Fichiers : https://drive.google.com/drive/folders/1-YwHjhF8jouimtMaian6mIq93CuC9LoP?usp=share\_link

PARTIE 1 :

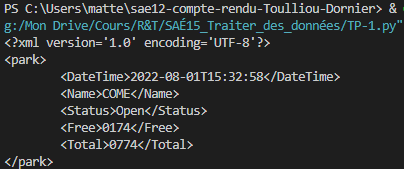


Import request : import le module requests précédement installé avec la commande "pip install requests" dans le cmd.

response=requests.get””[https://data.montpelliers3m.fr/…](https://data.montpelliers3m.fr/%E2%80%A6)”) : récupère dans la variable “response” le contenu de la page xml du lien mentionné.

print(response.text) : imprime sous forme de texte le contenu de la variable response.

Le programme retourne,



Il retourne le contenu de la page xml du lien mentionné dans le programme.

j’écris un programme permettant de print toutes les données de toute la liste ci-dessous.

| import requests  parkings=['FR\_MTP\_ANTI','FR\_MTP\_COME','FR\_MTP\_CORU','FR\_MTP\_EURO','FR\_MTP\_FOCH','FR\_MTP\_GAMB','FR\_MTP\_GARE','FR\_MTP\_TRIA','FR\_MTP\_ARCT','FR\_MTP\_PITO','FR\_MTP\_CIRC','FR\_MTP\_SABI','FR\_MTP\_GARC','FR\_CAS\_SABL','FR\_MTP\_MOSS','FR\_STJ\_SJLC','FR\_MTP\_MEDC','FR\_MTP\_OCCI','FR\_CAS\_VICA','FR\_MTP\_GA109','FR\_MTP\_GA250','FR\_CAS\_CDGA','FR\_MTP\_ARCE','FR\_MTP\_POLY']  def donnee\_parking(parking) :  '''  Fonction permettant de recuperer les données xml d'un parking renseigné.  '''  url\_param= "https://data.montpellier3m.fr/sites/default/files/ressources/"+parking+".xml"  response=requests.get(url\_param)  return response.text  for i in parkings :  print(donnee\_parking(i)) |
| --- |

PARTIE 2 :



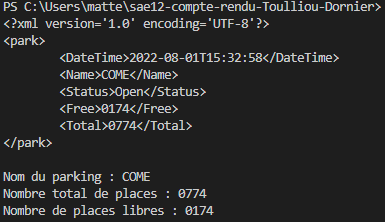


Ce programme retourne et imprime les données xml d’un parking récupéré à partir d’un lien.

Il crée ensuite un fichier texte du nom du parking et y écrit les données xml récupérées.

Il récupère et print les données importantes (en enlevant les balises) du fichier xml en les mettant un peu en forme.

Ce programme retourne ceci :





Ce programme récupère le nombre de places libres dans chaque parking et les sauvegarde dans un fichier “place-libres.txt” (une donnée par ligne contenant uniquement le nombre de places libres.je vérifie, pour chaque parking, s’il est ouvert.

| import requests  from lxml import etree  parkings=['FR\_MTP\_ANTI','FR\_MTP\_COME','FR\_MTP\_CORU','FR\_MTP\_EURO','FR\_MTP\_FOCH','FR\_MTP\_GAMB','FR\_MTP\_GARE','FR\_MTP\_TRIA','FR\_MTP\_ARCT','FR\_MTP\_PITO','FR\_MTP\_CIRC','FR\_MTP\_SABI','FR\_MTP\_GARC','FR\_MTP\_MOSS','FR\_STJ\_SJLC','FR\_MTP\_MEDC','FR\_MTP\_OCCI','FR\_CAS\_VICA','FR\_MTP\_GA109','FR\_MTP\_GA250','FR\_CAS\_CDGA','FR\_MTP\_ARCE','FR\_MTP\_POLY']  def donnee\_parking(parking) :  '''  Fonction permettant de recuperer les données xml d'un parking renseigné.  '''  url\_param= "https://data.montpellier3m.fr/sites/default/files/ressources/"+parking+".xml"  response=requests.get(url\_param)  return response.text  def fichier\_parking(parking) :  '''  Fonction permettant de créer un fichier contenant les données xml du parking mentionné  '''  nom\_fichier= parking+".txt"  f1=open(nom\_fichier,"w",encoding='utf8')  f1.write(donnee\_parking(parking))  f1.close()  def ouvert(parking) :  '''  Fonction permettant de retourner True si le parking est ouvert et False si ce n'est pas le cas.  '''  nom\_fichier=parking+".txt"  tree = etree.parse(nom\_fichier)  for user in tree.xpath("Status") :  if user.text == "Open" :  return True  else :  return False  def places\_libre(parking) :  '''  Fonction retournant le nombre de placves libre d'un parking  '''  nom\_fichier=parking+".txt"  tree = etree.parse(nom\_fichier)  for user in tree.xpath("Free") :  return user.text  f1 =open("places-libres.txt","w",encoding='utf8')  for i in parkings :  fichier\_parking(i)  if ouvert(i) == True :  f1.write(places\_libre(i) + "\n")  f1.close  print(donnee\_parking("FR\_MTP\_ANTI")) |
| --- |



