

Apunts Python - Scripts i Fitxers de Text

Conceptes bàsics

1. Importar mòduls necessaris

```
import sys
```

El mòdul `sys` ve inclòs amb Python, no cal instal·lar res addicional.

Treballar amb arguments de línia de comandes

1. Accedir als arguments

```
import sys
```

```
# sys.argv és una llista amb tots els arguments  
# sys.argv[0] → nom del script  
# sys.argv[1:] → arguments passats per l'usuari
```

Exemple:

```
python script.py -f fitxer.txt -p hola
```

Resultat:

- `sys.argv[0] = 'script.py'`
- `sys.argv[1] = '-f'`
- `sys.argv[2] = 'fitxer.txt'`
- `sys.argv[3] = '-p'`
- `sys.argv[4] = 'hola'`

2. Processar arguments

Mètode bàsic amb bucle:

```
i = 1  
while i < len(sys.argv):  
    if sys.argv[i] in ['-f', '--fitxer']:  
        if i + 1 < len(sys.argv):  
            nom_fitxer = sys.argv[i + 1]  
            i += 2  
        else:  
            print("Error: falta el valor després de -f")  
    else:  
        print(f"Opció desconeguda: {sys.argv[i]}")  
        i += 1
```

3. Detectar opcions d'ajuda

```
if '-h' in sys.argv or '--help' in sys.argv:
    print("Missatge d'ajuda")
    return
```

Lectura i escriptura de fitxers

1. Llegir un fitxer complet

```
with open('fitxer.txt', 'r', encoding='utf-8') as f:
    contingut = f.read() # Tot el contingut com a string
```

2. Llegir línia per línia

```
with open('fitxer.txt', 'r', encoding='utf-8') as f:
    for linia in f:
        print(linia.strip()) # strip() elimina el \n
```

3. Llegir totes les línies com a llista

```
with open('fitxer.txt', 'r', encoding='utf-8') as f:
    linies = f.readlines() # Llista amb cada línia (inclou \n)
```

4. Escriure en un fitxer

Sobreescriure (mode 'w'):

```
with open('fitxer.txt', 'w', encoding='utf-8') as f:
    f.write('Primera línia\n')
    f.write('Segona línia\n')
```

Afegir al final (mode 'a'):

```
with open('fitxer.txt', 'a', encoding='utf-8') as f:
    f.write('Nova línia\n')
```

5. Escriure múltiples línies

```
linies = ['Línia 1\n', 'Línia 2\n', 'Línia 3\n']
with open('fitxer.txt', 'w', encoding='utf-8') as f:
    f.writelines(linies)
```

Operacions amb fitxers

1. Buscar text dins d'un fitxer

Comptar ocurrencies d'una paraula:

```
with open('fitxer.txt', 'r', encoding='utf-8') as f:
    contingut = f.read()
    comptador = contingut.lower().count('paraula'.lower())
```

Buscar en cada línia:

```
with open('fitxer.txt', 'r', encoding='utf-8') as f:
    for num_linia, linia in enumerate(f, start=1):
        if 'paraula' in linia.lower():
            print(f'Trobat a la línia {num_linia}')
```

2. Inserir text en una posició específica

```
# Llegir el fitxer
with open('fitxer.txt', 'r', encoding='utf-8') as f:
    linies = f.readlines()

# Inserir nova línia a la posició 0 (al principi)
linies.insert(0, 'Nova primera línia\n')

# Guardar canvis
with open('fitxer.txt', 'w', encoding='utf-8') as f:
    f.writelines(linies)
```

Important sobre insert():

- insert(0, text) → insereix al principi
- insert(2, text) → insereix ABANS de la posició 2
- Si la posició és > len(llista), afegeix al final

3. Copiar fragment d'un fitxer

```
# Llegir fitxer origen
with open('origen.txt', 'r', encoding='utf-8') as f:
    linies = f.readlines()

# Extreure fragment (línies 2 a 4, comptant des d'1)
linia_inici = 2
linia_final = 4
fragment = linies[linia_inici - 1:linia_final] # Convertir a 0-indexed

# Escriure al fitxer destí
with open('desti.txt', 'w', encoding='utf-8') as f:
    f.writelines(fragment)
```

Gestió d'errors

1. Capturar errors de fitxers

```
try:
    with open('fitxer.txt', 'r', encoding='utf-8') as f:
        contingut = f.read()
except FileNotFoundError:
    print("Error: El fitxer no existeix")
except PermissionError:
    print("Error: No tens permís per accedir al fitxer")
except Exception as e:
    print(f"Error inesperat: {e}")
```

2. Validar arguments

```
# Comprovar que hi ha suficients arguments
if len(sys.argv) < 3:
    print("Error: Falten arguments")
    return
```

```
# Validar que un valor és un número
try:
    posicio = int(sys.argv[2])
except ValueError:
    print("Error: La posició ha de ser un número")
```

3. Validar límits de línia

```
with open('fitxer.txt', 'r', encoding='utf-8') as f:
    linies = f.readlines()

if linia_final > len(linies):
    print(f"Error: El fitxer només té {len(linies)} línies")
```

