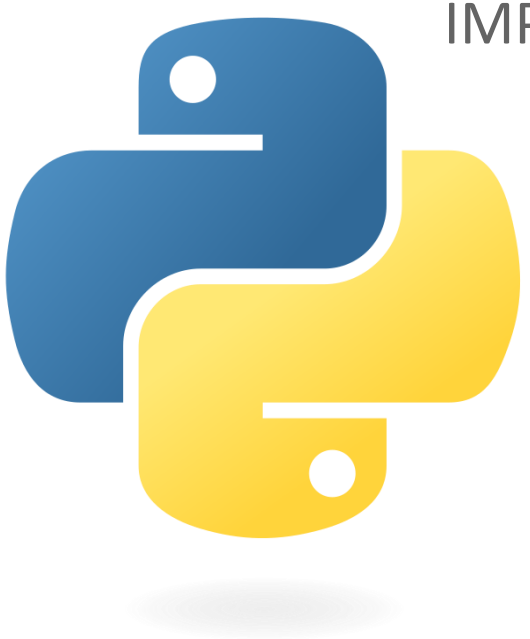


IMPARIAMO A PROGRAMMARE IN python™

LEZIONE 2

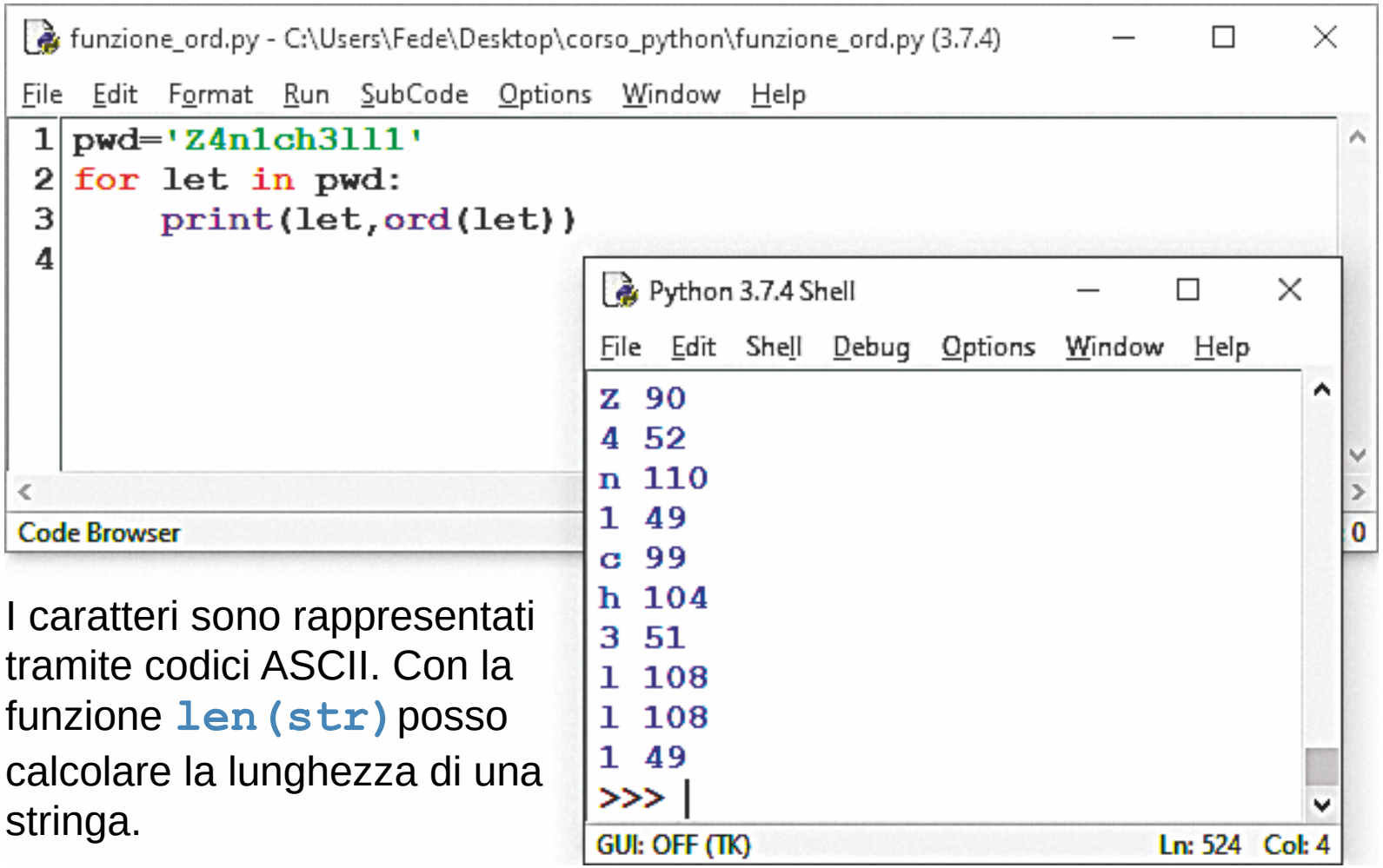
Capitolo 2 – Le stringhe e le strutture di dati complesse

Indice

2.1 Lavorare con le stringhe

Indice

In Python, una stringa è una sequenza di caratteri racchiusa tra apici e ordinata in base a un indice che parte da 0.



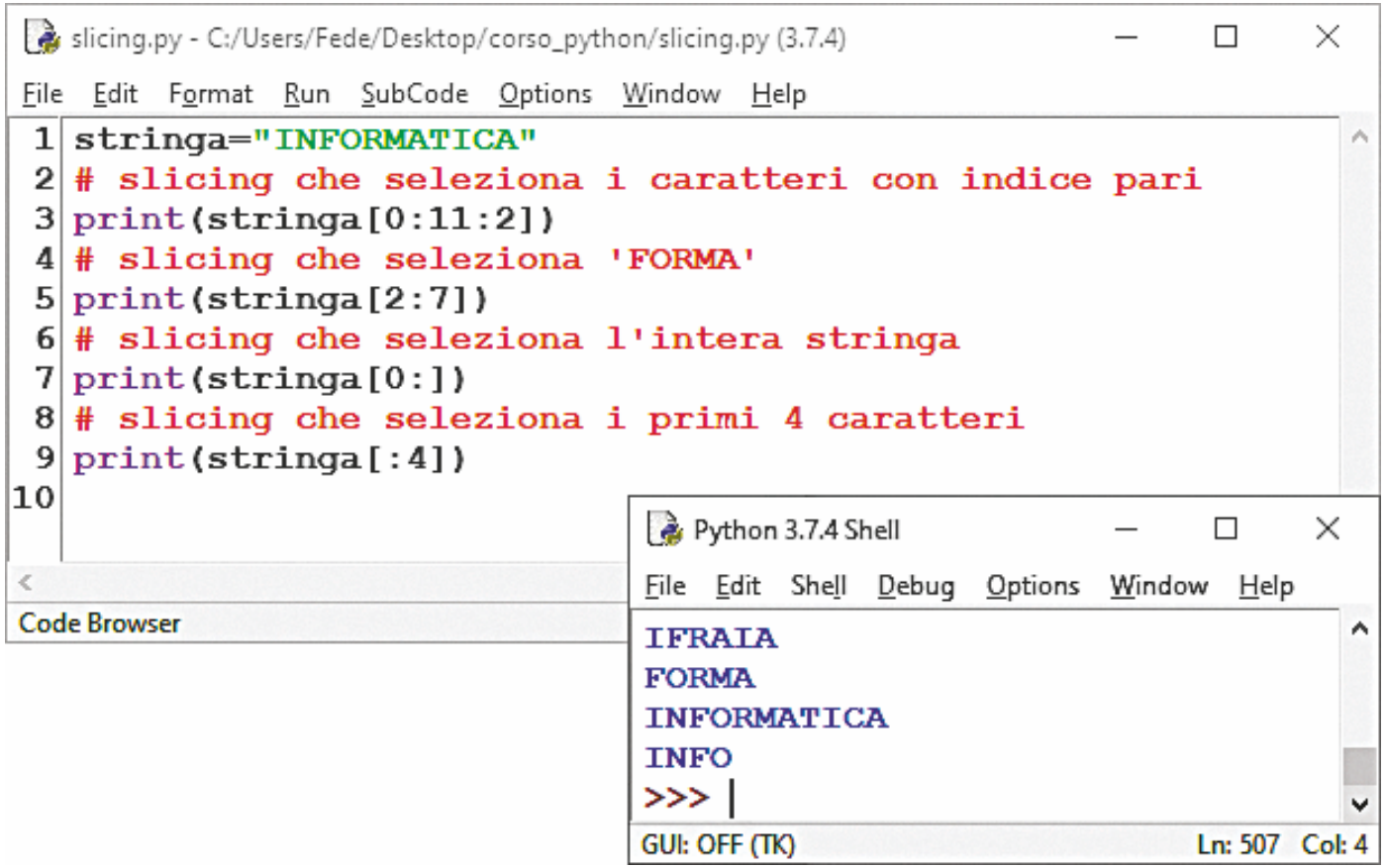
I caratteri sono rappresentati tramite codici ASCII. Con la funzione `len(str)` posso calcolare la lunghezza di una stringa.

