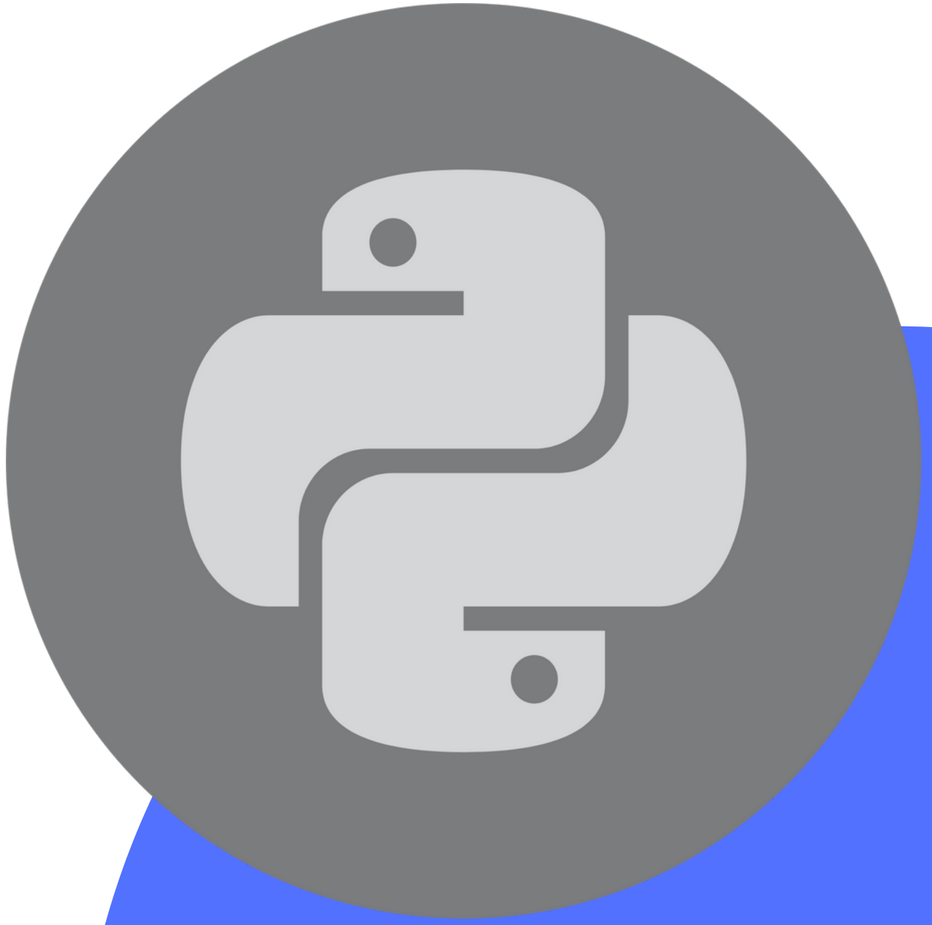
PYTHON COURSE

ENTRY LEVEL



Basics of programming in Python 3.10

This course will cover part of the arguments found in PCEP™ – Certified Entry-Level Python Programmer Certification

