

Commissariat régionale de l'enseignement de Tataouine/ Médenine

Formation en Python au profil des enseignants des classes de 2^{ème} Année
Scientifiques, Sciences Informatiques
Année Scolaire : 2019 - 2020



Tâche 1

Ecrire un programme en Python qui convertit et affiche en heures (**H**), en minutes (**Min**) et en secondes (**S**), une durée de temps (**T**) donnée en secondes.

Solution :

```
T=eval(input("saisir un temps en seconde : "))
H=T//3600
Min=T%3600//60
S=T%3600%60
print("Le nombre d'heures = ", H)
print("Le nombre de minutes = ", Min)
print("Le nombre de seconde = ", S)
```

Tâche 2

Ecrire un programme en Python qui permet de calculer et d'afficher le minimum (**Min**) et le maximum (**Max**) de deux nombres (**A**) et (**B**) donnés en utilisant seulement les opérateurs arithmétiques et logiques.

Solution :

```
A=eval(input("Saisir A : "))
B=eval(input("saisir B : "))
Max=int((A>=B))*A+int((A<B))*B
Min=int((A<=B))*A+int((A>B))*B
print("Le maximum est = ", Max)
print("Le minimum est = ", Min)
```

Tâche 3

Ecrire un programme en Python qui permet de :

1. Saisir un caractère **Ca**,
2. Convertir le caractère **Ca** en majuscule,
3. Chercher le rang **R** du caractère **Ca** dans l'alphabet majuscule.
4. Calculer la somme **S** en ajoutant à **R** une constante (**G** = 100).
5. Chercher la somme **S1** qui est égal à la somme des carrés des chiffres de **S**
6. Afficher **S1**.

Exemple : **Ca** = "f" , **Ca** = "F" donc **R** = 6 , **S** = 106 , **S1** = $1^2 + 0^2 + 6^2 = 37$

Solution :

```
G=100
Ca=input("Saisir un caractère Ca : ")
Maj= Ca.upper()
R=ord(Maj)-65+1
S=R+G
S1=(S//100)**2+(S%100//10)**2+(S%10)**2
```

print(S1)**Tâche 4**

Ecrire un programme en Python qui permet de lire une chaîne de caractère **Ch** et d'afficher sa longueur, son premier caractère, le caractère du milieu, son dernier caractère, les 3 premiers et les 3 derniers caractères de la chaîne **Ch**.

On supposera que la longueur de la chaîne de caractère **Ch** est impaire.

Solution :

```
Ch=input("Saisir la chaîne : ")
L=len(Ch)
m=L//2
print("Le premier caractère est = ", Ch[0])
print("Le caractère du milieu est = ", Ch[m])
print("Le dernier caractère est = ", Ch[L-1])
print("Les trois premiers caractères sont = ", Ch[:3])
print("Les trois derniers caractères sont = ", Ch[-3:])
```

Tâche 5

Ecrire un programme en Python qui permet d'afficher la parité d'un nombre entier (**N**) donné.

Solution :

```
N=int(input("Saisir N : "))
if (N%2==0) :
    print(N," est pair ")
else :
    print(N," est impair ")
```

Tâche 6

Ecrire un programme en Python qui permet de saisir un nombre **N** puis d'afficher s'il est un entier ou un réel.

Solution :

```
N=eval(input("Saisir N : "))
if (int(N)==N) :
    print(N," est un entier ")
else :
    print(N," est un réel ")
```

Tâche 7

Ecrire un programme en Python qui permet de lire une chaîne **Ch** puis d'afficher « **Chaîne numérique** » dans le cas où la chaîne est constituée uniquement par des caractères chiffre et « **Chaîne Autre** » dans le cas contraire.

Solution :

```

Ch=input("Saisir la chaîne : ")
if (Ch.isdecimal()):
    print(Ch ," est une Chaîne numérique ")
else :
    print(Ch ," est une chaîne autre ")

```

Tâche 8

Ecrire un programme en Python qui permet d'afficher le nombre de jours **Nj** d'un numéro du mois **Nm** donné.