2.1 Lavorare con le stringhe

Indice

Lo `slicing` permette di selezionare parti di una stringa.

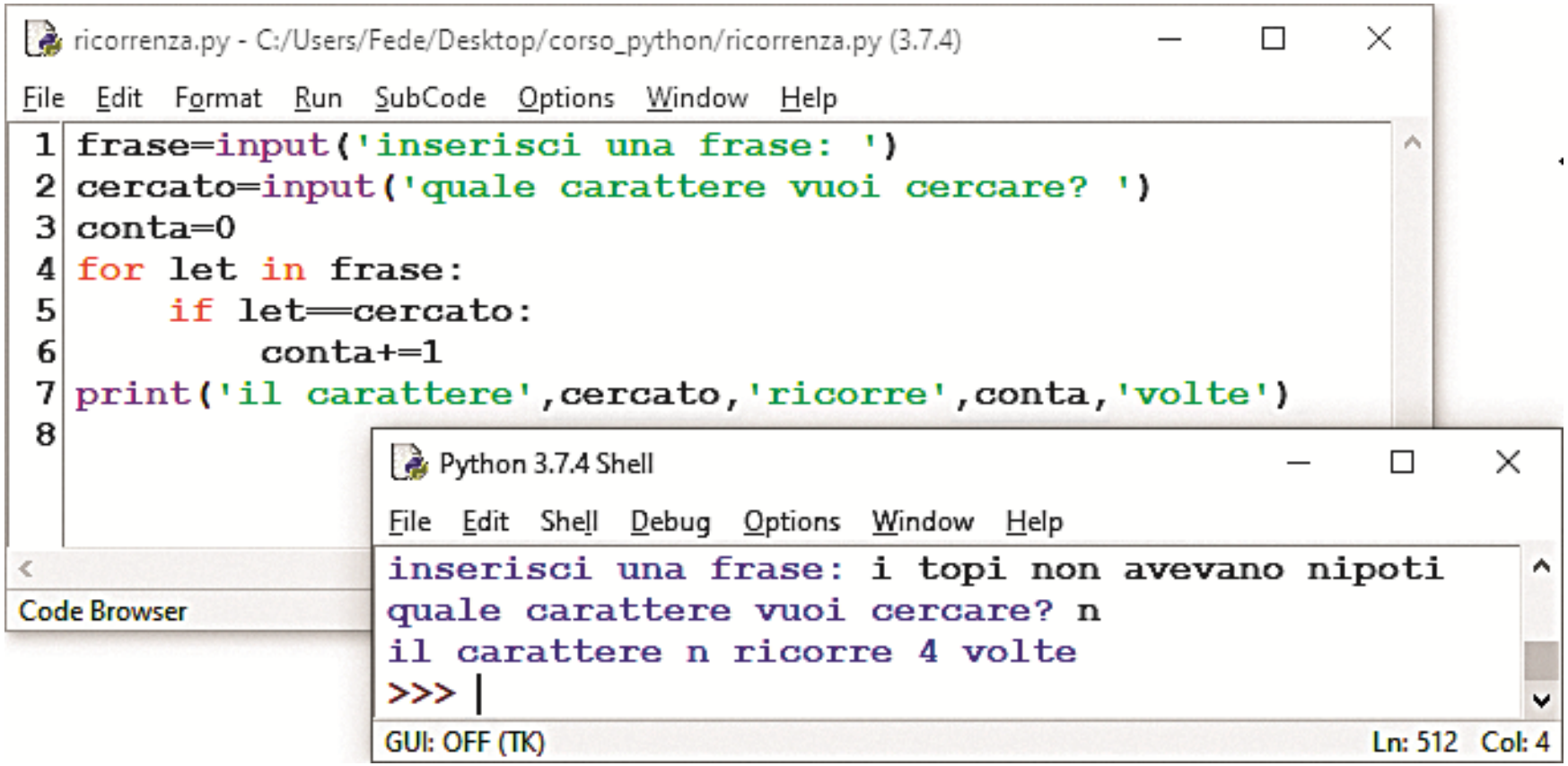
nome_stringa[inizio:fine:passo]



2.1 Lavorare con le stringhe

Indice

Per cercare ricorrenze o sottostringhe si usa la parola chiave `in`.

