

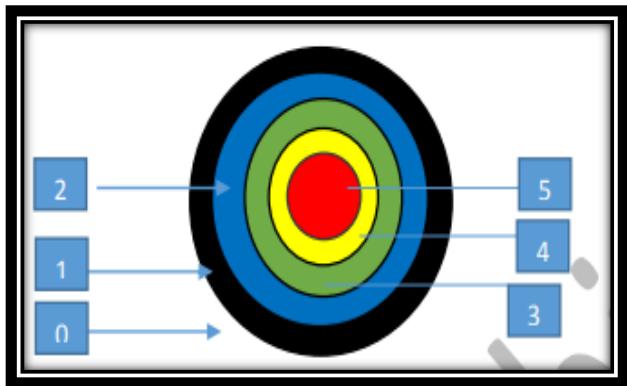
Les enregistrements :

1. Mise en situation :

Le Jeu de tir à l'arc

Principe de jeu :

Le jeu de tir à l'arc consiste à tirer des flèches en utilisant un arc vers un endroit cible. Un club de tir à l'arc organise une compétition entre ses joueurs. Cette compétition consiste à donner à chaque joueur trois 3 essai de tir. A chaque tir, le joueur gagne de 0 à 5 (un nombre aléatoire) points selon l'endroit touché par la flèche sur une cible composée de 5 cercles concentriques comme l'illustre la figure ci-contre.



Après le 3^{ème} essai, un score est calculé pour chaque joueur représentant la somme des résultats de ses trois essais, et ce pour classer les joueurs.

On se propose d'écrire un algorithme puis un programme Python permettant d'afficher les **noms** et les **scores** de 20 joueurs qui ont participé à la compétition :

Question 1: Donner la structure de données convenable aux objets de ce problème.

Réponse 1 : On utilise un tableau pour les noms des joueurs et un autre pour leurs scores

Question 2: Donner une déclaration algorithmique de ces objets

Réponse 2 :

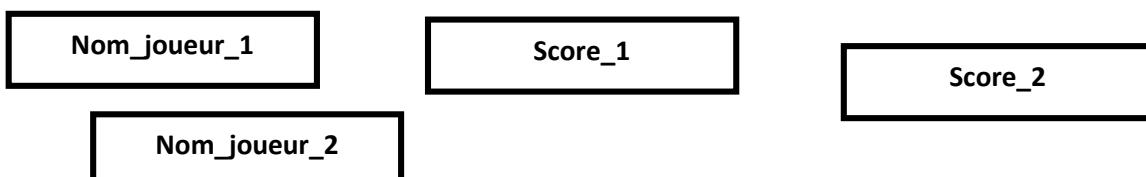
T.D.N.T
Tab1= Tableau de 20 chaines
Tab2=Tableau de 20 entiers

T.D.O	
Objet	Nature / Type
Nom	Tab1
Score	Tab2

Question 3 : Peut-on regrouper ces données dans un même tableau

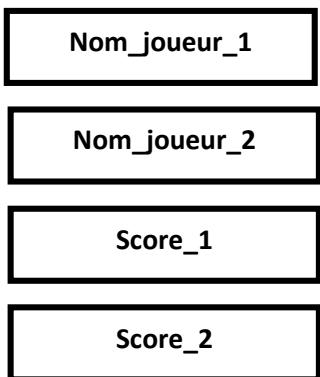
Réponse 3 : Non ; car un tableau ne peut contenir que des données de même type

Remarque : Prenons par exemple les données de deux joueurs ; on aura comme variables

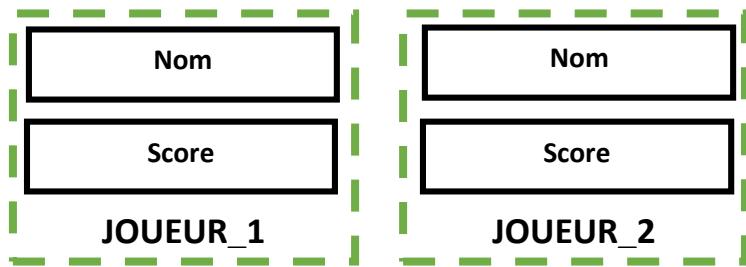


Dans ce cours, nous allons utiliser une nouvelle structure qui permet de regrouper des données hétérogènes dans une même structure : c'est la structure **ENREGISTREMENT**