Solution :

```

Nm=int(input("Saisir le mois : "))
if(Nm in (1 , 3 , 5 , 7 , 8 , 10 , 12)):
    print("Le nombre de jour est = 31")
else :
    if (Nm in (4 , 6 , 9 , 11)) :
        print("Le nombre de jour est = 30")
    else :
        if (Nm==2) :
            a=int(input("Saisir l'année : "))
            if (a%4==0) :
                print ("Le nombre de jour est = 29")
            else :
                print ("Le nombre de jour est = 28")

```

Tâche 9

Ecrire un programme en Python qui permet de calculer et d'afficher la somme des **N** premiers termes de la suite **Sn** suivante :

$$S_n = 1 + 2 + 3 + 4 + 5 + 6 + \dots + N \quad \text{Avec } N \text{ un entier donné et supérieur à 2.}$$

Solution :

```

N=int(input("Saisir N : "))
while N<= 2 :
    N=int(input("Saisir N : "))
Sn=0
for i in range(N+1) :
    Sn=Sn+i
print("La somme = ", Sn)

```

Tâche 10

Ecrire un programme en Python qui permet d'afficher la factorielle d'un nombre **N** donné (**N** doit être supérieur ou égal à **0**).

Solution :

```
N=int(input("Saisir N : "))
while N< 0 :
    N=int(input("Saisir N : "))
f=1
for i in range(2 , N+1) :
    f=f*i
print("La factorielle est = ", f)
```

Tâche 11

Ecrire un programme en Python qui permet d'afficher le nombre de voyelles **Voy** et le nombre de consonnes **Con** d'un texte **ch** donné.

Noter bien : Le programme ne fait pas de distinction entre caractère alphabétique majuscule et caractère alphabétique minuscule.

Solution :

```
Ch=input("Saisir la chaine de caractères : ")
nbvoy=0
nbcon=0
Ch1=Ch.upper()
for c in Ch1 :
    if (ord(c) in range(ord("A") , ord("Z"))):
        if (c in "OIYEAU") :
            nbvoy=nbvoy+1
        else :
            nbcon=nbcon+1
print("Dans la chaine ", Ch , " existe " , nbvoy , " voyelles")
print("Dans la chaine ", Ch , " existe " , nbcon , " consonnes ")
```

Tâche 12

Ecrire un programme en Python qui permet de calculer et d'afficher les occurrences de toutes les lettres alphabétiques constituant la phrase **Ph** saisie.

Solution :

```
ph=input("Saisir la phrase : ")
ph1=ph.upper()
for i in range(26) :
    c=chr(65+i)
    nbc=0
    for i in range(len(ph)) :
        if (c==ph1[i]) :
            nbc=nbc+1
    if (nbc!=0) :
        print(c , " existe " , nbc , " fois")
```

Tâche 13

Ecrire un programme en Python qui permet de remplir un tableau **T** par **N** moyennes avec (**N** > 1) puis de calculer et d'afficher leur somme et leur moyenne.

Solution :

```
T = []
N = int(input("Saisir la taille du tableau : "))
while (N < 1) :
    N = int(input("Saisir la taille du tableau : "))
for i in range(0, N) :
    x = float(input("saisir la moyenne N° " + str(i+1) + " : "))
    while ((x > 20) | (x < 0)) :
        x = float(input("saisir la moyenne N° " + str(i+1) + " : "))
    T.append(x)
s = 0
for i in range(0, N) :
    s = s + T[i]
print("la somme est = ", s)
print("la moyenne est = ", s/N)
```

Tâche 14

Ecrire un programme en Python qui permet de remplir aléatoirement un tableau **T** par **10** entiers appartenant à l'intervalle [**5** , **200**] puis de les afficher.

Solution :

```
from random import *
T = [randint(5, 200) for i in range(10)]
print(T)
```

Tâche 15

Ecrire un programme en Python qui permet de :

- ❖ remplir un tableau **T** par **N** chaînes de caractères non vide (**5** ≤ **N** ≤ **20**),
- ❖ saisir une chaîne de caractères **Ch** non vide,
- ❖ afficher le message « **Existe** » si la chaîne de caractères **Ch** existe dans le tableau ou le message « **N'existe pas** » dans le cas contraire.

Solution :

```
T = []
N = eval(input("Saisir un entier entre 5 et 20 : "))
while (not N in range(5, 20)) :
    N = eval(input("saisir un entier entre 5 et 20 : "))
for i in range(0, N) :
    ch1 = ""
    while (ch1 == "") :
        ch1 = input("saisir l'élément N° " + str(i+1))
    T.append(ch1)
Ch = ""
while (Ch == "") :
```

```

Ch = input("Saisir une chaine non vide : ")
existe = False
i = -1
while ( i < N-1 ) & ( existe == False ) :
    i = i + 1
    if ( T[i] == Ch ) :
        existe = True

if ( existe == True ) :
    print( Ch , "Existe dans T" )
else :
    print ( Ch , "N'existe pas dans T")

```

Tâche 16

Ecrire un programme en Python qui permet de calculer et d'afficher le nombre de combinaisons de p objets parmi n. n et p sont deux entiers strictement positifs (avec $n \geq p$).

$$\text{N.B : } C_p^n = \frac{n!}{p!(n-p)!}$$

N.B : on doit utiliser au moins deux modules.