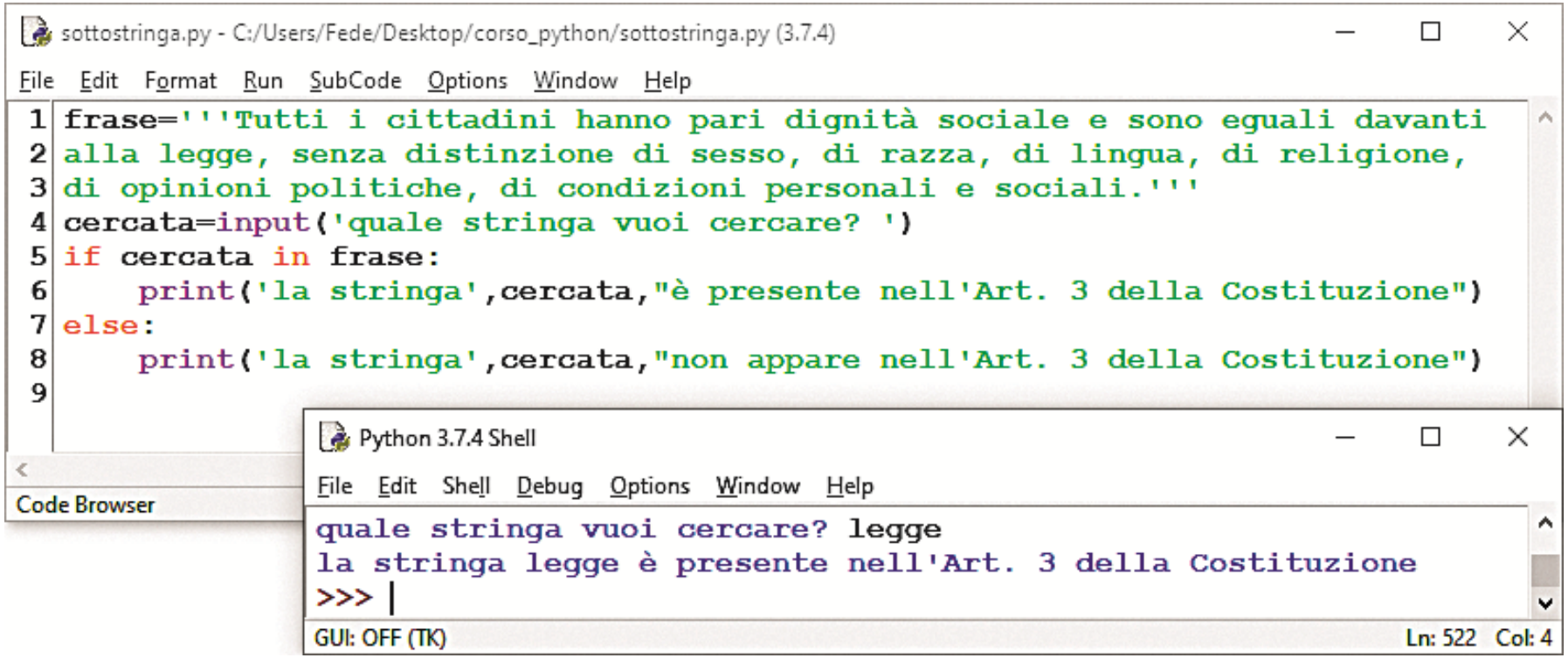


Ricerca di un carattere

2.1 Lavorare con le stringhe

Indice

Per cercare ricorrenze o sottostringhe si usa la parola chiave `in`.

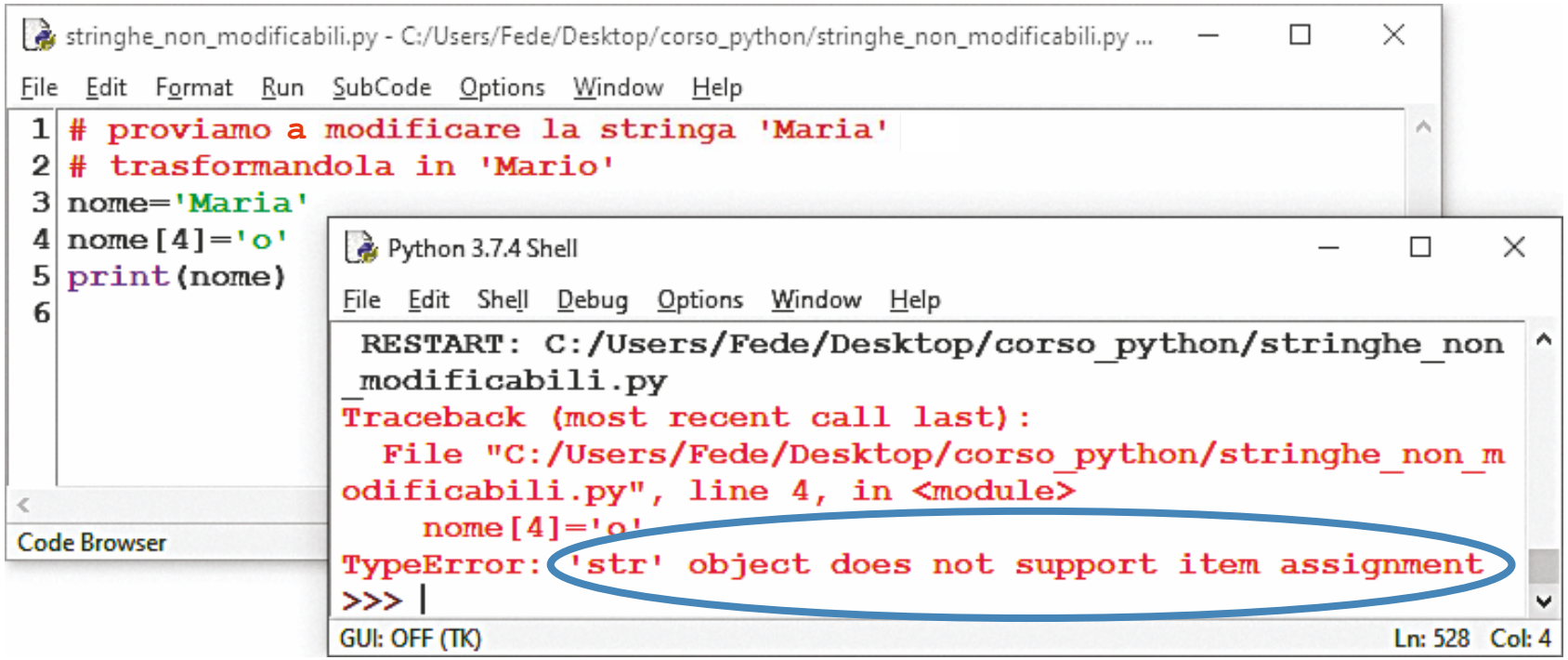


Ricerca di una sottostringa

2.1 Lavorare con le stringhe

Indice

Le stringhe in Python NON sono modificabili.

