

Strutture dati, condizionali, cicli

... e non solo

Il signor Libraio gestisce la sua libreria da più di 20 anni ormai, e la sua attività è cresciuta! Inizialmente vendeva solo due libri: *Il Signore degli Anelli* e *1984*. Più la sua libreria cresceva, più il signor Libraio faceva fatica a tenere traccia dei libri in magazzino. Si è rivolto dunque a noi per chiederci di sviluppare un programma che lo aiuti nel lavoro di gestione del magazzino che finora ha fatto completamente a mano. Ecco la sua mail di richiesta di aiuto:



Ogni giorno, quando la corriera con i nuovi ordini arriva, leggo sull'etichetta di ogni scatolone quali libri sono contenuti al suo interno e quanti ne sono contenuti per ogni tipo. Aggiorno la quantità di volumi sul mio registro, in corrispondenza del titolo corretto ovviamente. Se ciò che arriva non è presente sul mio registro, allora aggiungo una nuova riga e completo tutte le informazioni riguardo il libro. Le informazioni che mi sono più utili sono il titolo, l'autore, l'edizione (segno l'anno di stampa), il prezzo e il genere.

Tuple

Le tuple sono usate per memorizzare più valori in una singola variabile. Sono delle collezioni ordinate e non modificabili. Vengono scritte con le parentesi tonde `()`, e gli elementi sono separati da una virgola.

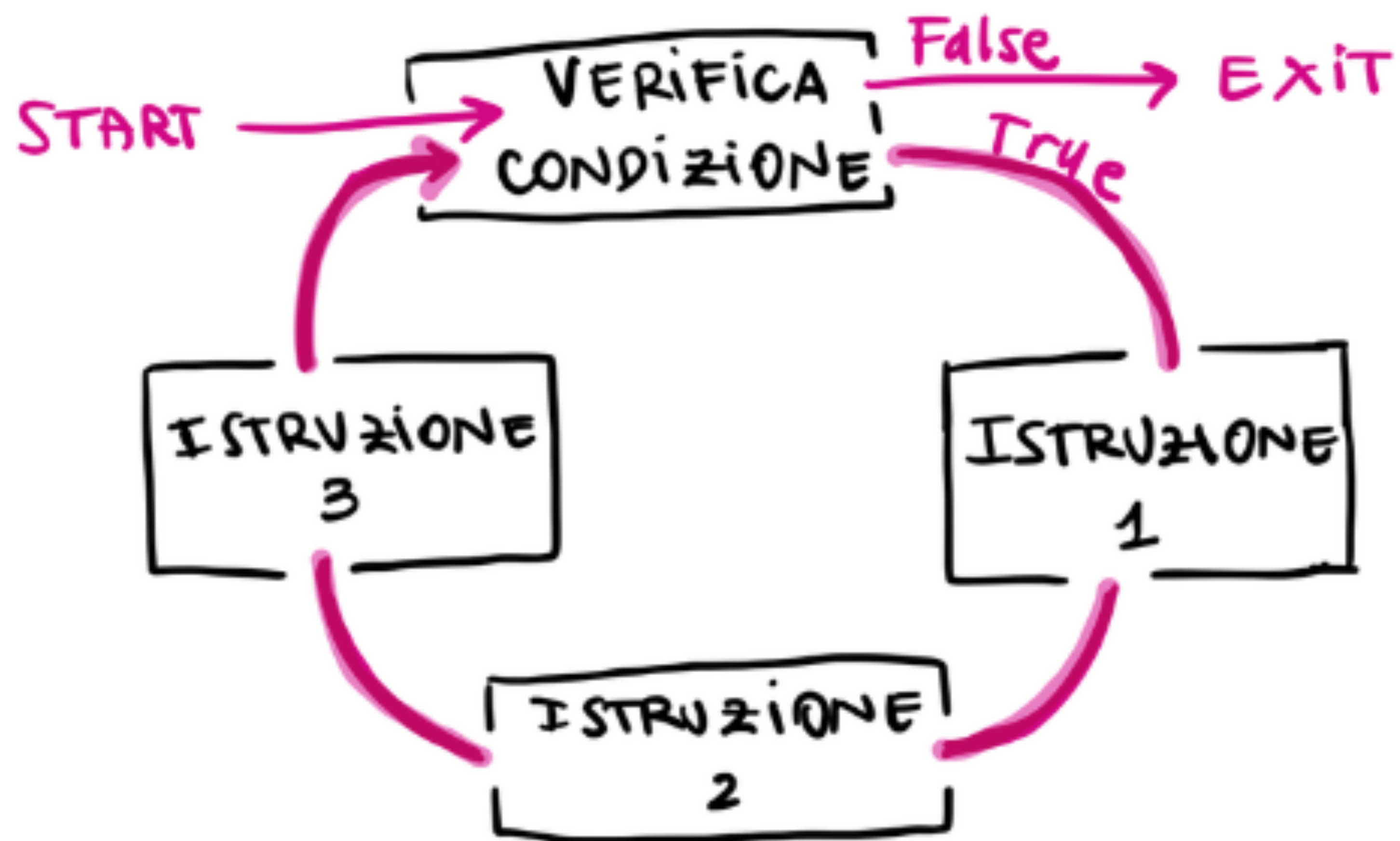
```
myTuple = (1, "Ciao", 8.1, (5, 7))
```