Sans enregistrement



Avec enregistrement



2. **Définition :** Un enregistrement est une structure de donnée permettant de grouper (ranger) des données hétérogènes (de types différents) dans une même structure

Déclaration d'un enregistrement

Exemple : Déclarer la structure Joueur

En algorithme		En Python						
1^{ère} méthode :								
<table border="1"> <thead> <tr> <th colspan="2">T.D.O</th> </tr> <tr> <th>Objet</th> <th>Nature/Type</th> </tr> </thead> <tbody> <tr> <td>Joueur</td> <td>Enregistrement Nom: Chaine Score : entier Fin</td> </tr> </tbody> </table>		T.D.O		Objet	Nature/Type	Joueur	Enregistrement Nom: Chaine Score : entier Fin	<pre> Joueur = { "Nom" : str(), "Score" : int() } </pre>
T.D.O								
Objet	Nature/Type							
Joueur	Enregistrement Nom: Chaine Score : entier Fin							
2^{ème} méthode :								
<table border="1"> <thead> <tr> <th colspan="2">T.D.N.TYPE</th> </tr> <tr> <th>Objet</th> <th>Nature/Type</th> </tr> </thead> <tbody> <tr> <td>Type_Joueur =</td> <td>Enregistrement Nom: Chaine Score : entier Fin</td> </tr> </tbody> </table>		T.D.N.TYPE		Objet	Nature/Type	Type_Joueur =	Enregistrement Nom: Chaine Score : entier Fin	
T.D.N.TYPE								
Objet	Nature/Type							
Type_Joueur =	Enregistrement Nom: Chaine Score : entier Fin							
<table border="1"> <thead> <tr> <th colspan="2">T.D.O</th> </tr> <tr> <th>Objet</th> <th>Nature/Type</th> </tr> </thead> <tbody> <tr> <td>Joueur</td> <td>Type_Joueur</td> </tr> </tbody> </table>		T.D.O		Objet	Nature/Type	Joueur	Type_Joueur	
T.D.O								
Objet	Nature/Type							
Joueur	Type_Joueur							

3. Syntaxe

1. Déclaration en algorithmique

- ❖ 1^{ère} méthode

Tableau de Déclaration des Objets (T.D.O)	
Objet	Type/Nature
Nom_Enregistrement	Enregistrement Champ1 : Type1 Champ2, Champ3 : Type2 ChampN : TypeM Fin

- ❖ 2^{ème} méthode

Tableau de Déclaration des Nouveaux Types (T.D.N.T)	
Nom_Type_Enregistrement = Enregistrement Champ1 : Type1 Champ2, Champ3 : Type2 ChampN : TypeM Fin	

2. Déclaration en Python

Déclaration en Python

```
Nom_Enregistrement = {}                                « signifie Enregistrement vide »  
Nom_Enregistrement = {  
    Champ1 : Type1,  
    Champ2 : Type2,  
    ChampN : TypeN  
}
```

3. Exemple

Déclarer un enregistrement nommé « Elève » formé de deux champs : « Nom » et « Age ».

- ❖ Déclaration en algorithmique (1^{ère} méthode)

Tableau de Déclaration des Objets (T.D.O)	
Objet	Type/Nature
Elève	Enregistrement Nom : Chaîne Age : Entier Fin

❖ Déclaration en algorithmique (2^{ème} méthode)

Tableau de Déclaration des Nouveaux Types (T.D.N.T)	
Personne = Enregistrement	
	Nom : Chaine
	Age : Entier
Fin	

Tableau de Déclaration des Objets (T.D.O)	
Objet	Type/Nature
Elève	Personne

❖ Déclaration en Python

Elève = {	
	"Nom " : str () ,
	"Age " : int ()
	}

N.B. :

- Un champ d'un enregistrement peut être aussi un enregistrement.
- On peut déclarer des tableaux d'enregistrements.
- L'accès à un champ d'un enregistrement se fait de la manière suivante :