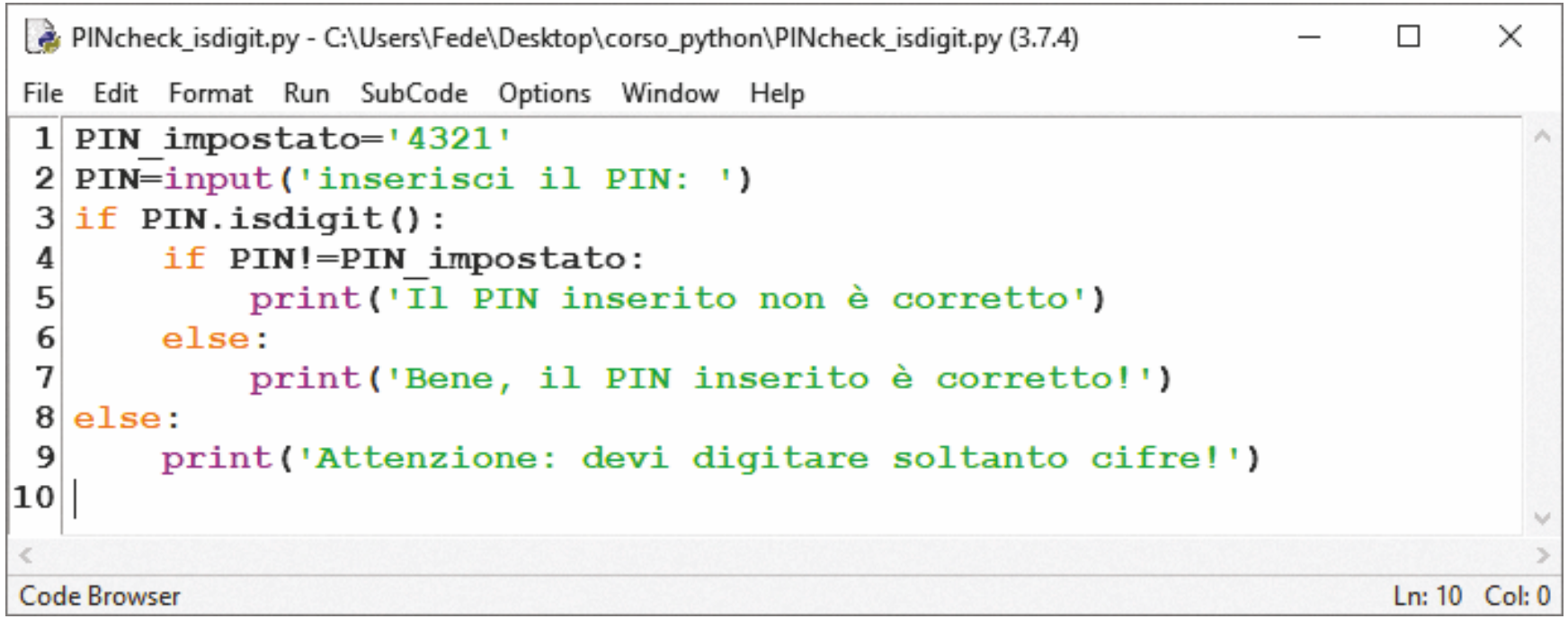


Il tentativo di cambiare il contenuto di una stringa fornisce come risultato un errore.

2.2 L'oggetto stringa e i suoi metodi

Indice

In Python le stringhe sono oggetti dotati di metodi specifici.



Metodi comuni per il trattamento delle stringhe sono **`isalpha()`**, **`isalnum()`**, **`islower()`**, **`isupper()`**.

2.2 L'oggetto stringa e i suoi metodi

Indice

Si possono creare stringhe a partire da altre stringhe.

upper_lower.py - C:\Users\Fede\Desktop\corso_python\upper_lower.p...

File Edit Format Run SubCode Options Window Help

```
1 nome_ins=input('inserisci il tuo nome: ')
2 # crea una stringa con tutti i caratteri maiuscoli
3 nome_mai=nome_ins.upper()
4 print(nome_mai)
5 # stampa il nome con tutti i caratteri minuscoli
6 print(nome_ins.lower())
7
```

Code Browser

Python 3.7.4 Shell

File Edit Shell Debug Options Window Help

```
inserisci il tuo nome: Karim
KARIM
karim
>>>
```

GUI: OFF (TK) Ln: 8 Col: 4

Il metodo **`replace()`** permette di sostituire una sottostringa con un'altra, anche di lunghezza diversa.

`find()` e **`index()`** restituiscono la posizione di una sottostringa, **`count()`** conta le ricorrenze.

2.3 Le liste e le matrici

Indice

Una lista è un oggetto che contiene una serie di dati, anche di tipo diverso.

lista.py - C:/Users/Fede/Desktop/corso_python/lista.py (3.7.4)

File Edit Format Run SubCode Options Window Help

```
1  voti=['Franco', 5.5, 'Anna', 8]
2  print('questa è la lista originale:',voti)
3  # verifichiamo i tipi di dati contenuti
4  for el in voti:
5      print(el,'è un dato di tipo',type(el))
6  # modifichiamo la lista
7  voti[0]=input('inserisci il tuo nome: ')
8  voti[1]=float(input('inserisci il tuo voto: '))
9  print('questa è la lista modificata:',voti)
10 # stampiamo singoli elementi della lista
11 print('i nomi sono',voti[0],'e',voti[2])
12 print('la media dei voti è',(voti[1]+voti[3])/2)
13
```

Code Browser

Dichiarazione della lista

Python 3.7.4 Shell

File Edit Shell Debug Options Window Help

```
questa è la lista originale: ['Franco', 5.5, 'Anna', 8]
Franco è un dato di tipo <class 'str'>
5.5 è un dato di tipo <class 'float'>
Anna è un dato di tipo <class 'str'>
8 è un dato di tipo <class 'int'>
inserisci il tuo nome: Edisa
inserisci il tuo voto: 7.5
questa è la lista modificata: ['Edisa', 7.5, 'Anna', 8]
i nomi sono Edisa e Anna
la media dei voti è 7.75
>>>
```

GUI: OFF (TK) Ln: 124 Col: 4

2.3 Le liste e le matrici

Indice

Esistono diversi modi di creare una lista.

```
voti_str='Franco 5.5 Anna 8':
```

```
voti=voti_str.split()
```

```
voti=['Franco','5.5','Anna','8']
```

```
mia_lista=['nero']*3
```

```
['nero','nero','nero'].
```

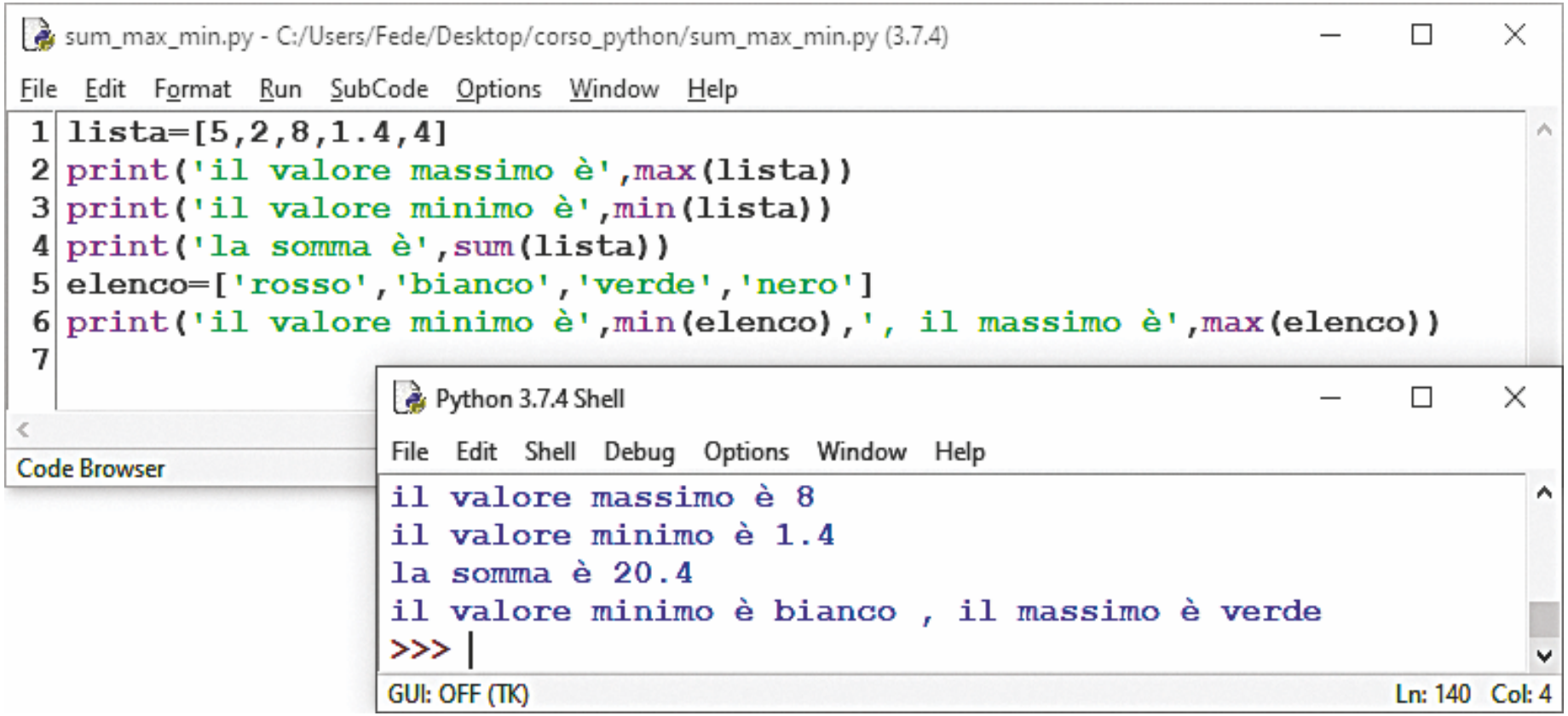
```
mia_lista=list(range(5))
mia_lista=list(range(5,51,5))
```

