En aquesta unitat aprendrem algunes funcions i llibreries de Python que ens permeten fer el tractament d'alguns formats textuals. També aprendrem a tractar tots els arxius de un directori i de tots els seus subdirectoris de manera recursiva. Dedicarem una secció a la descàrrega de contingut de la web i la seva conversió a text.

D'aquesta unitat teniu els següents arxius disponibles:

- Aquest mateix capítol en pdf:
- Els programes i arxius necessaris:

9.1. Codi de caràcters: detecció, tractament i conversió

Per a la detecció de la codificació de caracters en Python podem fer servir la llibreria de Python chardet. Si no la teniu instal·lada, ho podeu fer utilitzant pip:

```
pip install chardet
```

(recorda que en Linux és possible que hagis d'executar l'ordre amb permissos d'administrado i que probablement tinguis instal·lat tant Python 2 com Python 3. En aquests casos has d'escriure:

```
sudo pip3 install chardet
```

Ara podem escriure un programa (programa-9-1.py) per detectar la codificació d'un arxiu de text de la següent manera:

```
import sys
import chardet
fitxer_entrada=sys.argv[1]
raw_data=open(fitxer_entrada,"rb").read()
codificacio=chardet.detect(raw_data)
print("Fitxer:",fitxer_entrada,"Codificació:",codificacio)
print(codificacio["encoding"])
```

Com a paràmetre d'entrada li donarem l'arxiu del que volem conèixer la codificació (per exemple l'arxiu noticia2-jap.txt)

```
python3 programa-9-1.py  noticia2-jap.txt
```

Que ens retornarà per pantalla:

```
Fitxer: doc1.txt Codificació: {'encoding': 'ISO-2022-JP', 'confidence': 0.99, 'language': 'Japanese'}
ISO-2022-JP
```

Fixeu-vos que en la variable codificació el paquet chardet ens dona una diccionari amb tres claus: encoding, confidence i language. En el primer print escrivim directament tot el diccionari i en canvi en el segon print escrivim només la codificació. Proveu el programa amb la resta de llengües i versions (noticia-cat.txt, noticia2-cat.txt, noticia-rus.txt, noticia2-rus.txt i noticia-jap.txt). Observareu que el programa no sempre és capaç de detectar la llengua.

Hi ha altres opcions vàlides per detectar la codificació de caràcters d'un arxiu de text. En el programa-9-2.py fem servir la llibreria magic. Recorda que pots instal·lar-la amb pip p pip3:

```
pip install python-magic
```

El programa 9.2;

```
import sys
import magic

fitxer_entrada=sys.argv[1]
m = magic.from_file(fitxer_entrada)
```

```
print(m)
```

Proveu aquesta llibreria amb els arxius de noticies adjunts a aquesta unitat.
Recordem que un cop que coneixem la codificació d'un arxiu el podrem obrir correctament fent servir la llibreria codecs:

```
import codecs
entrada=codecs.open("noticia-cat","r",encoding="utf-8")
```

Hi ha tres modes per obrir un arxiu, depenent de l'operació que vulguem fer amb ell:

- r: lectura
- w: escriptura
- a: per afegir dades al final de l'arxiu
- r+: tant per llegir com per escriure

Ara, si volem obrir i llegir un arxiu de text, farem:

```
import codecs
entrada=codecs.open("noticia-cat.txt","r",encoding="utf-8")
for linia in entrada:
    linia=linia.rstrip()
    print(linia)
```

Quan fem rstrip() a la línia estem traient el caràcter o caràcters de salt de línia que pugui tenir. Ara podem obrir l'arxiu en la codificació adequada i guardar el seu contingut en un altre arxiu amb una altra codificació (programa-9-3.py):

```
import codecs
entrada=codecs.open("noticia-cat.txt","r",encoding="utf-8")
sortida=codecs.open("noticia3-cat.txt","w",encoding="iso-8859-1")
for linia in entrada:
    sortida.write(linia)
```

Fixeu-vos que ara no cal que fem servir rstrip() ja que quan gravem la línia al fitxer de sortida de fer volem que hi hagi el salt de línia al final de la línia.

