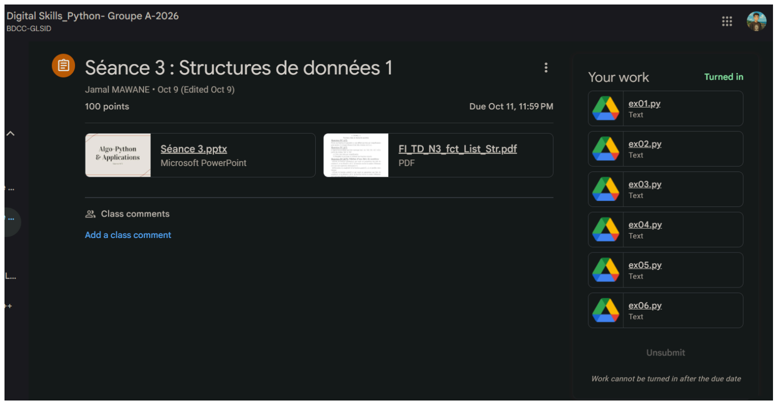
TD5

Bonjour Professeur,
Je me suis rendu compte que j’ai déposé mes devoirs dans la mauvaise classroom ces trois dernières semaines.
Je joins une capture d’écran et les fichiers TD3 et TD4 pour vous montrer que je les avais bien faits à temps.
Merci pour votre compréhension.



Digital Skills_Python- Groupe A-2026
BDCC-GLSID

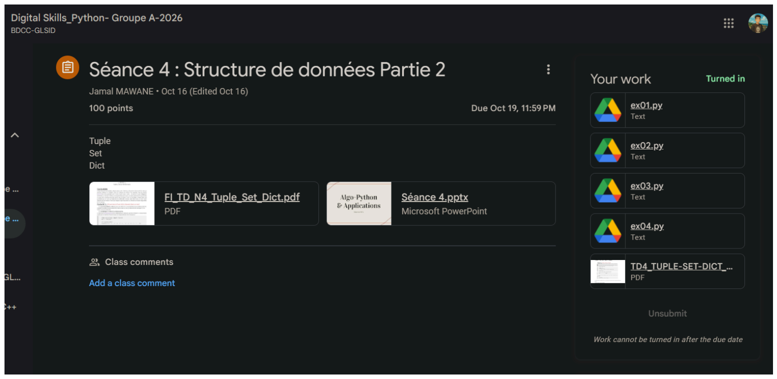
Séance 3 : Structures de données 1
Jamal MAWANE • Oct 9 (Edited Oct 9)
100 points
Due Oct 11, 11:59 PM

Séance 3.pptx
Microsoft PowerPoint

FI_TD_N3_fct_List_Str.pdf
PDF

Class comments
Add a class comment

Your work — Turned in
ex01.py Text
ex02.py Text
ex03.py Text
ex04.py Text
ex05.py Text
ex06.py Text
Unsubmit
Work cannot be turned in after the due date



Digital Skills_Python- Groupe A-2026
BDCC-GLSID

Séance 4 : Structure de données Partie 2
Jamal MAWANE • Oct 16 (Edited Oct 16)
100 points
Due Oct 19, 11:59 PM

Tuple
Set
Dict

FI_TD_N4_Tuple_Set_Dict.pdf
PDF

Séance 4.pptx
Microsoft PowerPoint

Class comments
Add a class comment

Your work — Turned in
ex01.py Text
ex02.py Text
ex03.py Text
ex04.py Text
TD4_TUPLE-SET-DICT_... PDF
Unsubmit
Work cannot be turned in after the due date

ex01

```
def fct(n):
    if( n <= 1):
        return 1
    return n * fact(n-1)

def Empiler_Fat(n, P1):
    for i in range(n+1):
        P1.append(fact(i))
    return P1



def Empiler_Puiss(n,x,P2):
    for i in range(n+1):
        P2.append(x**i)
    return P2

def Empiler_Frac(P1 , P2):
    P3 = []
    for i in range(len(P1)):
        P3.append(P2[i]/P1[i])
    return P3

def Somme(n ,x):
    sum = 0
    p1=[]
    p1=Empiler_Fat(n, p1)
    p2=[]
    p2=Empiler_Puiss(n,x,p2)
    P3 = Empiler_Frac(p1 , p2)
    for elemnt in P3:
        sum += elemnt
    return sum

Somme_result=Somme(2 , 2)
print("La somme de la série est :", Somme_result)
```