Numeri tra 0 e 4
Tabellina del 5

2.3 Le liste e le matrici

Indice

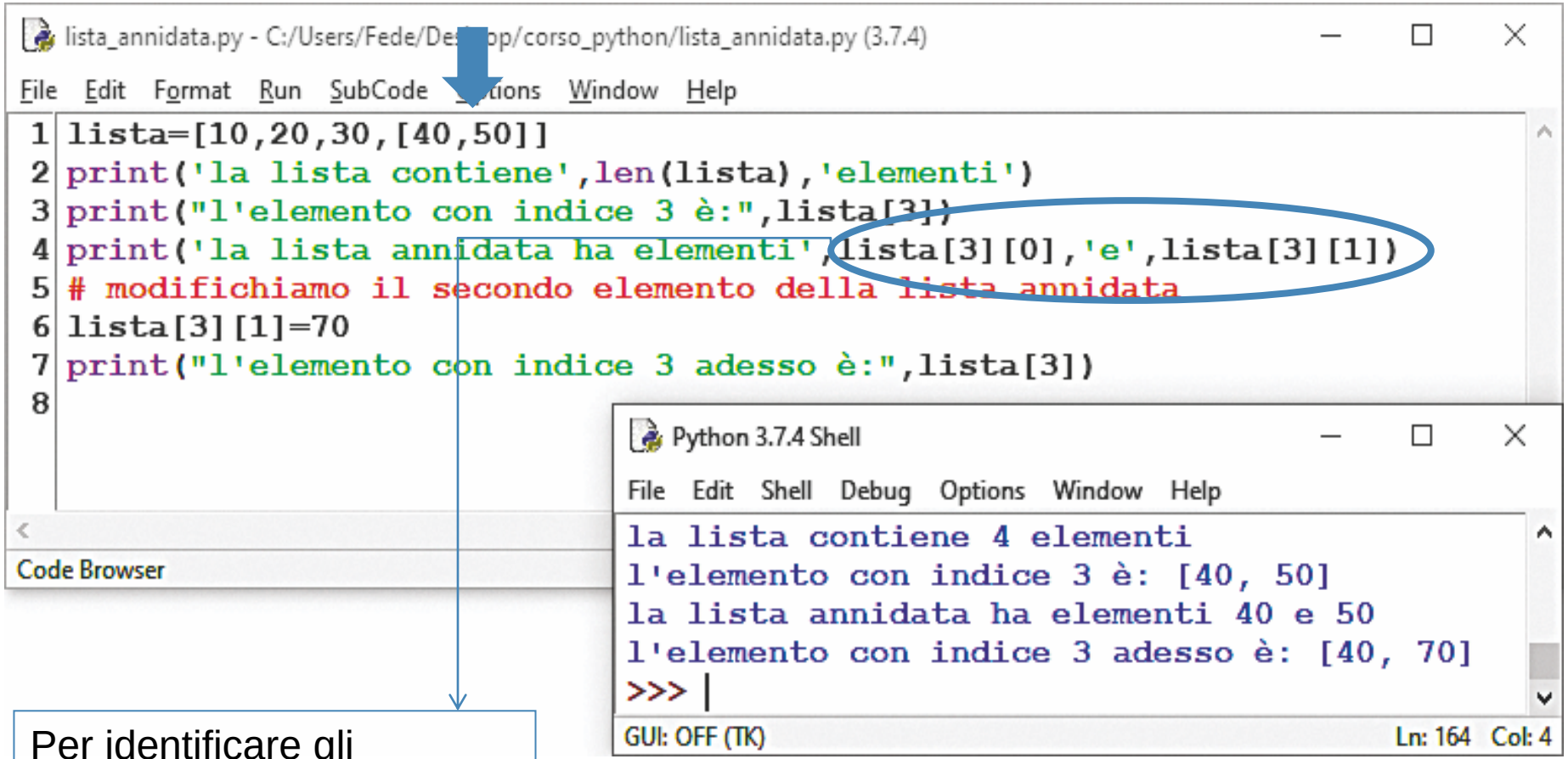
Alle liste posso applicare le funzioni built-in.



2.3 Le liste e le matrici

Indice

Una lista annidata è una lista contenuta come elemento in un'altra lista.



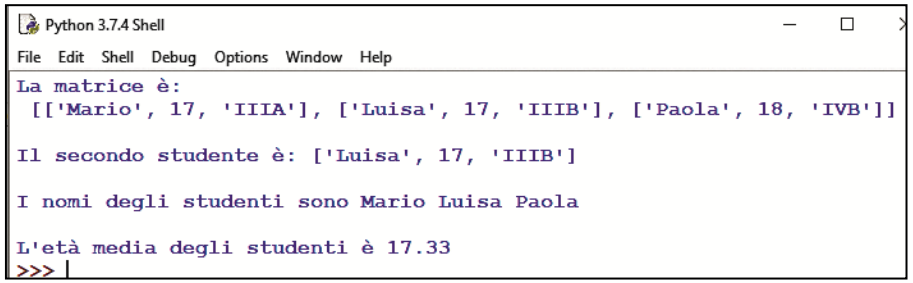
Per identificare gli elementi si usano 2 indici

2.3 Le liste e le matrici

Indice

Una lista che contiene solo liste tutte della stessa lunghezza è detta matrice.

```
# definiamo la matrice studenti
studenti=[['Mario',17,'IIIA'],['Luisa',17,'IIIB'],['Paola', 18,'IVB']]
indice1,elem=0,0
# stampiamo l'intera matrice
print('La matrice è:\n',studenti,'\n')
# stampiamo i dati relativi a un singolo studente
print('Il secondo studente è:',studenti[1],'\n')
# stampiamo i nomi di tutti gli studenti su una sola riga
print("I nomi degli studenti sono",end=' ')
while elem<len(studenti):
    print(studenti[indice1][0],end=' ')
    indice1+=1
    elem+=1
print('\n')
# stampiamo l'età media degli studenti
indice1,anni,elem=0,0,0
while elem<len(studenti):
    anni+=studenti[indice1][1]
    indice1+=1
    elem+=1
print("L'età media degli studenti è",format(anni/len(studenti),'.2f'))
```