Liste

Le liste, come le tuple, vengono usate per contenere più elementi in una stessa variabile. Sono delle collezioni di elementi ordinati e mutevoli (si possono cambiare gli elementi di una lista, se ne possono aggiungere e togliere). Sono segnate dalle parentesi quadre `[]` e gli elementi sono separati da una virgola.

```
myList = [1, "Ciao", 8.1, (5, 7)]
```

- Un ciclo racchiude una porzione di codice che deve essere ripetuta fintanto che una *condizione* sia verificata. Ogni volta che viene eseguita una ripetizione, tale condizione viene controllata nuovamente: se rimane verificata, allora si prosegue con un'altra ripetizione; altrimenti, il ciclo termina e si prosegue con l'esecuzione delle istruzioni successive.



```
contatore = 0
while contatore < 100:
    print(contatore)
    contatore = contatore + 1
```

oppure

```
funziona = True
while funziona:
    print("ripeto il codice")
    funziona = False
print("non ripeto il codice")
```

oppure

```
contatore = 0
funziona = True
while contatore < 100 or funziona:
    print(contatore)
    print("ripeto il codice")
    funziona = False
    contatore = contatore + 1
```

While

Il while è il costrutto che implementa esattamente ciò che abbiamo definito sopra: esegue una porzione di codice finché la condizione è vera (True). La sintassi è la seguente.

```
while condizione:
```

```
    istruzione1
```

```
    istruzione2
```

