

Esercitazione 8

FILE e DIZIONARI

File

- I file vengono gestiti in modo molto semplice e simile al C
- “open(nomefile[, modo])” apre “nomefile” in modalità “modo” (“r” è il valore di default) e ritorna un oggetto di tipo “file”
- I metodi principali degli oggetti file sono:
 - “read([n])” ritorna “n” byte dal file. Se “n” è omesso legge tutto il file
 - “readline()” ritorna una riga
 - “readlines()” ritorna una lista con le righe rimanenti nel file
 - “write(str)” scrive “data” sul file

File

- “writelines(list)” scrive tutti gli elementi di “list” su file
- “close()” chiude il file (richiamato automaticamente dall’interprete)
- “flush()” scrive su disco i dati presenti in eventuali buffer
- “seek(offset[, posiz])” muove di “offset” byte da “posiz”. I valori di posiz sono:
 - 0: dall’inizio del file (valore di default)
 - 1: dalla posizione corrente
 - 2: dalla fine del file (“offset” è normalmente negativo)
- “tell()” ritorna la posizione corrente
- “truncate([n])” tronca il file a non più di “n” byte. Il valore di default è la posizione corrente

File

- Leggi_matrice.py
- Leggi_file.py
- Scrivere su file:

With `open('file2.txt','w')` as `f`:

```
f.write('ciao')
```

Liste

$L = [1, 2, 3, 4, 5, 6, 7, 8, 9]$

$A = [x \text{ for } x \text{ in } L]$

$B = [x \text{ for } x \text{ in } L \text{ if } x > 5]$

$C = [x \text{ for } x \text{ in } L \text{ if } x \% 2 == 0]$

$D = [3, 5, 6]$

$E = [x \text{ for } x \text{ in } L \text{ if } x \text{ not in } D]$

$F = [(x, x) \text{ for } x \text{ in } L]$

$G = [\text{str}(x) \text{ for } x \text{ in } L]$

Dizionari

- Associano ad una chiave un valore
- Creati nella forma "{chiave1: val1, chiave2: val2}"
- **{"nome": "Mario", "cognome": "Rossi"}**
- L'accesso e l'inserimento di elementi avviene come per le liste
- **{"a": 1, "b": 2}["a"]**
- **{"a": 1, "b": 2}["X"]** !!!!!!!!!!!!!!!!!!!!!!!!!!!!!
- **{ }["X"] = 2**
- • Le chiavi devono essere immutabili
- • Le chiavi non vengono tenute ordinate!

Dizionari

- I metodi principali dei dizionari sono:
- “D.clear()” elimina tutti gli elementi dal dizionario
- “D.copy()” restituisce una copia di “D”
- “D.has_key(k)” restituisce 1 se “k” è nel dizionario, 0 altrimenti. Si può usare anche l'operatore “in”
- “D.items()”, “D.keys()”, “D.values()” restituiscono rispettivamente una lista con le tuple “(chiave, valore)”, la lista delle chiavi, la lista dei valori
- “D.update(D2)” aggiunge le chiavi e valori di “D2” in “D”
- “D.get(k, d)” restituisce “D[k]” se la chiave è presente nel dizionario, “d” altrimenti. Il valore di default di “d” è “None”

Dizionari

- 1) Usando i dizionari, scrivere un programma che verifica se una lista non ha duplicati
- 2) Data una frase (o un testo), costruire un dizionario con chiavi le lettere dell'alfabeto e come valori le occorrenze delle lettere
- 3) Creare un dizionario studenti con chiave matricola e valore una lista di coppie (esame,voto). Cercare:
 - il numero di esami dello studente 12345
 - la media dello studente 12345
 - gli studenti che hanno sostenuto l'esame 'Fisica'
 - gli studenti con più di 3 esami

File

- Stampare solo le righe pari lette da un file
- Leggere un file di testo e salvare il contenuto su un nuovo file eliminando le righe che superano gli 80 caratteri.
- Salvare su file il seguente dizionario
 {1:[123], 2:[0,9,8], 3:[3,4,6]}
- Scrivere un blocco di codice che legge un file html e salva tutti gli URL su un file di testo

IPOTESI: nel file html tutti i link sono racchiusi nel blocco **href="http://www.google.it"** Usare la funzione `objstr.index(pattern,indicedipartenza)` per trovare la prima occorrenza del pattern