En Python també disposem de llibreries específiques per poder detectar la llengua d'un document. Per exemple, langdetect. En el programa-9-4.py podem veure un exemple d'ús d'aquesta llibreria. (recorda instal·lar-la amb pip o pip3)

```
from langdetect import detect

lang = detect("Metges de família que han de visitar nens perquè els serveis de pediatria del CAP estan desbordats i urgències hospitalàries amb el doble d'activitat. Com cada tardor, el virus respiratori sincicial (VRS), causant de la bronquiolitis, ja fa setmanes que circula i ja ha assolit els nivells d'epidèmia, de manera que ha tensionat els serveis de pediatria dels CAP i els hospitals, com passa també cada temporada.")
print("Llengua text 1",lang)

lang = detect("Телеканал ссылается на заявку на исследование в Научно-техническую организацию НАТО, к которой получил доступ.Недавние учения на восточной границе НАТО, присутствие в Прибалтике и война на Украине продемонстрировали успешность российской стратегии радиоэлектронной борьбы (РЭБ). И потенциал радиоэлектронных атак (например, подавления), и радиоэлектронное обеспечение (например, обнаружение целей) показывают, что существующая инфраструктура тактической связи НАТО уязвима перед средствами РЭБ и столкнется с серьезными угрозами, - говорится в документе, передает RT.")
print("Llengua text 2",lang)
```

Si l'executem, a la sortida apareixerà:

```
Llengua text 1 ca
Llengua text 2 ru
```

Exercici 9.1. Crea un programa que converteix un arxiu de text en qualsevol codificació en un arxiu de text en Unicode UTF-8. Podeu descarregar la solució (solucio-9-1.py), però primer intenteu fer-lo, ja que no és massa complicat

Exercici 9.2. Modifica el programa anterior de manera que si l'arxiu ja està en Unicode UTF-8 el programa avisi i no faci res (és a dir, que no creï l'arxiu de sortida). Podeu descarregar la solució (solucio-9-2.py), però primer intenteu fer-lo, ja que no és massa complicat

9.2. Tractament de directoris

Tractament de tots els arxius d'un directori programa-9-5.py):

```
import os
for file in os.listdir("./directori"):
 print(file)
```

També es poden tractar els directoris de forma recursiva, és a dir, anant per tota l'estructura de subdirectoris (descarregueu i descomprimiu l'arxiu directori-recursiu.zip) (programa-9-6.py)

```
import os
for root, dirs, files in os.walk("./directori-recursiu"):
 for file in files:
  print(os.path.join(root, file))
```

Podem fer que només consideri els arxius amb una determinada extensión (per exemple .txt) fent (programa-9-6b.py):

```
import os
for root, dirs, files in os.walk("./directori-recursiu"):
 for file in files:
  if file.endswith(".txt"):
   print(os.path.join(root, file))
```

Exercici: Feu un programa que converteixi tots els arxius amb extensió .txt que no estiguin en codificació Unicode utf-8 a Unicode utf-8. Si algun arxiu ja ho està, no cal que el converteixi. Feu que l'arxiu de sortida tingui l'extensió .utf8. (el arxius necessaris són els mateixos d'abans i els podeu obtenir d'aquí)

9.3. Tractament d'arxius tabulats i CSV

El tractament d'arxius de text amb camps separats per tabuladors es pot fer fàcilment amb la funció estàndard split().

Prenem com a exemple l'arxiu cdlconsulteca.txt que conté entrades terminològiques amb la següent informació:

abdicar v tr abdicar abdicate Dret civil Renunciar al domini d'alguna cosa, una propietat, un dret o una opinió. Consulteca-13

i cada un dels camps està separat per tabulador. Fem un programa que llegeixi aquest arxiu i en guardi un altre que contingui només: denominació en anglès, denominació en català, i l'àrea d'especialitat (fixeu-vos que són el 4t camp, el 1r camps i el 5è camps (recordeu, però, que els índexos d'una llista comencen per 0, no per 1).