Notation en algorithmique	Notation en Python
Nom_Objet . Champ	Nom_Objet ["Champ"]

4. Manipulation des enregistrements :

<i>Affectation</i>	
<i>En algorithme</i>	<i>En Python</i>
Joueur.Nom ← "Sami"	Joueur["Nom"] = "Sami"
Joueur.Score ← 12	Joueur["Score"] = 12
<i>Lecture</i>	
Lire(Joueur.Nom)	Joueur["Nom"] = input()
Lire(Joueur.Score)	Joueur["Score"] = input()
<i>Ecriture</i>	
Ecrire(Joueur.Nom)	print(Joueur["Nom"])
Ecrire(Joueur.Score)	print(Joueur["Score"])

Application : Revenons à l'enregistrement **Joueur**; faire un programme qui remplit puis affiche l'enregistrement d'un joueur.

En Algorithme	En Python
Procédure remplir(@ joueur:Type_joueur) Début Ecrire("Donner nom joueur:") Lire(joueur.nom) Joueur.score← Aléa(1,6)+ Aléa(1,6)+ Aléa(1,6) Fin Procédure Afiche(joueur:Type_joueur) Début	<pre>from random import randint #--Procédure remplir-- def remplir(): global joueur joueur["nom"]=input("Donner nom joueur:") joueur["score"]=randint(1,6)+randint(1,6)+randint(1,6) #--Procédure affiche-- def affiche(joueur):</pre>

<pre> print("Nom=" , joueur.nom) print("Score joueur=", joueur.score) Fin #----PP----- Début Remplir(joueur) Affiche(joueur) Fin </pre>	<pre> print("Nom=" , joueur["nom"]) print("Score joueur=", joueur["score"]) #--PP— joueur={ "nom":str(), "score":int() } remplir() affiche(joueur) </pre>
---	---

EXERCICE 1 :

Définir un type **TEMPS** qui contient les **champs heure, minute et seconde**

EXERCICE 2 :

Ecrire une fonction **TRANSFORM** qui transforme un temps **T** de type **TEMPS** en un entier **S** qui exprime ce temps en **secondes**.

Exemple : pour **T = 3** heures **15** minutes **57** secondes, **S = 11757** secondes.

EXERCICE 3 :

Ecrire une procédure **DECOMPOS** qui décompose un temps **S** exprimé en secondes en un temps **T** de type **TEMPS**.

Exemple : pour **S = 4968** secondes, **T = 1** heures **22** minutes **48** secondes.

EXERCICE 4 :

Etant donnés deux temps **T1** et **T2** de type **TEMPS**, écrire un algorithme qui calcule le temps **T** somme des temps **T1** et **T2** (**T, T1** et **T2** sont de type **TEMPS**) en utilisant les (la Fonction et la Procédure) **TRANSFORM** et **DECOMPOS**.

Les tableaux d'enregistrements

Application Enregistrement

EXERCICE 1 :

Définir un type **TEMPS** qui contient les **champs heure, minute et seconde**

Tableau de Déclaration des Nouveaux Types (T.D.NT)

Types
Temps = Enregistrement
Heure : Entier
Minute : Entier
Seconde : Entier
Fin

EXERCICE 2 :

Ecrire une fonction **TRANSFORM** qui transforme un temps **T** de type **TEMPS** en un entier **S** qui exprime ce temps en **secondes**.

Exemple : pour **T = 3** heures **15** minutes **57** secondes, **S = 11757** secondes.

Fonction TRANSFORM (T:Temps) : Entier

Début

Retourner T.Heure*3600+T.Minute*60 + T.Seconde

Fin

EXERCICE 3 :

Ecrire une procédure **DECOMPOS** qui décompose un temps **S** exprimé en secondes en un temps **T** de type **TEMPS**.

Exemple : pour **S = 4968** secondes, **T = 1** heures **22** minutes **48** secondes.