Solution :

```

def Saisie_p () :
    global p
    p=0
    while ( p <= 0 ) :
        p = eval( input("saisir un entier p>0 : "))

def Saisie_n ( p ) :
    global n
    n = 0
    while ( n <= 0 or n <= p ) :
        n = eval(input("saisir un entier n >0 et n>= : " + str(p)+ " "))

def factorielle ( n ) :
    f = 1
    for i in range ( 2 , n+1 ) :
        f = f * i
    return f

Saisie_p()
Saisie_n(p)
x = factorielle( n ) / ( factorielle( p ) * factorielle( n - p ))
print ("La combinaison de " , n , " parmi " , p , " est = " , x )

```

1 Annexes

Opérateur	Nomination	Effet	Exemple	Résultat (avec x = 7)
+	Addition	Ajouter deux valeurs	x+3
-	Soustraction	Soustraire deux valeurs	x-3
*	Multiplication	Multiplier deux valeurs	x*3
/	Division réelle	Diviser deux valeurs	x/3
%	Reste de la division entière	Le reste de la division entière	x%3
//	Quotient de la division entière	Le quotient de la division entière	x//3
**	La puissance	Retourne x à la puissance n	x**2

Les méthodes de formules mathématiques

Méthode	Description
<code>abs (a)</code>	Retourne la valeur absolue de a
<code>round (a)</code>	Retourne l'entier arrondi le plus proche de a
<code>divmod (a , b)</code>	Retourne un couple formé par le quotient et le reste de la division entière (Q, R)
<code>pow (a , b)</code>	Retourne a à la puissance b
<code>min (a , b)</code>	Retourne le plus petit des paramètres a ou b
<code>max (a , b)</code>	Retourne le plus grand des paramètres a ou b

Les méthodes de formules mathématiques

Méthode	Description	Bibliothèque
<code>sqrt (a)</code>	Retourne la racine carrée de a	<code>from math import *</code>
<code>randrange(a,b)</code>	Retourne aléatoirement un entier entre [a,b[<code>from random import *</code>
<code>randint(a,b)</code>	Retourne aléatoirement un entier entre [a,b]	<code>from random import *</code>
<code>random.random()</code>	Retourne aléatoirement un réel entre [0,1[<code>from random import *</code>

Les opérateurs logiques (booléens)

Opérateur	Nomination	Effet	Syntaxe
ou or	OU logique	Vérifie qu'une des conditions est réalisée	((condition1) (condition2))
& ou and	ET logique	Vérifie que toutes les conditions sont réalisées	((condition1) & (condition2))
is not	NON logique	Inverse l'état d'une variable booléenne	(is not condition)

Les opérateurs de comparaison				
Opérateur	Dénomination	Effet	Exemple	Résultat (avec x valant 7)
== à ne pas confondre avec le signe d'affectation (=)!!	opérateur d'égalité	Compare deux valeurs et vérifie leur égalité	x == 3
<	opérateur d'infériorité stricte	Vérifie qu'une variable est strictement inférieure à une valeur	x < 3
<=	opérateur d'infériorité	Vérifie qu'une variable est inférieure ou égale à une valeur	x <= 3
>	opérateur de supériorité stricte	Vérifie qu'une variable est strictement supérieure à une valeur	x > 3
>=	opérateur de supériorité	Vérifie qu'une variable est supérieure ou égale à une valeur	x >= 3
!=	opérateur de différence	Vérifie qu'une variable est différente d'une valeur	x != 3
in (liste de valeur)	dans	Vérifier que la valeur d'une variable dans une liste de valeurs	x in (1, 2, 8)
in range (Borne inf , Borne sup)	dans	Vérifier que la valeur d'une variable existe dans un intervalle de borne inférieur à borne supérieur - 1 ces deux bornes doivent être des entiers.	x in range (8,50) x = 50

