

A1. Exercicis bàsics fitxers

1. Escriu un programa que compti quantes paraules hi ha dins d'un fitxer. Després escriu un programa que llegeixi una paraula de teclat i la posi dins d'un fitxer línia per paraula.

1a. Part

```
import tkinter.filedialog
##Podem utilitzar aquest altre metode per tal de obrir el fitxer
##ruta_fitxer = tkinter.filedialog.askopenfilename()
def comptarParaules(ruta_fitxer):

    pintor_file = open(ruta_fitxer, 'r')
    content= pintor_file.read()
    content = content.split()

    return len(content)

print("El vostre document conté",comptarParaules('Fortuny.txt'), \
      "paraules")
```

```
[evaluate A1.Exercicis_basics_fitxers_0.py]
El vostre document conté 181 paraules
>>>
```

Podem fer un altre versió del programa *comptar_paraules*, perquè compti caràcters.

```
def comptar_caracters(ruta_fitxer):
    '''
    '''
    pintor_file = ruta_fitxer
    caracters = pintor_file.read()
    pintor_file.close()

    print("El fitxer té " + str(len(caracters)) + " caracters")

#ppal
comptar_caracters(open('Fortuny.txt', 'r'))
```

```
[evaluate A1.Exercicis_basics_fitxers_0.py]
El fitxer té 988 caracters
>>>
```

2a. Part

```
def afegir_paraula(ruta_fitxer):  
    '''  
    ...  
    '''  
  
    #obrir fitxer en mode append  
    pintor_fitxer= ruta_fitxer  
    paraula= input("Introdueix una frase per incloure al fitxer: ")  
    pintor_fitxer.write(paraula)  
    pintor_fitxer.close()  
  
afegir_paraula(open('Fortuny.txt', 'a'))
```



En aquest exemple podem veure perfectament quina es la metodologia per escriure en un fitxer.

```
[evaluate A1.Exercicis_basics_fitxers_0.py]  
>>> Introdueix una frase per incloure al fitxer: Fortuny fou un pintor molt important
```

Comprovem el contingut del fitxer per veure que s'afegit la nova línia de text.

Fortuny no va ser indiferent a tot el que va veure i viure en els seus diferents viatges. I totes aquestes experiències vitals van traspuar en la seva obra. Fortuny fou un pintor molt important

2. Defineix una estructura de dades per a contenir una sèrie de dates importants a la nostra vida (dia, mes i any) i realitza un programa que guardi dins d'un fitxer les dates que vagi introduït l'usuari.

```
def afegir_dates(ruta_fitxer):  
    '''(str) -> NoneType  
    ...  
    '''  
  
    dates_fitxer= open(ruta_fitxer, 'a')  
    #demanen les dates a l'usuari  
    data= input("Introdueix una data de la teva vida (DD/MM/AAAA): ")  
    #Contruïm un bucle amb el controlem la introducció de dades al fitxer
```



Amb la funció crear dates, volem controlar el número de línies que s'introdueixen al fitxer. Per fer això utilitzarem l'enfocament *readline()*.

```
while data != '':  
    data = input("Introdueix un altre data en el mateix format: ")  
    dates_fitxer.write(data+'\n')  
  
    dates_fitxer.close()  
  
afegir_dates('les_meves_dates.txt')
```

```
[evaluate A1.Exercicis_basics_fitxers_1.py]  
Introdueix una data de la teva vida (DD/MM/AAAA): 23/10/2011  
Introdueix un altre data en el mateix format: 04/06/2008  
Introdueix un altre data en el mateix format:  
>>>
```

15/11/2011
12/03/2001
23/10/2011
04/06/2008

Comprovem el fitxer
les_meves_dates.txt i
veiem que les dates que
hem introduït han estat

3. Realitza un programa que trobi una data concreta del fitxer on s'han guardat les dates de l'exercici 1. A més, a més el programa comptarà el número de registres que conté el fitxer.

```
import tkinter.filedialog  
##ruta_fitxer = tkinter.filedialog.askopenfilename()  
  
#ruta_fitxer = '\\Users\\Roser\\Desktop\\dates.txt'  
  
def compta_registre(ruta_fitxers, data):  
    ...  
    Retorna el número de registres de data que conté el fitxer les_meves_dates  
    ...  
    #readlines ens retorna llista amb el contingut del fitxer  
    #obrir fitxer en mode append  
  
    dates_fitxer = ruta_fitxers  
  
    content_lst= dates_fitxer.readlines()  
    dates_fitxer.close()  
  
    cont= 0
```

Com que en aquest cas tan sols
necessitem tractar el contingut
del fitxer per recorre'l, obrirem
el fitxer en mode de lectura 'r'.

```
for d in content_lst:
    if data == d.strip('\n'):
        cont+= 1

return (cont)
```