Procédure DECOMPOS (S:Entier; @ T : Temps)

Début

 T.Heure $\leftarrow S \text{ Div } 3600$
 T.Minute $\leftarrow (S \text{ Mod } 3600) \text{ Div } 60$
 T.Seconde $\leftarrow S \text{ Mod } 60$

Fin

EXERCICE 4 :

Etant donnés deux temps **T1** et **T2** de type **TEMPS**, écrire un algorithme qui calcule le temps **T** somme des temps **T1** et **T2** (**T, T1 et T2** sont de type **TEMPS**) en utilisant les (la Fonction et la Procédure) **TRANSFORM** et **DECOMPOS**.

(T.D.NT)
Types
Temps = Enregistrement
Heure : Entier
Minute : Entier
Seconde : Entier
Fin

T.D.O	
Objet	Nature/Type
T	Temps
T1	Temps
T2	Temps

Analyse de problème:

- 1.Saisir les deux temps **T1** et **T2**
- 2.Convertir **T1** en secondes (**S_T1**)
- 3.Convertir **T2** en secondes (**S_T2**)
- 4.Décomposer un temps (**S_T1+S_T2**) exprimé en secondes en un temps **T**

Algorithme du PP: CALCUL_TEMPS

Début

 Saisie_Temps(T1)
 Saisie_Temps(T2)
 S_T1 $\leftarrow \text{TRANSFORM}(T1)$
 S_T2 $\leftarrow \text{TRANSFORM}(T2)$
 DECOMPOS(S_T1+S_T2 , T)
 Ecrire(T.Heure , " Heure" , T.Minute , " Min , "T.Seconde , "
 Sec")

Fin

Traduction en Python

TEMPS.py ×

```
1 #--Procédure saisie_temps--
2 def saisie_temps():
3     #   global temps
4     temps["heure"] = int(input("Donner heure:"))
5     temps["minute"] = int(input("Donner minute:"))
6     temps["seconde"] = int(input("Donner seconde:"))
7 #--transformer temps en seconde
8 def transform(t1):
9     return t1["heure"]*3600+t1["minute"]*60+t1["seconde"]
10 #--decomposer s sec en H, Min, Sec
11 def decompos(s):
12     global temps
13     temps["heure"] = s // 3600
14     temps["minute"] = (s % 3600) // 60
15     temps["seconde"] = s % 60
16 #--PP--
17 temps = {
18     "heure":int(),
19     "minute":int(),
20     "seconde":int()
21 }
22 saisie_temps()
23 t1 = {}
24 t1["heure"] = temps["heure"]
25 t1["minute"] = temps["minute"]
26 t1["seconde"] = temps["seconde"]
27 saisie_temps()
28 t2 = {}
29 t2["heure"] = temps["heure"]
30 t2["minute"] = temps["minute"]
31 t2["seconde"] = temps["seconde"]
32 s_t1 = transform(t1)
33 s_t2 = transform(t2)
34 decompos(s_t1 + s_t2)
35 t = {}
36 t["heure"] = temps["heure"]
37 t["minute"] = temps["minute"]
38 t["seconde"] = temps["seconde"]
39 print(t["heure"], " H ", t["minute"], " Min ", t["seconde"], " Sec")
```

Les tableaux d'enregistrements

Activité :

Revenons au jeu de tire à l'arc. On a défini la structure enregistrement joueur formée par les deux champs Nom e Score.

T.D.N.TYPE	
Objet	Nature/Type
Type_Joueur =	Enregistrement
	Nom: Chaine
	Score : entier
	Fin

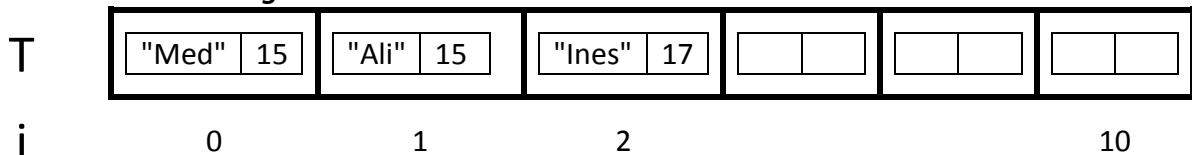
On se propose de faire un programme qui permet de saisir les enregistrements de n ($5 \leq n \leq 10$) joueurs puis les affiche triés en ordre décroissant selon leurs scores.