Bones pràctiques

1. Usar context manager (with)

Correcte:

```
with open('fitxer.txt', 'r', encoding='utf-8') as f:
    contingut = f.read()
# El fitxer es tanca automàticament
```

Incorrecte:

```
f = open('fitxer.txt', 'r')
contingut = f.read()
f.close() # Pot oblidar-se o no executar-se si hi ha error
```

2. Especificar sempre l'encoding

```
# Amb encoding
with open('fitxer.txt', 'r', encoding='utf-8') as f:
    contingut = f.read()

# Sense encoding (pot donar problemes amb accents)
with open('fitxer.txt', 'r') as f:
    contingut = f.read()
```

3. Documentar funcions amb docstrings

```
def buscar_paraula(nom_fitxer, paraula):
    """
    Busca una paraula en un fitxer i compta les ocurrencies

    Args:
        nom_fitxer (str): Nom del fitxer on buscar
        paraula (str): Paraula a buscar

    Returns:
        int: Nombre d'ocurrencies trobades
    """
    # Codi de la funció...
```

4. Separar responsabilitats en funcions

```
def mostrar_ajuda():
    """Mostra el missatge d'ajuda"""
    pass

def processar_arguments(args):
    """Processa i valida els arguments"""
    pass

def executar_accio(param1, param2):
    """Executa l'acció principal"""
    pass

def main():
    """Coordina l'execució del programa"""
```

```

if '-h' in sys.argv:
    print(mostrar_ajuda())
    return

params = processar_arguments(sys.argv)
resultat = executar_accio(params[0], params[1])
print(resultat)

if __name__ == "__main__":
    main()

```

5. Retornar resultats per fer print des de main

```

def processar_fitxer(nom_fitxer):
    """Processa el fitxer i retorna el resultat"""
    try:
        with open(nom_fitxer, 'r', encoding='utf-8') as f:
            linies = len(f.readlines())
            return f"El fitxer té {linies} línies"
    except FileNotFoundError:
        return f"Error: El fitxer '{nom_fitxer}' no existeix"

def main():
    resultat = processar_fitxer('text.txt')
    print(resultat)  # Print des de main

```

Consells útils

Index 0 vs Index 1

Python usa índexs començant per 0:

```

llista = ['a', 'b', 'c']
llista[0]  # 'a' (primer element)
llista[1]  # 'b' (segon element)

```

Però els usuaris solen comptar des d'1:

```

python script.py --liniai 1 --liniaf 3
# Usuari vol: línia 1, 2 i 3
# En Python: índexs 0, 1 i 2

```

Conversió:

```

linia_usuari = 1  # L'usuari diu "línia 1"
index_python = linia_usuari - 1  # Convertim a 0

```

Slicing de llistes

```
linies = ['L1', 'L2', 'L3', 'L4', 'L5']

# De l'índex 1 fins al 3 (3 no inclòs)
linies[1:3] # ['L2', 'L3']

# De l'índex 1 fins al 4 (4 no inclòs, línia 4 inclosa)
linies[1:4] # ['L2', 'L3', 'L4']

# Copiar tot
linies[:] # Còpia completa
```

Gestionar línies amb o sense \n

```
# Les línies llegides amb readlines() inclouen \n
linies = f.readlines() # ['Línia 1\n', 'Línia 2\n']

# Afegir \n si no el té
if not text.endswith('\n'):
    text += '\n'

# Eliminar \n
linia_neta = linia.rstrip('\n')
# o
linia_neta = linia.strip() # Elimina espais i \n dels extrems
```

Exemple d'estructura d'un script complet

```
import sys

def mostrar_ajuda():
    return """
Ús: python script.py [opcions]

Opcions:
    -f, --fitxer      Nom del fitxer
    -p, --param       Paràmetre
    -h, --help        Mostra aquesta ajuda
    """

def processar_arguments(args):
    if '-h' in args or '--help' in args:
        return 'help', None
```

```

        # Processar arguments...
        return param1, param2

def accio_principal(param1, param2):
    try:
        # Lògica del programa...
        return "Resultat exitós"
    except Exception as e:
        return f"Error: {e}"

def main():
    if len(sys.argv) < 2:
        print("Error: Cal especificar arguments")
        print(mostrar_ajuda())
        return

    resultat = processar_arguments(sys.argv)

    if resultat[0] == 'help':
        print(mostrar_ajuda())
    else:
        missatge = accio_principal(resultat[0], resultat[1])
        print(missatge)

if __name__ == "__main__":
    main()

```

Funcions útils de Python per fitxers

Comptar elements

```

# Comptar ocurrencies d'una substring
text = "hola hola món"
text.count("hola")  # 2

```

```

# Comptar línies
len(linies)

```

Manipular llistes

```

# Inserir element
llista.insert(posicio, element)

```

```

# Afegir al final
llista.append(element)

```



```
# Extreure fragment  
fragment = llista[inici:final]
```

Treballar amb strings

```
# Convertir a minúscules  
text.lower()
```

```
# Comprovar si acaba amb substring  
text.endswith('\n')
```

```
# Comprovar si comença amb substring  
text.startswith('Hola')
```

```
# Dividir string  
paraules = text.split() # Per espais  
linies = text.split('\n') # Per salts de línia
```