Les méthodes prédéfinies sur les chaînes de caractères dans python

La méthode	Syntaxe	Rôle	Exemple
len	len (chaîne)	Détermine la longueur d'une chaîne de caractères c.a.d. le nombre de caractères constituant la chaîne	<pre>ch = "cours python" L = len (ch) print ("la longueur " , L)</pre> Le programme affiche 12
find	Chaîne.find (chaîneR , position initiale gauche , position finale droite) Position initiale et Position finale sont facultatives	Chercher une chaîne de caractères chaîneR dans une chaîne de caractères chaîne de la gauche vers la droite	<pre>⇒ chaîne = "Cours python 2018-2019" chaîneR = "python" p = chaîne.find (chaîneR) print (chaîneR , " existe à la position = " , p)</pre> le programme affiche : chaîneR existe à la position = 6 <pre>⇒ chaîne = "Cours python 2018-2019" chaîneR = "Python" p = chaîne.find (chaîneR) print (chaîneR , " existe dans la position = " , p)</pre> le programme affiche -1 car la chaîne Python n'existe pas
upper	chaîne.upper ()	Convertit une chaîne en majuscule	<pre>chaîne = "cours python" ch = chaîne.upper () print (" La chaîne en majuscule = " , ch)</pre> le programme affiche : COURS PYTHON
Capitalize ()	chaîne.capitalize ()	Convertit le premier caractère en majuscule	<pre>chaîne = "cours python et cours programmation 2018-2019" chaîne1 = chaîne.capitalize () print (chaîne1)</pre> le programme affiche : Cours python et cours programmation 2018-2019
lower	chaîne.lower ()	Convertir une chaîne en minuscule	<pre>chaîne = "COURS PYTHON" chaîne = chaîne.lower () print (" La chaîne en minuscule = " , chaîne)</pre> le programme affiche cours python
strip	chaîne.strip ()	enlève les espaces éventuels au début et à la fin de la chaîne	<pre>chaîne = " Cours python et cours programmation 2018-2019 "</pre> <pre>chaîne1 = chaîne.strip () print (chaîne1)</pre> Le programme affiche : Cours python et cours programmation 2018-2019

replace	chaine.replace (ancienne_chaine , nouvelle_chaine)	Remplacer toutes les anciennes chaînes de caractères par les nouvelles chaînes caractères.	chaine = "cours python et cours programmation 2018-2019" chaine1 = chaine.replace ("cours" , "formation") print (chaine1) Le programme affiche : formation python et formation programmation 2018-2019
str	str (nb)	Convertie une valeur numérique nb en une chaîne de caractères.	nb = 1235 Ch = str (nb) print (Ch) Le programme affiche "1235"
ord	ord (car)	Renvoie le code Ascii du caractère car. Le résultat est un entier positif.	⇒ ord ("A") = 65 ⇒ ord ("a") = 97
chr	chr (x)	Renvoie le caractère dont le code ASCII est x.	⇒ chr (65) = "A" ⇒ chr (48) = "0"
isalnum	Chaine.isalnum ()	Retourne Vrai si la chaine contient uniquement des lettres alphabétiques et des chiffres	⇒ ch = "Mohamed5" print (ch.isalnum ()) Le programme affiche : True ⇒ ch = "13.5" print (ch.isalnum ()) Le programme affiche : False
isalpha	Chaine.isalpha ()	Retourne Vrai si la chaine contient seulement des lettres alphabétiques	⇒ Ch = "Lycée" print (Ch.isalpha ()) Le programme affiche : True ⇒ Ch = "Bonne année" print (Ch.isalpha ()) Le programme affiche : False
Isdecimal / isdigit	Chaine.isdecimal () ou Chaine.isdigit ()	Retourne Vrai si la chaine contient seulement des caractères chiffres	⇒ s = "28212" print (s.isdecimal ()) Le programme affiche : True ⇒ s = "282.12" print (s.isdecimal ()) Le programme affiche : False
in	chaine1 in "chaine"	Permet de vérifier l'existence une chaine de caractères chaine1 dans une chaine de caractères chaine	chaine = "Cours python et Cours programmation 2018-2019" chaine1 = "Cours" print (chaine1 in chaine) Le programme affiche : True