Question : Peut-on utiliser la structure Tableau pour stocker les enregistrements des joueurs ?

Réponse :

Un tableau ne peut contenir que des éléments de même type y compris le type enregistrement. Nous pouvons donc utiliser un tableau ou un vecteur d'enregistrement.

Exemple d'un tableau d'enregistrements



Algorithme du PP : Jeux

Début

```
Saisie(n,T)           # saisir n et remplir T par les enregistrements de n joueurs
Tri_bulles(T,n)
Affiche(T, n)
```

Fin

T.D.N.TYPE	
Type	
Type_joueur=	Enregistrement
	Nom: Chaine
	Score : entier
	Fin
Tab = Tableau de 10 éléments de Type_joueurs	

T.D.O	
Objet	Nature/Type
T	Tab
n	Entiers
Saisie	Procédure
Tri_bulles	Procédure
Affiche	Procédure

Algorithme de la procédure Saisie

Procédure Saisie(@ n: Entier ; @ T:Tab)

Début

```
Répéter
  Ecrire ("Donner n entre 5 et 10:")
  Lire(n)
  Jusqu'à(5≤n≤10)
  Pour i de 0 à (n-1) Faire
    Ecrire("Donner nom:")
    Lire(T[i].Nom)
    T[i].Score ← Aléa(1,6)+Aléa(1,6)+Aléa(1,6)
  FinPour
Fin
```

Algorithme de la procédure Affiche

Procédure Affiche(T:Tab; n:Entier)

Début

```
Pour i de 0 à (n-1) Faire
  Ecrir(T[i].Nom, " ", T[i].Score)
FinPour
Fin
```

Fin

Application : Tableau d'enregistrements

Un gérant d'une salle de sport veut récompenser les adhérents fidèles en leur offrant un bonus sous forme d'heures d'entraînement gratuites.

Le bonus est calculé en fonction de l'ancienneté de l'adhérent, exprimée en **nombre de mois** par rapport à la date du jour.

En effet, une heure supplémentaire est offerte pour chaque mois d'ancienneté, sachant que le bonus ne sera pris en considération que si l'ancienneté **dépasse 5 ans**.

Les informations relatives à un adhérent sont :

- Un numéro d'abonnement est formé de 5 chiffres.
- Une catégorie de l'adhérent (A : Adulte, J : Junior, E : Enfant)
- L'année d'adhésion qui doit être comprise entre 2000 et 2021.
- Le mois d'adhésion dont la valeur doit être compris entre 1 et 12

Exemple : Pour un adhérent dont l'année d'adhésion est **2015**, le mois d'adhésion est **1** (janvier). Le bonus accordé à cet adhérent est de **74 heures**.

En effet, son ancienneté est égale à **six ans et deux mois** par rapport à la date d'aujourd'hui (**10/03/2021**), en nombre de mois elle est égale à **74 (12*6 + 2)**.

On se propose d'écrire un script python permet de :

1. Saisir un entier N représentant le nombre d'adhérents de cette salle (Avec $2 \leq N \leq 30$)
2. Remplir un tableau T par les N adhérents de ces adhérents.
3. Afficher les numéros d'abonnements des adhérents ayant une ancienneté supérieure à 5 ans et appartenant à une catégorie donnée ainsi que leurs bonus correspondants sachant que la date du jour à considérer est 10/03/2021.

Explication :

1. La lecture de nombre d'adhérent (n) est réalisée par la **procédure Saisie_n()**
2. Les informations relatives à un adhérent sont représentées **par un enregistrement**
3. Le remplissage du tableau T par les enregistrements de n adhérents est réalisé par la **procédure Remplir()**:
 - a. **Les adhérents à afficher:**
 - i. **De catégorie donnée**
 - ii. **Ont une ancienneté > 5 ans le 10/03/2021**
 - iii. **On affiche num adhérent et bonus en heures**
 4. Pour afficher les adhérents ainsi que leurs bonus est la tâche de la procédure **Affiche_ad_bonus()**.