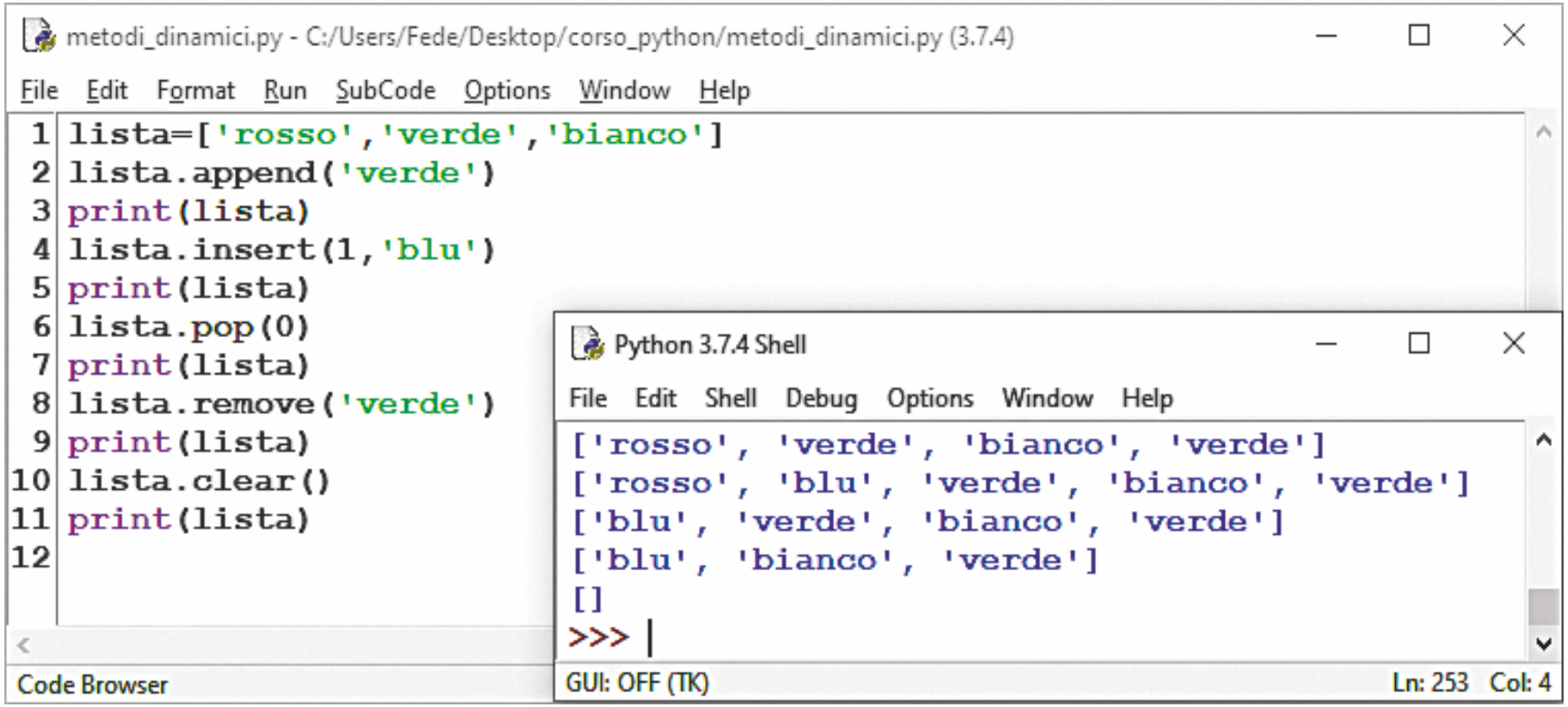


I dati sono trattati come se fossero contenuti in una tabella

2.3 Le liste e le matrici

Indice

Le liste sono oggetti che possono essere elaborate usando appositi metodi.



Con i metodi **`reverse()`** e **`sort()`** si possono rispettivamente rovesciare e ordinare una lista.

2.4 Tuple, dizionari e set

Una tupla contiene una sequenza di dati ordinati in base a un indice, e NON può essere modificata.

Si può creare scrivendo:

```
tp='alfa',37,'tre'
```

O alternativamente:

```
tp=('alfa',37,'tre')
```

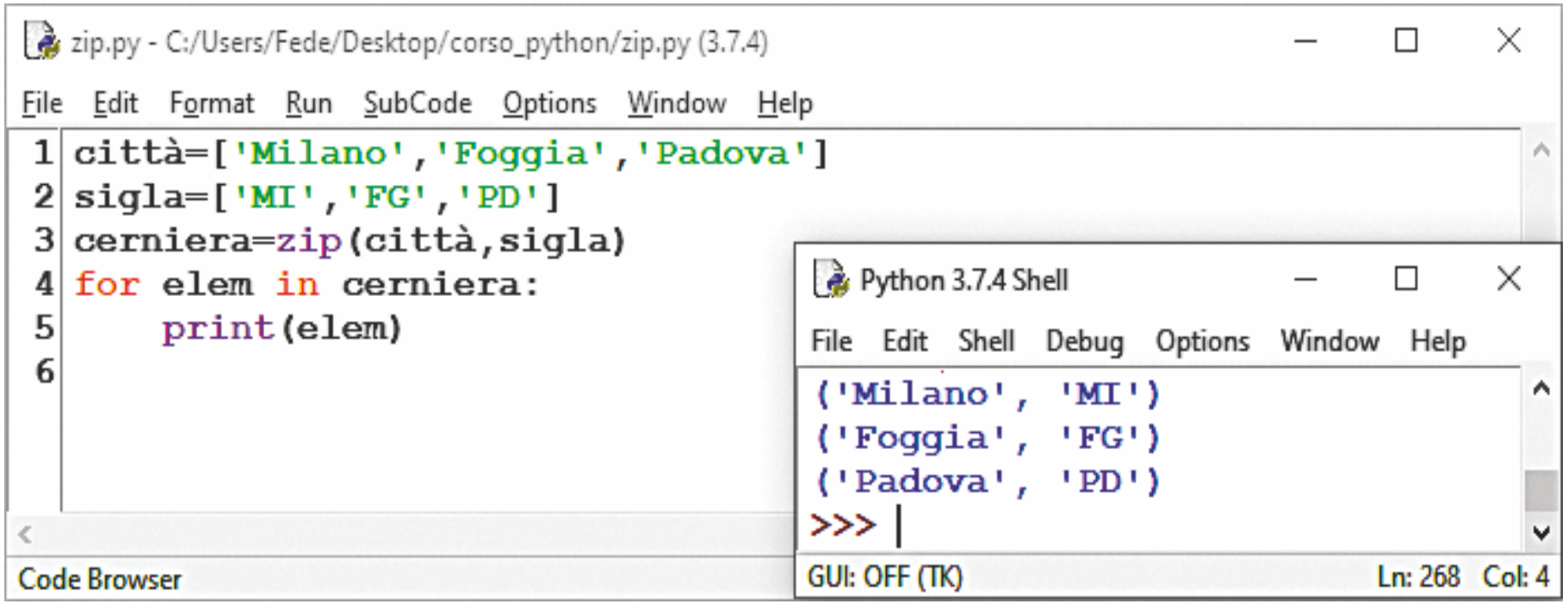
La funzione **`tuple()`** converte liste, stringhe o altri oggetti iterabili in tuple.

```
tp=tuple('abc')
```