ex02

q1

```
17 10 -
⇒ 17 - 10 = 7
========================================
3 28 7 / +
⇒ 28/7=4 ⇒ 3 + 4 = 7
========================================
3 28 + 7 /
⇒ 3+28=31 / 7 = 4.42
```

q2

```
(19 ∗ 6) – 7
⇒ 19 6 ∗ 7 -
=======================================
(4 - 2 * (7 + 6)) + 3 ∗ 5
⇒ 7 6 + 2 ∗ 4 - 3 5 * +
```

q3 q4 q5 q6 q7

```



#q3
def PileVide():
    return []

def EstVide(P):
    if len(P) == 0:
        return True
    return False
```

```
def Empiler(P,n):
    P.append(n)
    return P

def Depiler(P):
    if EstVide(P):
        print("erreur: pile est vide")
        return
    return P.pop()

def SommetPile(P):
    return P[-1]

#q4
def EstChiffre(c):
    if c >= '0' or c <= '9' :
        return 1
    return 0

#q5
def Convertir(c):
    return int(c)

#q6
def Evaluer(expression):
    op = ['+','-','*','/']
    nums = expression.split()
    P=[]
    for element in nums:
        if(element not in op):
            Empiler(P,element)
        else:
            if len(P) < 2:
                print(P)
                print("erreur")
                return
            else:
                p2 = Convertir(Depiler(P))
                p1 = Convertir(Depiler(P))
                if element == "+":
                    P = Empiler(P,p1 + p2)
                if element == "-":
                    P = Empiler(P,p1 - p2)
                if element == "*":
                    P = Empiler(P,p1 * p2)
                if element == "/":
                    P = Empiler(P,p1 / p2)

    return SommetPile(P)

print(Evaluer("3 28 + 7 /"))

#q7
def EvaluerTexte(Fsrc, Fdest):
    ff = open(Fsrc, "r", encoding="utf-8")
    contenu = ff.read()
    ff.close()
    resultat = Evaluer(contenu)
    f = open(Fdest, "w", encoding="utf-8")
    f.write(str(resultat))
    f.close()

EvaluerTexte("./src.txt","./dest.txt")
```

ex3

```
"""
file = {
    0: [élément1, élément2],  # least priority
    1: [élément3],
    2: [élément4, élément5]   # highest priority
}
"""

def deposer_dans_queue(file, element,priorite):
    if priorite not in file:
        file[priorite] = []
    file[priorite].append(element)

def retirer_de_queue(queue):
    maxPriorite = max(queue.keys())
    element = queue[maxPriorite].pop(0)
    if not queue[maxPriorite]:
        del queue[maxPriorite]
    return element
```

ex4

```



#q a
"""
client = {"numero","date_arrive","duree_traitement","date_fin"}
"""
#b
import random

INTERVALLE_MAX = 300
DUREE_TRAITEMENT_MAX = 600

def CreerListeClients():
    n = int(input("nombre de clients :"))
    clients = []
    date_arrivee = 8 * 3600  # 8h in seconds
    date_fin_precedent = date_arrivee

    for i in range(1, n+1):
        intervalle = random.randint(0, INTERVALLE_MAX)
        duree = random.randint(0, DUREE_TRAITEMENT_MAX)

        if i == 1:
            arrivee = date_arrivee
        else:
            arrivee += intervalle

        debut_traitement = max(arrivee, date_fin_precedent)
        date_fin = debut_traitement + duree

        client = {
            "numero": i,
            "date_arrivee": arrivee,
            "intervalle": intervalle,
            "duree_traitement": duree,
            "date_fin": date_fin
        }
        clients.append(client)
        date_fin_precedent = date_fin

    return clients
#c
def afficherClients(clients):
    for c in clients:
        h1 = c["date_arrivee"] // 3600
        m1 = (c["date_arrivee"] % 3600) // 60
        s1 = c["date_arrivee"] % 60

        h2 = c["date_fin"] // 3600
        m2 = (c["date_fin"] % 3600) // 60
        s2 = c["date_fin"] % 60
        print(f"Client {c['numero']}: arrivee {h1}h {m1}min {s1}s, fin {h2}h {m2}mi
n {s2}s")

clients = CreerListeClients()
afficherClients(clients)
```