Podem trobar la solució al programa programa-9-7.py:

```
import codecs
entrada=codecs.open("cdlconsulteca.txt","r",encoding="utf-8")
sortida=codecs.open("cdlconsulteca-mod.txt","w",encoding="utf-8")

for linia in entrada:
 linia=linia.rstrip()
```

```
camps=linia.split("\t")
cadena=camps[3]+"\t"+camps[1]+"\t"+camps[4]
print(cadena)
sortida.write(cadena+"\n")
```

Exercici: Aquest programa té el problema de que si una entrada no té denominació anglesa igualment escriu la sortida. Modifica el programa per a que només escrigui la sortida si tenim tant la denominació anglesa com la catalana.

Els arxius CSV (Comma Separated Values) es poden tractar d'una manera genèrica fent servir el paquet csv. Aquests arxius es caracteritzen per:

- Un separador: , o un altre
- Un delimitador de text: " o ' o qualsevol altre caràcter

El de text tabulat és un cas especial on el separador és un tabulador i no hi ha delimitador. Si ara tenim l'arxiu de glossari en csv fent servir comes i sense delimitador de text (cdlconsulteca.csv). És a dir, té la següent forma:

```
abdicar,v tr,abdicar,abdicate,Dret civil,"Renunciar al domini d'alguna cosa, una propietat, un dret o una opinió.",Consulteca-13
```

Podem fer el tractament de l'arxiu amb el paquet csv fent (programa-9-8.py):

```
import csv
import codecs
entrada=codecs.open("cdlconsulteca.csv","r",encoding="utf-8")
lector=csv.reader(entrada, delimiter=',', quotechar='"')
for camps in lector:
    print(camps)
```

El paquet csv també ens proporciona funcionalitats per escriure arxius csv.

```
escriptor = csv.writer(csvfile, delimiter=';',quotechar='"', quoting=csv.QUOTE_MINIMAL)
```

- csv.QUOTE_ALL: Instructs writer objects to quote all fields.
- csv.QUOTE_MINIMAL : Instructs writer objects to only quote those fields which contain special characters such as delimiter, quotechar or any of the characters in lineterminator.
- csv.QUOTE_NONNUMERIC: Instructs writer objects to quote all non-numeric fields.
- csv.QUOTE_NONE: Instructs writer objects to never quote fields. When the current delimiter occurs in output data it is preceded by the current escapechar character. If escapechar is not set, the writer will raise Error if any characters that require escaping are encountered.

Podem fer per exemple un programa que converteixi l'arxiu csv del glossari en un altre arxiu on el separador sigui punt i coma (;) i el delimitador de text siguin les cometes ("). Ho podem fer amb el programa-9-9.py:

```
import csv
import codecs
entrada=codecs.open("cdlconsulteca.csv","r",encoding="utf-8")
lector=csv.reader(entrada, delimiter=',', quotechar='"')
sortida=codecs.open("cdlconsulteca2.csv","w",encoding="utf-8")
escriptor = csv.writer(sortida, delimiter=';',quotechar='"', quoting=csv.QUOTE_ALL)
for linia in lector:
    escriptor.writerow(linia)
```

Si executem aquest programa, en el arxiu de sortida (cdlconsulteca2.csv) obtindriem la informació de la següent manera:

```
"abdicar";"v tr";"abdicar";"abdicate";"Dret civil";"Renunciar al domini d'alguna  cosa,  una propietat, un dret o una opinió.";"Consulteca-13"
```

9.4. Obtenció d'informació de la web

Amb Python podem obtenir fàcilment informació de la web. Per exemple, per obtenir una web en format htm (programa-9-10.py) (NOTA: posa una adreça web completa a la tercera línia)

```
   import urllib.request
import codecs
f = urllib.request.urlopen("https://www.ara.cat/esports/barca/resultats-observatori-blaugrana-
febrer-2018_0_1954004729.html")
web=f.read().decode('utf-8')

   sortida=codecs.open("noticia.html","w",encoding="utf-8")
sortida.write(web)
```

En el programa-9-11.py obtenim aquesta informació en html i la convertim a text.

```
   import urllib.request
import codecs
import html2text
f = urllib.request.urlopen("https://www.ara.cat/esports/barca/resultats-observatori-blaugrana-
febrer-2018_0_1954004729.html")
web=f.read().decode('utf-8')
text=html2text.html2text(web)
sortida=codecs.open("noticia.txt","w",encoding="utf-8")
sortida.write(text)
```