2.4 Tuple, dizionari e set

Indice

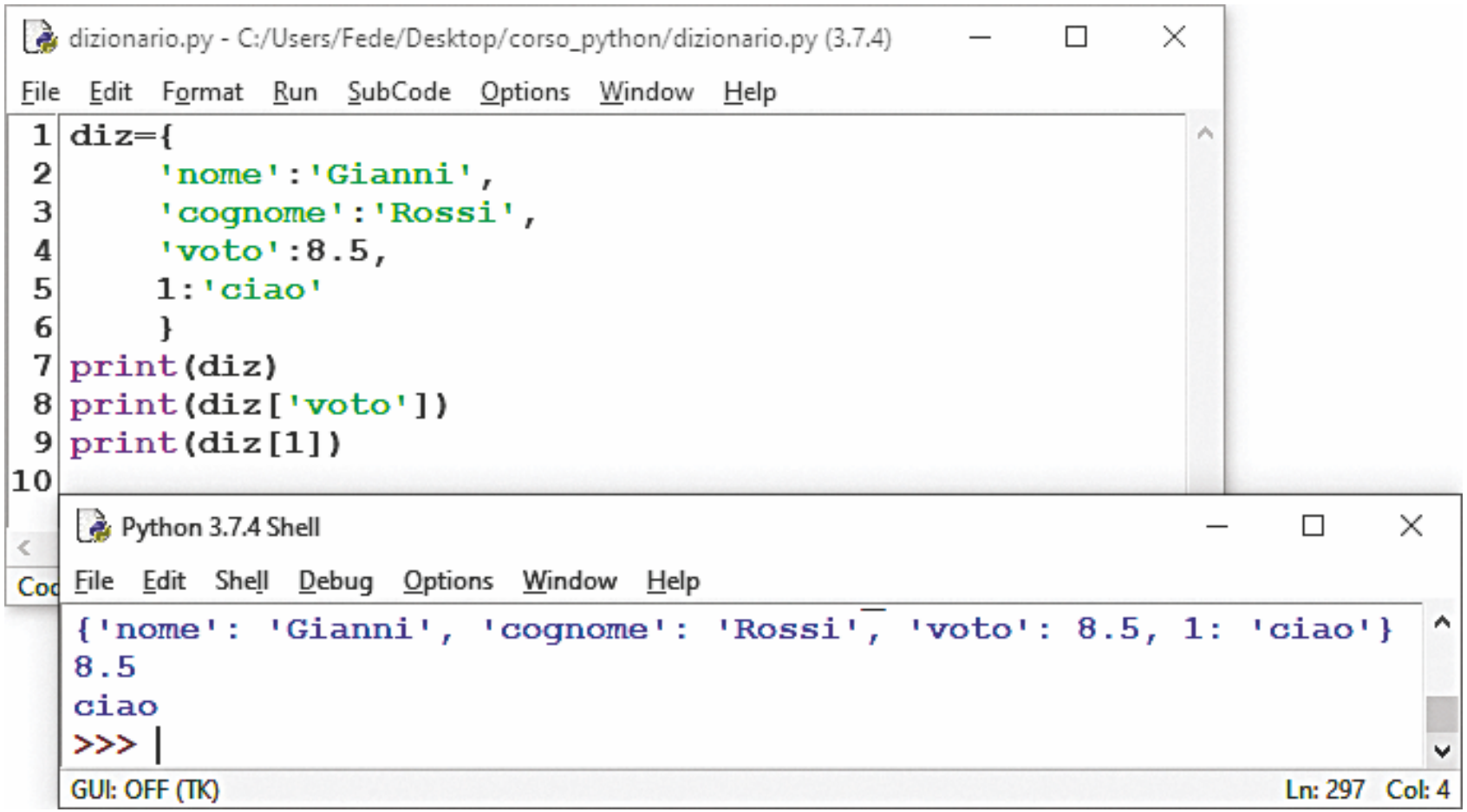
La funzione `zip`() accetta come argomenti oggetti iterabili e restituisce gli elementi corrispondenti associati sotto forma di tuple.



2.4 Tuple, dizionari e set

Indice

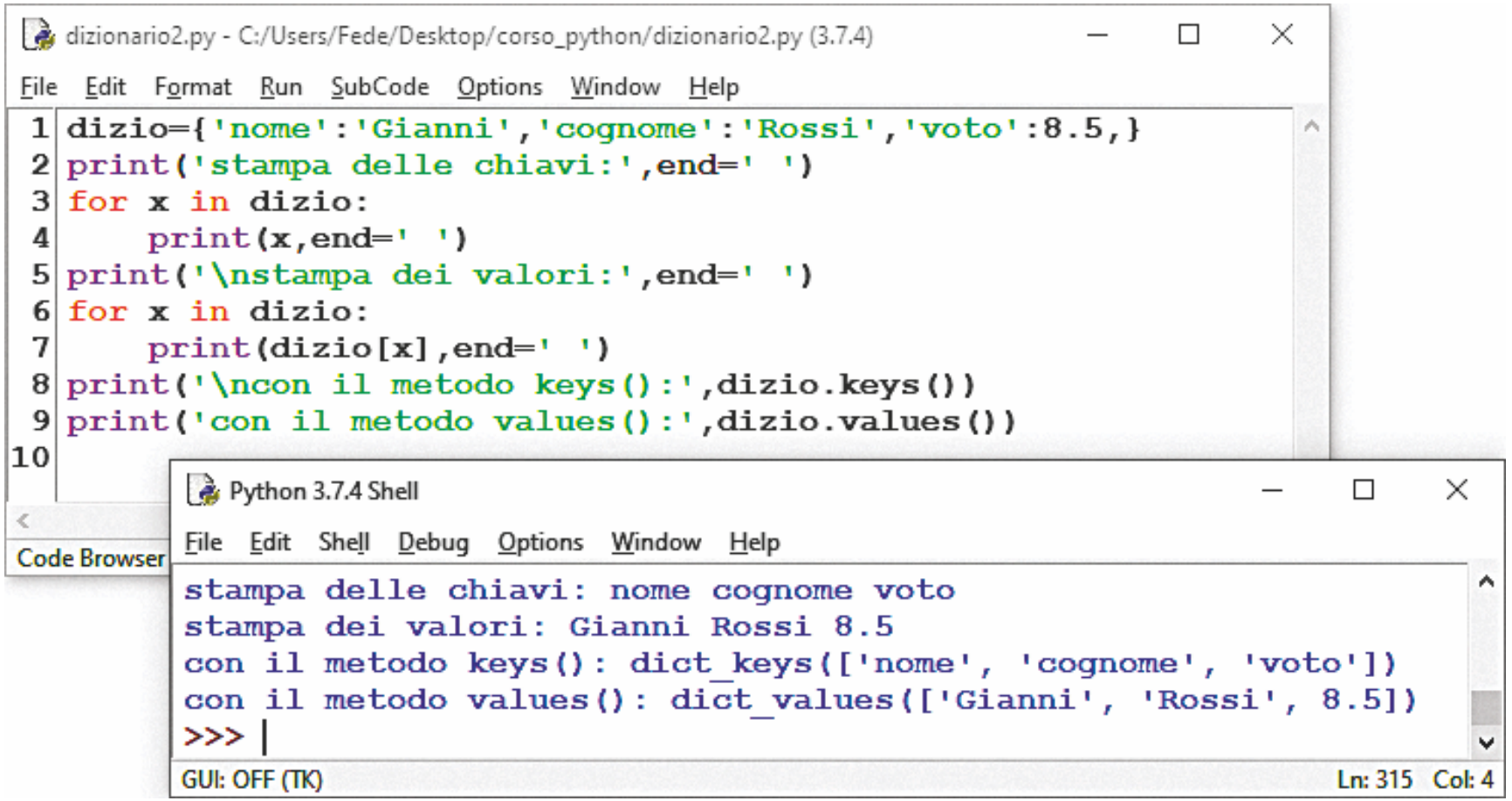
Un dizionario è un oggetto modificabile che contiene dati di vario tipo non ordinati ma indicizzati da chiavi scelte dal programmatore.