Les méthodes prédéfinies sur les tableaux dans python

méthode	Syntaxe	Rôle	Exemple
len	len (tableau)	Détermine la longueur d'un tableau	<pre> classe = [7 , 8 , 9 , 1 , "2 Tec info", "2 Sciences" , "2Lettres" , "2 Eco et services" , "2 Mécanique"] L = len (classe) print ("La longueur = " , L) Le programme affiche : La longueur = 9 </pre>
del	del (tableau [indice])	Supprime d'un tableau un élément quelconque (à partir de son index).	<pre> classe = [7 , 8 , 9 , 1 , "2 Tec info", "2 Sciences" , "2Lettres" , "2 Eco et services" , "2 Mécanique"] del (classe [8]) print (classe) Le programme affiche : [7 , 8 , 9 , 1 , '2 Tec info' , '2 Sciences' , '2Lettres' , '2 Eco et services'] </pre>
append	tableau.append (valeur)	Ajouter une valeur à un tableau	<pre> classe = [7 , 8 , 9 , 1 , "2 Tec info", "2 Sciences" , "2Lettres" , "2 Eco et services" , "2 Mécanique"] classe.append ("3 Math") print (classe) Le programme affiche : [7 , 8 , 9 , 1 , '2 Tec info' , '2 Sciences' , '2Lettres' , '2 Eco et services' , '3 Math'] </pre>
insert	tableau.insert (position , valeur)	insérer une valeur à un tableau dans une position donnée	<pre> classe = [7 , 8 , 9 , 1 , "2 Tec info", "2 Sciences" , "2 Lettres" , "3 Math"] classe.insert (7 , "2 Eco et services") print (classe) Le programme affiche : [7 , 8 , 9 , 1 , '2 Tec info' , '2 Sciences' , '2Lettres' , '2 Eco et services' , '3 Math'] </pre>
sort	tableau.sort (reverse = True/False)	Trier un tableau dans l'ordre croissant si reverse = True ou dans l'ordre décroissant si reverse = False	<pre> ⇒ T = [1 , 2 , 3 , 8 , 7 , 2 , 6 , 1] Nb = T.sort (reverse = False) print (T) Le programme affiche : [1 , 1 , 2 , 2 , 3 , 6 , 7 , 8] ⇒ T = [1 , 2 , 3 , 8 , 7 , 2 , 6 , 1] Nb = T.sort (reverse = True) print (T) Le programme affiche : [8 , 7 , 6 , 3 , 2 , 2 , 1 , 1] ⇒ T = [1 , 2 , 3 , 8 , 7 , 2 , 6 , 1] Nb = T.sort () print (T) Le programme affiche : [1 , 1 , 2 , 2 , 3 , 6 , 7 , 8] </pre>
index	tableau.index (valeur)	Retourne la position d'une valeur recherchée dans le tableau	<pre> T = [123 , "Ali" , "Kamel" , "abc"] print ("Indices de Ali : " , T.index ("Ali")) Le programme affiche : Index de Ali : 1 T = [123 , "Ali" , "Kamel" , "abc"] print ("Indices de Ali : " , T.index ("Alia")) Le programme génère une erreur </pre>

2 Les syntaxes des structures de contrôle conditionnelles, des structures itératives, les fonctions et les procédures

⊙ Vocabulaires et syntaxes des structures de contrôle conditionnelles

Forme simple	Forme complète
<pre>if (condition) : instruction1 instructionN</pre>	<pre>if (condition) : instruction1 instruction1N else : instruction2 instruction2N</pre>
Forme généralisée	Forme généralisée 2 ^{ème} méthode
<pre>if (condition1) : instruction1 else : if (condition2) : instruction2 else : if (condition3) : instruction3 else : instructionN</pre>	<pre>if (condition1) : instruction1 instruction1N elif (condition2) : instruction2 instruction2N elif (condition3) : instruction3 instruction3N else : instructionN</pre>

La boucle POUR :

⊙ Vocabulaire et syntaxe :

```
for compteur in séquence :
    Instruction1
    Instruction2
    InstructionN
```

⊙ N.B : séquence peut être :

- ❖ une liste de valeur (valeur1 , valeur2 , valeur3 , valeurN)
- ❖ enumerate(un tableau):
- ❖ un intervalle **range** (Borne Inf , Borne Sup)
- ❖ une chaîne de caractères

La boucle TANTQUE :

⊙ Vocabulaire et syntaxe :

```
while ( condition ) :
    Instruction1
    InstructionN
```

Les Modules

Les fonctions	Les procédures
<pre>def Nom_fonction (Pf1 , ... , Pfn) Liste d'instructions return Résultat</pre>	<pre>def Nom_procedure (Pf1 , , Pfn) Liste d'instructions</pre>