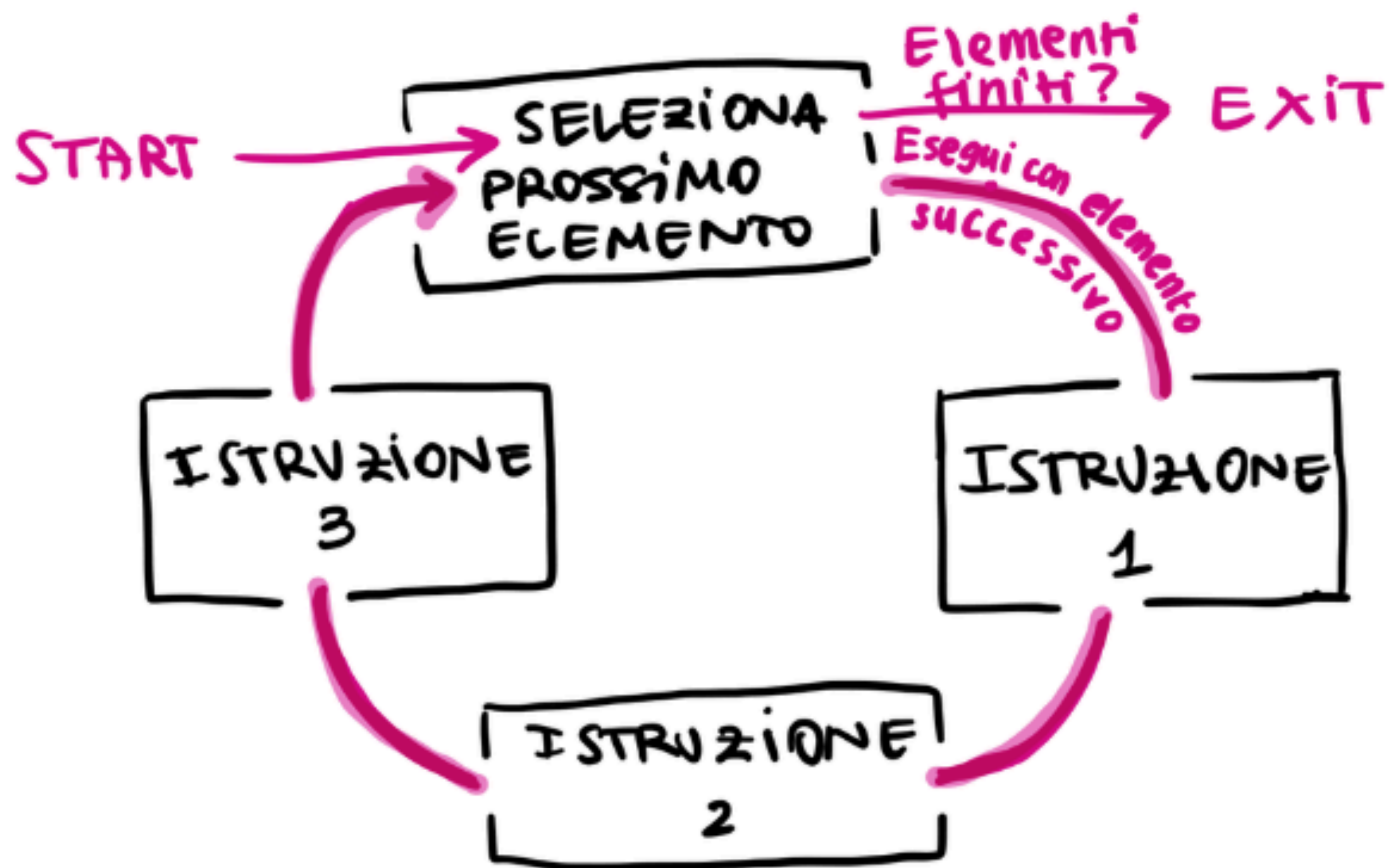
```
    istruzione3
```


For

Un ciclo for viene usato per ripetere la stessa porzione di codice, ogni volta su un elemento diverso tra quelli in strutture dati che contengono sequenziali, come le liste, le tuple, i dizionari, gli insiemi.

La sintassi è la seguente:

```
for item in lista:  
    print(item)
```



```
print('Enter your name:')  
x = input()  
print('Hello, ' + x)
```



if

Il costrutto condizionale if permette di verificare una condizione. Se questa condizione ha valore True, allora il codice racchiuso subito dopo il costrutto if verrà eseguito, altrimenti verrà ignorato e l'esecuzione prosegue con i comandi successivi.

La sintassi è la seguente:

```
if condizione:
```

```
    corpo da eseguire se la condizione è verificata
```


else

L'else è il costrutto che accompagna l'if, e il codice che racchiude viene eseguito nel caso le condizioni verificate precedentemente non siano True.

Ecco la sintassi.

```
if condizione:
```

```
    codice da eseguire se condizione è True
```

```
else:
```

```
    codice da eseguire se condizione è False
```



elif

Il costrutto if può avere anche una seconda (o molteplici) condizione da verificare se la prima non è valutata a True. Si possono inserire tanti elif dopo l'if quante sono le condizioni aggiuntive da verificare. Alla prima condizione verificata a True, l'esecuzione riprende con il codice che si trova in seguito al blocco if-elif-else. La sintassi è

```
if condizione:
```

```
    codice da eseguire se condizione è True
```

```
elif condizione2:
```

```
    codice da eseguire se condizione2 è True
```

```
# altri elif se necessario
```

```
else:
```

```
    codice da eseguire se nessuna condizione precedente è True
```

```
try:  
    # codice a rischio  
except:  
    # gestiamo l'errore
```

```
with open(filename, mode) as nome_file:  
    # leggere, scrivere il file rinominato nome_file
```


Dizionari

I dizionari sono delle strutture dati che contengono coppie chiave-valore. È quindi possibile identificare ciascun elemento attraverso una chiave, che deve essere univoca. La chiave può assumere quei tipi di dato che sono immutabili, tra cui intero, stringa, tupla, ecc... Il valore invece può assumere qualsiasi tipo, mutabile e immutabile.

Ecco la sintassi per definire un dizionario:

```
dizionario = {chiave: valore, chiave2: valore2}
```