
	TD Révisions Correction	
---	--	---

1. Numération

Valeur décimale	Valeur binaire	Valeur hexadécimale
23	10111	17
55	110111	37
973	1111001101	3CD
128	10000000	80
426	110101010	1AA
2046	011111111110	7FE

2. Les boucles

2.2.1.

```
operande=7
for n in range(1,21):
    print(operande,"*",n,"=",n*operande)
```

2.2.2.

```
operande=7
for n in range(1,21):
    multiplication=n*operande
    if multiplication%3==0:
        print(multiplication,end='* ')
    else:
        print(multiplication,end=' ')
```

2.2.3.

```
euros=1
for n in range(0,11):
    print(euros,'euro(s) =', euros*1.12,'dollar(s)')
    euros*=2
```

2.2.4.

```
nombre=1
for n in range(0,12):
    print(nombre)
    nombre*=3
```

3. Les Fonctions

3.2.1.

#TF=TC ×1,8+32

#TC=(TF-32)/1.8

```
def conversion(tf):
```

```
    tc=(tf-32)/1.8
```

```
    return tc
```

```
tf=60
```

```
tc=conversion(tf)
```

```
print('la température de',tf,'°F est de',tc,'°C')
```

3.2.2.

```
def nomMois(n):
```

```
    mois=['janvier','février','mars','avril','mai','juin','juillet','août','septembre','octobre','novembre','décembre']
```

```
    return mois[n-1]
```

```
print(nomMois(4))
```

3.2.3.

```
def lettre(N):
```

```
    if N>=80:
```

```
        lettre='A'
```

```
    elif N>=60 and N <80:
```

```
        lettre='B'
```

```
    elif N>=50 and N <60:
```

```
        lettre='C'
```

```
    elif N>=40 and N <50:
```

```
        lettre='D'
```

```
    elif N <40:
```

```
        lettre='E'
```

```
    return lettre
```

```
for n in range(0,100):
```

```
    print(n,lettre(n))
```

3.2.4.

```
a=int(input("longueur du coté a"))
b=int(input("longueur du coté c"))
c=int(input("longueur du coté c"))

def triangle(a,b,c):
    if a==b and a==c:
        nom='equilateral'
    elif (a==b and a!=c) or (a==c and a!=b) or (c==b and c!=a):
        nom='isocele'
    else:
        nom='quelconque'
    return nom

if (a+b)>c and (a+c)>b and (b+c)>a:
    nom=triangle(a,b,c)
    print('triangle',nom)
else:
    print("ce n'est pas un triangle")
```

3.3. Affichage triangle

```
def triangle(hauteur):
    for h in range(hauteur,0,-1):
        for c in range(1,h+1):
            print('*',end='')
        print()

triangle(5)
```

4 Les Listes

4.2.1.

```
def moyenneVersion1(liste):
    total=0
    for n in range(0,len(liste)):
        total=total+liste[n]
    moy=total/len(liste)
    return moy
```

```
lst=[20,8,9,2,35,49]
print(moyenneVersion1(lst))
```

4.2.2.

```
listeSource=[1,5,6,1,3,5,7,8]
```

```
listeDestination=[]
```

```
for n in range(len(listeSource)):
    if listeSource[n] not in listeDestination:
        listeDestination.append(listeSource[n])
```

```
listeDestination.sort()
```

```
print(listeSource)
print(listeDestination)
```

5 Les chaines de caractères

5.2.1.

```
listePrenom=['Jean-Michel', 'Marc', 'Vanessa', 'Anne', 'Maximilien','Alexandre-Benoît', 'Louise']
for n in range(len(listePrenom)):
    print(listePrenom[n],len(listePrenom[n]))
```

ou

```
listePrenom=['Jean-Michel', 'Marc', 'Vanessa', 'Anne', 'Maximilien','Alexandre-Benoît', 'Louise']
for prenom in listePrenom:
    print(prenom,len(prenom))
```

5.2.2.

```
chaine="Bonjour, il fait beau aujourd'hui"
```

```
def recherche(chaine,car):
    trouve=False
    for n in range(len(chaine)):
        if chaine[n]==car:
            trouve=True
    return trouve
```

OU

```
chaine="Bonjour, il fait beau aujourd'hui"
def rechercheV2(chaine,car):
```

```
    if car in chaine:
        trouve=True
    else:
        trouve=False
    return trouve
```

```
print(recherche(chaine,'e'))
print(rechercheV2(chaine,'e'))
```

5.2.3.

```
chaine="Bonjour, il fait beau aujourd'hui, n'est-ce pas ?"
```

```
def compte(chaine,car):  
    total=0  
    for n in range(len(chaine)):  
        if chaine[n]==car:  
            total+=1  
    return total
```

```
print(compte(chaine,'e'))
```

5.2.4.

```
def recopie(chaine):  
    destination=""  
    for n in range(len(chaine)):  
        if n<len(chaine)-1:  
            destination=destination+chaine[n]+'*'  
        else:  
            destination=destination+chaine[n]  
    return destination
```

```
print(recopie('toto'))
```

5.2.5.

```
phrase="Python est un langage de programmation interprété"  
#phrase=input('votre phrase?')
```

```
def lePlusLong(chaine):  
    mots= chaine.split(' ')  
    print(mots)  
    maxLettre=len(mots[0])  
    for n in range(1,len(mots)):  
        if len(mots[n])>maxLettre:  
            maxLettre=len(mots[n])  
            indice=n  
    return mots[indice]
```

```
print(lePlusLong(phrase))
```

6. Les dictionnaires

6.2.

```
def occurrences(chaine) :
```

```
    D = {}
```

```
    for caractere in chaine :
```

```
        if caractere in D.keys() :
```

```
            D[caractere] = D[caractere] + 1
```

```
        else :
```

```
            D[caractere] = 1
```

```
    return D
```

```
print(occurrences('tortue'))
```

7. Rendu de monnaie

7.2.

```
def renduMonnaie(somme,pieces):
```

```
    choisies = {}
```

```
    for p in pieces:
```

```
        nb = somme // p
```

```
        choisies[p] = nb
```

```
        somme = somme - nb * p
```

```
    return choisies
```

```
#pieces en centimes d'euros
```

```
pieces=[500,200,100,50,20,10,5,2,1]
```

```
somme=780
```

```
print('Les pièces choisies sont')
```

```
print(renduMonnaie(somme,pieces))
```

8. Calcul d'une dimension (Voir document cours)

9. Rechercher une valeur dans un tableau (Voir document cours)

10. Tri par sélection (Voir document cours)

11. Recherche dichotomique (Voir document cours)