=input() #lire(x)
-----
>>> x
-----
>>> type(x)
-----
>>> x=input() #lire(x)
-----
>>> x
-----
>>> type(x)
-----
>>> x=input() #lire(x)
-----
>>> x+10*20
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: -----
>>> x=input() #lire(x)
-----
>>> x
-----
>>> int(x)
-----

```

```

>>> x=int(x)
>>> x
-----
>>> x+10*2001
-----
>>> x=int(input()) #lire(x)
-----
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: -----
>>> x=int(input()) #lire(x)
-----
>>> x
-----
>>> x+10
-----
>>> l=list(input())
-----
>>> l
-----
>>> x=input()
-----
>>> x=input('svp donner qq chose')
-----
>>> x=int(input(donner un entier)) #lire(x)
  File "<stdin>", line 1
    x=int(input(donner un entier)) #lire(x)
                  ^
SyntaxError: -----
>>>
>>> f(g(h(x)))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: -----
>>> x=int(input('donner un entier')) #lire(x)
-----
>>> x=float(input('donner un reel')) #lire(x)
-----
>>> x=str(input('donner un reel')) #lire(x)
-----
>>> str('110')
-----
>>> int(10)
-----

```