2.4 Tuple, dizionari e set

Indice

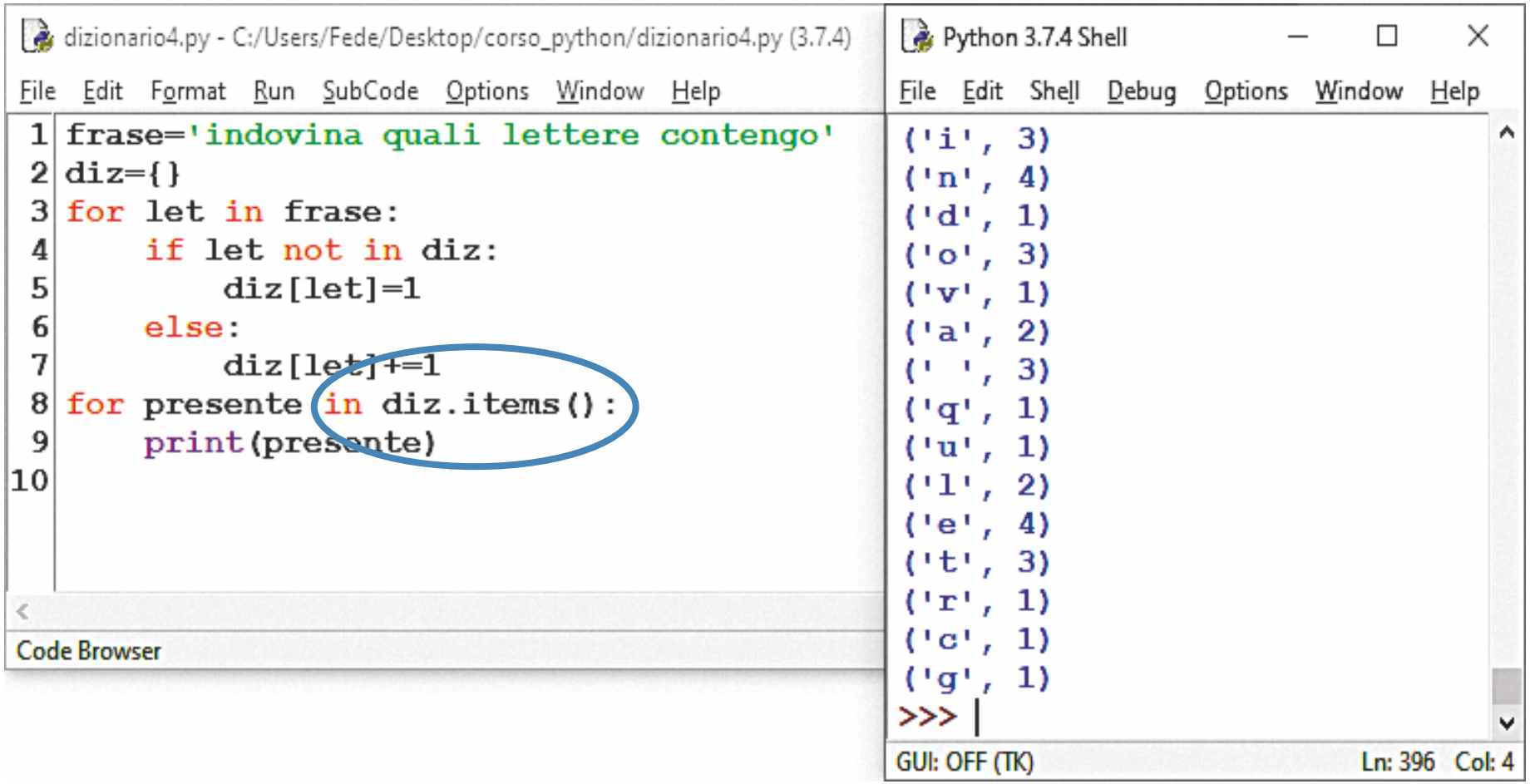
Un dizionario può essere elaborato per mezzo di un ciclo `for`.
I metodi `keys`() e `values`() selezionano le chiavi e i valori.



2.4 Tuple, dizionari e set

Indice

A un dizionario è possibile applicare dei metodi. Risulta molto utile il metodo `items()`.

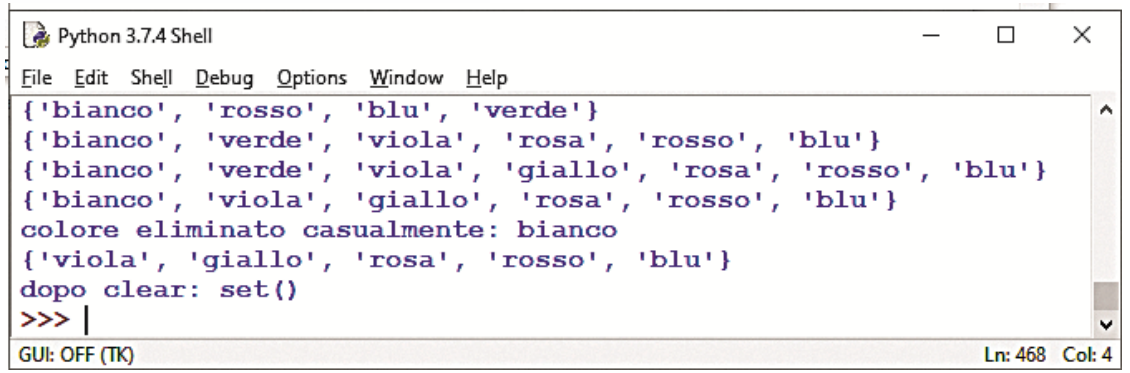


2.4 Tuple, dizionari e set

Indice

Un set è un oggetto modificabile che contiene dati di vario tipo, non indicizzati né ordinati con valori tutti diversi tra loro. Può essere manipolato utilizzando gli appositi metodi.

```
colori={'bianco','rosso','verde','blu'}
colori2={'viola','blu','rosa','verde'}
print(colori)
colori.update(colori2)
print(colori)
colori.add('giallo')
print(colori)
colori.remove('verde')
print(colori)
print('colore eliminato casualmente:',colori.pop())
print(colori)
colori.clear()
print('dopo clear:',colori)
```



Esistono metodi che calcolano le operazioni tipiche degli insiemi:
unione, intersezione, differenza, differenza simmetrica e verifica di inclusione.