

# Programmazione in Python per la bioinformatica

University of Verona

# Aprire un file

## Uso della funzione `open(filename, mode)`

- La funzione `f=open('myfile')` ritorna un oggetto file
- `f=open('myfile')` apre il file `myfile` in lettura (equivalente a `open('myfile', 'r')`)
- `f=open('myfile', 'w')` apre il file `myfile` in scrittura (cancella il file)
- `f=open('myfile', 'a')` apre il file `myfile` in scrittura preservando il contenuto. Le nuove linee vengono aggiunte in coda al file.

## Errori nell'apertura dei file

- se il file che voglio aprire non esiste l'interprete ritorna un errore
- posso gestire queste situazioni specificando cosa fare a fronte di questi errori
- utilizzo eccezioni:

```
try:
```

```
(codice che potrebbe creare errore)
```

```
except <nomeErrore>:
```

```
(codice che gestisce l'errore)
```

```
else:
```

```
(codice da eseguire se non si verifica errore)
```

# Errori di Input/Output, esempio

Programmazione  
in Python  
per la bioin-  
formatica

leggere e  
scrivere file

## Errori nell'apertura dei file, esempio

```
>>> f=open('nofile')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IOError: [Errno 2] No such file or directory: 'nofile'
>>> try:
...     f=open('nofile')
... except IOError:
...     print('il file non esiste')
...
il file non esiste
>>> █
```

# Errori di Input/Output, esempio gestione

Programmazione  
in Python  
per la bioin-  
formatica

leggere e  
scrivere file

## Errori nell'apertura dei file, esempio gestione

```
"""  
Esempio uso try ... except  
"""  
  
filename = input("inserisci nome file\n")  
while True:  
    try:  
        f = open(filename)  
    except IOError:  
        print("non posso aprire il file")  
        filename = input("inserisci nome file\n")  
    else:  
        print("letto: %s" % f.read())  
        f.close()  
        break
```

# Errori di Input/Output, esempio gestione

Programmazione  
in Python  
per la bioin-  
formatica

leggere e  
scrivere file

## Errori nell'apertura dei file, esempio gestione

```
inserisci nome file  
nofile  
non posso aprire il file  
inserisci nome file  
try-file.txt  
letto: successo!
```

# leggere un intero file

Programmazione  
in Python  
per la bioin-  
formatica

leggere e  
scrivere file

## Come leggere tutte le linee di un file

```
test-file.txt ✖  
Prima linea  
Seconda Linea|
```

```
>>> for line in f:  
...     print(line)  
...  
Prima linea
```

```
Seconda Linea
```

```
>>> f.read()
```

```
''
```

```
>>> █
```

# cambiare posizione all'interno di un file

Programmazione  
in Python  
per la bioin-  
formatica

leggere e  
scrivere file

## Usare il metodo seek(offset)

```
>>> f.seek(0)
0
>>> f.read()
'Prima linea\nSeconda Linea\n'
>>> f.seek(0)
0
>>> f.readline()
'Prima linea\n'
>>> █
```

# Scrivere in un file

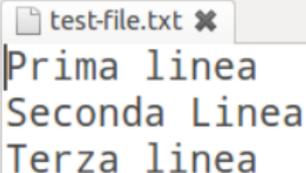
Programmazione  
in Python  
per la bioin-  
formatica

leggere e  
scrivere file

## Usare il metodo write(string)

Il metodo write scrive una stringa nel file e ritorna il numero di caratteri scritti

```
>>> f = open('test-file.txt', 'a')
>>> f.write('Terza linea')
11
>>> f.close()
>>> █
```



test-file.txt ✕  
Prima linea  
Seconda Linea  
Terza linea

# File con formato fasta

Programmazio  
in Python  
per la bioin-  
formatica

leggere e  
scrivere file

## File Fasta contiene sequenze di dna

```
myseq.fasta ✕  
>id1 sequence with highest 'a' count|  
CATGCTACGGTGCTAAAAGCATTACGCCCTATAGTGATTTTCGAGACATACTGTGTTT  
TTAAATATAGTATTGCC  
>id2 description of id2  
ATGTGTGTCCGTTGTGTAA  
AGTGTGTccccgtggttATg  
gtagatTTTTga  
>id3 description of id3  
ccccagtggggagtagggc  
AAAcgtatAA
```

# Leggere un file fasta

Programmazio  
in Python  
per la bioin-  
formatica

leggere e  
scrivere file

## codice per leggere un file in formato fasta

```
def read_fasta(filename) :  
    """  
    legge tutte le sequenze contenute in un  
    file fasta e associa gli id alle sequenze  
    """  
    seqs={}  
    try :  
        f = open(filename)  
    except IOError:  
        print("the file does not exists")  
        return seqs  
  
    for line in f :|  
        #toglie \n se presente  
        line = line.rstrip()  
        #controlla se header  
        if line.startswith('>') :  
            words=line.split() #divide una stringa  
            name=words[0][1:] #toglie il >  
            seqs[name]=''  
        else : #se non e' un header  
            seqs[name] = seqs[name] + line  
  
    f.close()  
    return seqs
```

Vedere il modulo [fastutil.py](#)

# Gestire il dizionario delle sequenze

Programmazione  
in Python  
per la bioin-  
formatica

leggere e  
scrivere file

codice per processare tutte le coppie di un dizionario

```
def print_fasta_seqs(filename):  
    """  
    assumendo che filename sia un file in formato  
    fasta, legge tutte le sequenze in un dizionario  
    stampa tutti gli elementi del dizionario  
    """  
    seqs = read_fasta(filename)  
    for name, seq in seqs.items() :  
        print(name, seq)
```

Vedere il modulo [fastutil.py](#)

# Esercizi su scrittura lettura files

Programmazione  
in Python  
per la bioin-  
formatica

leggere e  
scrivere file

## Eserc files

- Q1** Scrivere una funzione che dato un filename ritorna l'estensione del file. Ad esempio se l'input e' 'myfile.tar.gz' la funzione dovrebbe ritornare '.gz'. **SUGGERIMENTO:** usare il metodo *split(...)* delle stringhe. Scaricare e modificare [eserc-files-Q1.py](#) [Sol: [eserc-files-Q1.sol](#)]
- Q2** Scrivere una funzione che dato il nome di un file in formato fasta ed il nome di un file di output, copia solo le sequenze di dna contenenti un codone di terminazione. **SUGGERIMENTO:** usare la funzione `read_fasta` del modulo `fastutil.py`. Scaricare e modificare [eserc-files-Q2.py](#) [Sol: [eserc-files-Q2.sol](#)].

# Esercizi su scrittura lettura files

Programmazio  
in Python  
per la bioin-  
formatica

leggere e  
scrivere file

## Eserc files

**Q3** Scrivere una funzione che dato il nome di un file fasta ritorna l'id associato alla sequenza che ha un numero di occorrenze del nucleotide 'a' piu' alto. **SUGGERIMENTO:** usare la funzione `read_fasta` del modulo `fastutil.py`. Scaricare e modificare [eserc-files-Q3.py](#) [Sol: [eserc-files-Q3.sol](#)]