Je modifie mon programme afin de rajouter à chaque ligne le nom du parking.

| f1 =open("places-libres.txt","w",encoding='utf8')  for i in parkings :  fichier\_parking(i)  if ouvert(i) == True :  f1.write(i + " | " + places\_libre(i) + "\n")  f1.close |
| --- |

Surligné en vert : code modifié.



j’écris un programme qui donne le pourcentage de places libre sur chaque parking ainsi qu'un pourcentage de tous les parkings :

| total\_libre = 0  total = 0  for i in parkings :  print (i," | ",calcul.pourcentage(int(places\_libre(i)),int(places\_total(i))))  total\_libre+=int(places\_libre(i))  total+=int(places\_total(i))  print("Total |",calcul.pourcentage(total\_libre,total)) |
| --- |

PARTIE 3 :



Le programme renvoie un entier qui augmente de 1 toute les secondes.

time.time() utilise une commande unix “gettimeofday()” le nombre retourné est le nombre de secondes depuis le début de la journée.



J’écris un programme permettant de récupérer l’occupation du parking « FR\_MTP\_GARE » toutes les 10 secondes pendant 5 minutes et qui sauvegarde ces données dans un fichier.

| start=int(time.time())  f1=open("places\_FR\_MTP\_GARE.txt","w",encoding="utf8")  while int(time.time()) < start+300 :  start2 = round(time.time(),5)  print("1",int(time.time()))  while round(time.time(),5) <= start2+10 :  if round(time.time(),5) == start2+10 :  f1.write(str(int(time.time()))+" | "+places\_libre("FR\_MTP\_GARE")+" | "+places\_total("FR\_MTP\_GARE")+"\n")  print("2",int(time.time()))  f1.close() |
| --- |



J’écris un programme qui permet le suivi de l’occupation de tous les parkings de Montpellier en permettant à l’utilisateur de choisir la période d’échantillonnage (Te) et la durée de l’acquisition (Duree). Il pourra également indiquer le nom du fichier (Nom) dans lequel seront enregistrées les données.

| Te=int(input("Perdiode d'échantillonage (seconde) : \n"))  Duree=int(input("Durée d'acquisition (seconde) : \n"))  Nom=input("Nom du fichiers (sans extentions) : \n")+".txt"  start=int(time.time())  f1=open(Nom,"w",encoding="utf8")  while int(time.time()) < start+Duree :  start2 = round(time.time(),4)  print("1",int(time.time()))  while round(time.time(),4) <= start2+Te :  if round(time.time(),4) == start2+Te :  print("2",int(time.time()))  for i in parkings :  f1.write(str(int(time.time()))+" | "+i+" | "+places\_libre(i)+" | "+places\_total(i)+" | "+str(calcul.pourcentage(int(places\_libre(i)),int(places\_total(i))))+"\n")  print("3",int(time.time()))  f1.write("=========================================================\n")  f1.close() |
| --- |

Pour créer un module, je crée un fichier “TP-1-module.py” je coupe et colle toutes mes fonctions de mon fichier principal. En haut de ce fichier principal j’y renseigne “from TP-1-module import \*”. Je n’ai rien d’autre à modifier. Tout marche niquel.

Je fini par commenter mes fonctions.