Recorrem el fitxer utilitzant l'enfocament *for line in file*. L'ús d'aquest mètode permet processar les línies del fitxer una.

```
#ppal

data = input("Introdueix una data en el format DD/MM/AAA: ")
##Fent-ho així, no funciona, és queixa de que no es poden cridar strings
##print("El fitxer conté %d registres amb la data %s." \
        ##%(compta_registre(open('les_meves_dates.txt', 'r'),data)))

print("El vostre fixer, conté",compta_registre\
      (open('les_meves_dates.txt', 'r'),data), "registres iguals")
```

```
[evaluate A1.Exercicis_basics_fitxers_2.py]
Introdueix una data en el format DD/MM/AAA: 23/10/2011
El vostre fixer, conté 3 registres iguals
>>>
```

```
'15/11/2011
12/03/2001
23/10/2011
04/06/2008
23/10/2011
08/09/2001
23/10/2011'
```

4. Realitza un programa que donada una seqüència en caràcters introduïda per l'usuari escrigui dins d'un fitxer la mateixa seqüència en majúscules.

```
import tkinter.filedialog
#ruta_fitxer = tkinter.filedialog.askopenfilename()
#ruta_fitxer = '\\Users\\Roser\\Desktop\\dates.txt'

def majuscles(ruta_fitxer, cadena):
    """
    Retorna el número de registres de data que conté el fitxer ruta_fitxer
    """
```

```
#readlines ens retorna llista amb el contingut del fitxer
#obrir fitxer en mode append
dades_fitxer = open(ruta_fitxer, 'a')

cad_app= cadena.upper()
dades_fitxer.write(cad_app)
dades_fitxer.close()
```

Fent ús del mètode per a *strings* **upper** podem resoldre aquest problema. Tot seguit al mètode **write** li passem com a paràmetre la variable 'cad_app'.

```
sequencia= input("Introdueix una seqüència de caràcters: ")
majuscles('les_meves_dates.txt', sequencia)
```

```
[evaluate A1.Exercicis_basics_fitxers_3.py]
Introdueix una seqüència de caràcters: Hola Manola
>>>
```

Per donar per finalitzat el procés comprovem el el fitxer que hem modificat.

```
15/11/2011
12/03/2001
23/10/2011
04/06/2008
23/10/2011
08/09/2001
23/10/2011
```

HOLA MANOLA

5. Realitza un programa que obri un fitxer qualsevol de text i faci un altre fitxer a partir d'aquest canviant les majúscules per minúscules i les minúscules per majúscules.

Partim d'un fitxer "input_fitxer" amb el següent contingut.

setze Jutges d'un jutjat mengen
fetge d'un penjat; Si el penjat es
despengés els setze jutges menjarien
fetge d'un Despenjat.

```
import tkinter.filedialog
##ruta_fitxer = tkinter.filedialog.askopenfilename()

def swap_fitxer(ruta_fitxer_in, ruta_fitxer_out):
    '''
    '''

    f_in= ruta_fitxer_in
    content = f_in.read()
    f_in.close()

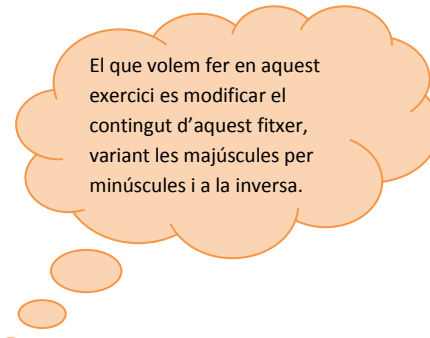
    #tractar contingut

    swap_content = ''
    for c in content:
        if c.isupper():
            swap_content+= c.lower()

        elif c.islower():
            swap_content+= c.upper()
        else:
            swap_content+= c

    #obrir fitxer en mode escriptura
    f_out = ruta_fitxer_out
    f_out.write(swap_content)
    f_out.close()

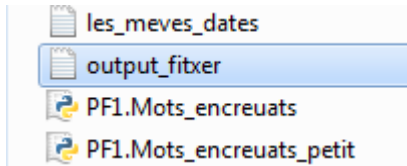
    ruta_fitxer_in= open('input_fitxer.txt', 'r')
    ruta_fitxer_out= open('output_fitxer.txt', 'w')
```



El que volem fer en aquest exercici es modificar el contingut d'aquest fitxer, variant les majúscules per minúscules i a la inversa.

```
[evaluate A1.Exercicis_basics_fitxers_4.py]
>>>
```

La funció no ens retorna res ja que ha estat dissenyada per crear un nou fitxer.



Amb el mètode d' escriptura de fitxers *write*, és crea automàticament el fitxer dins del mateix directori on s'ubica el programa.

El fitxer ha estat modificat i totes les lletres que en el fitxer "input_fitxer" apareixien com a minúscules ara en el fitxer de nova creació apareixen en majúscules.

SETZE JUTGES D'UN JUTJAT MENGEN
FETGE D'UN PENJAT;SI EL PENJAT ES
DESPENGÉS ELS SETZE JUTGES MENJARIEN
FETGE D'UN dESPENJAT.