ex5

```
#Importer
#q1
import os

def copyfile():
    source = os.path.expanduser(r"C:\Users\moham\Downloads\utilisateurs.cs
v")
    destination = os.path.join(os.getcwd(), "utilisateurs.csv")

    os.rename(source, destination)
    print("File moved successfully!")
    return

#q2
import csv

def readingcsv():
    f = open("utilisateurs.csv")
    lecteur = csv.DictReader(f,delimiter=";")
    f.close
    arr= []
    for ligne in lecteur:
        arr.append(ligne)
    return arr
#q3
arrayofdicts = readingcsv()
```

```
############################################################
############
#Sélectionner
#1
def joueurs300score(arrayofdicts):
    joueurs = []
    for ligne in arrayofdicts:
        raw = (list(ligne.values())[0])
        nom, genre, score1, score2, email = raw.split(",")
        if int(score1) >= 300:
            joueurs.append({nom, genre, score1, score2, email})
    return joueurs
#2
def joueursfills(arrayofdicts):
    joueurs = []
    for ligne in arrayofdicts:
        raw = list((ligne.values()))[0]
        nom, genre, score1, score2, email = raw.split(",")
        if genre == "F":
            joueurs.append({nom, genre, score1, score2, email})
    return joueurs
#3
def cherche_domaine(arrayofdicts, domaine):
    joueurs = []
    for ligne in arrayofdicts:
        raw = list((ligne.values()))[0]
        nom, genre, score1, score2, email = raw.split(",")
        if domaine in email:
            joueurs.append({nom, genre, score1, score2, email})
    return joueurs

#projection
#q1
parsed = []
for d in arrayofdicts:
    raw = list(d.values())[0]
    nom, genre, score1, score2, email = raw.split(",")
    parsed.append({
        "nom": nom,
        "genre": genre,
        "score_1": int(score1),
        "score_2": int(score2),
        "email": email
    })

"""
#a
sorted_by_score1 = sorted(parsed, key=lambda x: x["score_1"])
print(sorted_by_score1)

#b
sorted_by_score2 = sorted(parsed, key=lambda x: x["score_2"])
print(sorted_by_score2)
"""
#c
def moy1scores(arrayofplayers):
    sum1 = 0
    for ligne in arrayofplayers:
        sum1 += ligne["score_1"]
    return sum1/len(arrayofplayers)
def mo2scores(arrayofplayers):
    sum2 = 0
    for ligne in arrayofplayers:
        sum2 += ligne["score_2"]
    return sum2/len(arrayofplayers)

#q2
#a
def emails_players(arrayofplayers):
    emails = []
    for ligne in arrayofplayers:
        emails.append({"email": ligne["email"] })
    return emails
#b
def emails_mavJoueurs(arrayofplayers):
    sorted_by_score1 = sorted(arrayofplayers, key=lambda x: x["score_1"])
    i = 0
    arr= []
    while(i < 10):
        arr.append({"email":sorted_by_score1[i]["email"]})
        i+=1
    print(arr)

def emails_mavJoueurs2(arrayofplayers):
    sorted_by_score2 = sorted(arrayofplayers, key=lambda x: x["score_2"])
    i = 0
    arr= []
    while(i < 10):
        arr.append({"email":sorted_by_score2[i]["email"]})
        i+=1
    print(arr)

#emails_mavJoueurs(parsed)
#c
def effacer_doublent(arrayofplayers):
    array = []
    for ligne in arrayofplayers:
        if ligne["email"] not in array:
            array.append(ligne["email"])
    print(array)
    return array
effacer_doublent(parsed)
```