Algorithme de la procédure Saisie_n

Procédure Saisie_n(@ n : Entier)

Début

Répéter

Ecrire("Donner n entre 2 et 30:")

Lire(n)

Jusqu'à (n ∈ [2..30])

Fin

Algorithme de la procédure Remplir

Procédure Remplir(@T:Tab ; n:Entier)

Début

Pour i de 0 à (n-1) Faire

Répéter

Ecrire("Donner num adhérent de la forme 0000:")

Lire(ad.num)

T.D.O.Locaux	
Objet	N/T
i	Entier
ad	adhérent

Jusqu'à (estnum(ad.num)=Vrai) ET (long(ad.num)=4)

Répéter

- Ecrire("Donner catégorie A:Adulte, J:Junior ou E:Enfant:")
- Lire(ad.cat)
- Jusqu'à (ad.cat ∈ {"A", "J", "E"})
- Répéter

 - Ecrire("Donner année adhésion 2000..2021:")
 - Lire(ad.an)
 - Jusqu'à (ad.an ∈ [2000..2021])
 - Répéter

 - Ecrire("Donner mois adhésion:")
 - Lire(ad.mois)
 - Jusqu'à (ad.mois ∈ [1..12])
 - T[i]=ad

FinPour

Fin

Algorithme de la procédure Affiche_ad_bonus

Procédure Affiche_ad_bonus(T:Tab;n:Entier,C:caractère):

Début

Pour i de 0 à (n-1) Faire

Si (T[i].cat=C) ET ((2021-T[i].an)>5) Alors

Bonus= (2021-T[i].an)*12+(3-T[i].mois)

Ecrire(T[i].num, " a gagné ", Bonus, " Heures")

Fin

FinPour

Fin

Algorithme du PP SALLE_SPORT

Début

Saisie_n(n)

Remplir(T,n)

Répéter

Ecrire("Donner catégorie A:Adulte, J:Junior ou E:Enfant:")

Lire(categorie)

Jusqu'à (categorie ∈ {"A", "J", "E"})

Affiche_ad_cat(T,n,categorie)

Fin

T.D.O.Locaux	
Objet	N/T
i	Entier
Bonus	Entier

T.D.N.T	
Types	
<u>Adherent= Enregistrement</u>	
Num: Chaîne	
Cat:caractère	
An: Entier	
Mois : Entier	
<u>Fin</u>	
<u>Tab=Tableau de 30 Adherent</u>	

T.D.O.Globaux	
Objet	N/T
T	Tab
n	Entier
categorie	caractère
Saisie_n	Procédure
Remplir	Procédure
Affiche_ad_bonus	Procédure

```

1 from numpy import array
2 def saisie_n():
3     global n
4     n=int(input("donner n entre 2et30:"))
5     while not(n in range(2,31)):
6         n=int(input("donner n entre 2et30:"))
7 def remplir(n):
8     global t
9     for i in range(n):
10        ad={}
11        ad["num"]=""
12        ad["cat"]="z"
13        ad["an"]=0
14        ad["mois"]=0
15        while (ad["num"].isnumeric()==False)or(len(ad["num"])!=5):
16            ad["num"]=input("donner num de forme 00000:")
17        while not (ad["cat"] in {"A","J","E"}):
18            ad["cat"]=input("saisir catégorie A,J ou E:")
19        while not (ad["an"] in range (2000,2022)):
20            ad["an"] = int(input("danner an entre 2000et2022:"))
21        while not(ad["mois"] in range(1,13)):
22            ad["mois"]=int(input("donner mois :"))
23        t[i]=ad
24 def affiche_ad_bonus(t,n,categorie):
25     for i in range(n):
26         if (t[i]["cat"]==categorie) and (2021-t[i]["an"]>=5):
27             bonus = (2021-t[i]["an"])*12 + (3-t[i]["mois"])
28             print(t[i]["num"], " a gagné ", bonus , " Heures")
29 #---PP---
30 ad={
31     "num":str(),
32     "cat" :str(),
33     "an" :int(),
34     "mois":int()
35 }
36 saisie_n()
37 t=array([ad]*n)
38 remplir(n)
39 categorie="x"
40 while not (categorie in {"A","J","E"}):
41     categorie=input("saisir catégorie A,J ou E:")
42 affiche_ad_bonus(t,n,categorie)

```