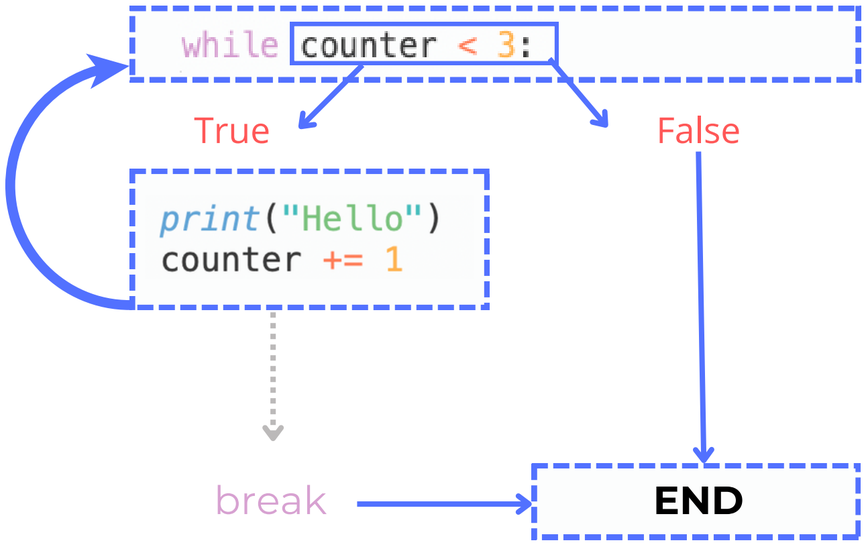
CONTROL FLOW: CONDITIONAL BLOCKS AND LOOPS

LOOPS: WHILE-ELSE

Used to handle the False branch of the program flow in while loops

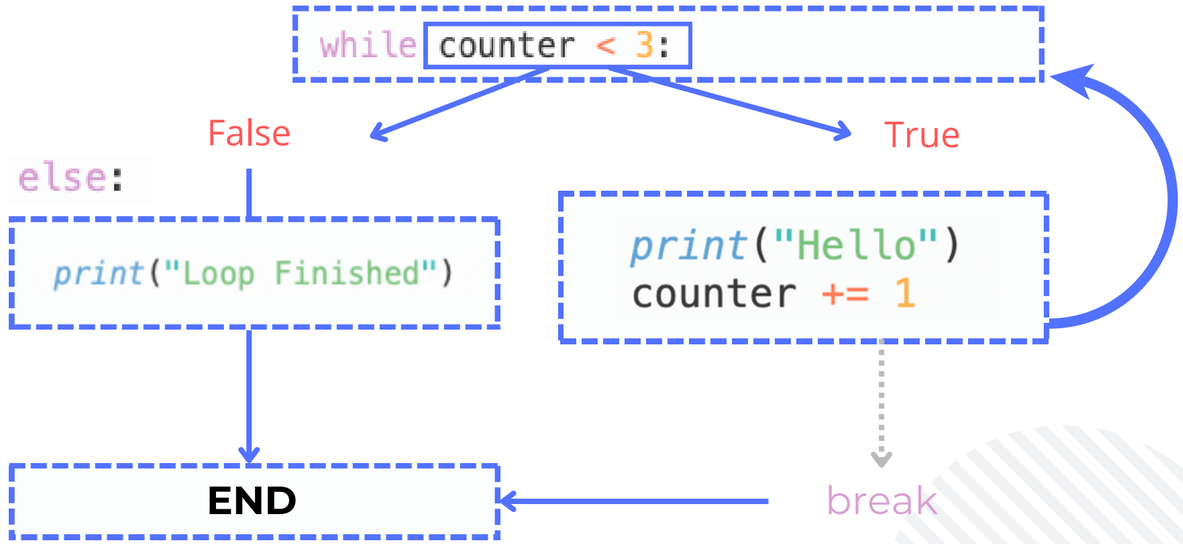
```
counter = 0
```

WHILE LOOP PROGRAM FLOW




```
counter = 0
```

WHILE ELSE LOOP PROGRAM FLOW



LOOPS: WHILE-ELSE

Only way to not execute the else block is exiting loop from break

```
counter = 0

while counter <= 3:
    print(counter)
    counter += 1
    if counter == 4:
        print("exiting loop from break")
        break
else:
    print("Condition False")
```

break is **executed before the programs can check the condition one last time**, so the else block is skipped

```
> python3 while_else.py
0
1
2
3
exiting loop from break
```

LOOPS: WHILE-ELSE

when tested contition becomes false, the loop is terminated and the else block executed

```
counter = 0

while counter <= 3:
    print(counter)
    counter += 1
else:
    print(f"Loop exited, counter is {counter}, and is higher than 3")
```

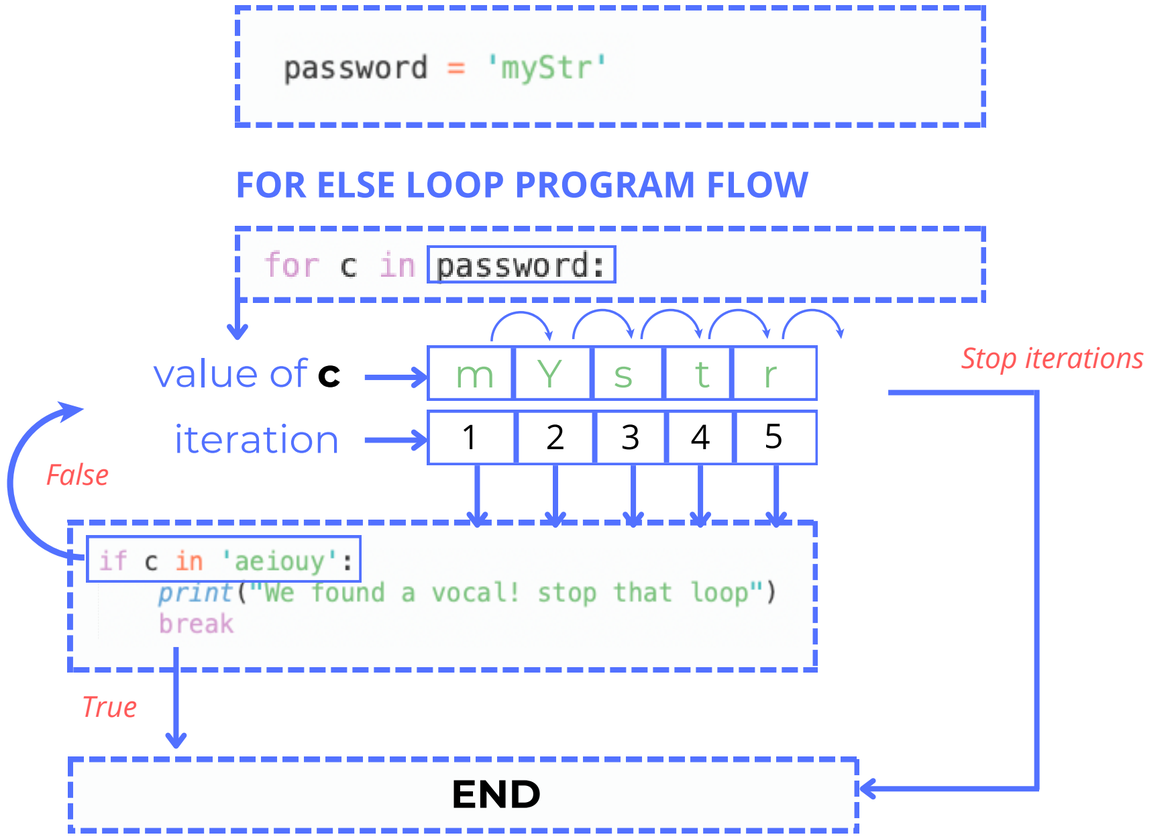
no *break* applied so loop is exited because this condition becomes *False*. The *else* block is executed

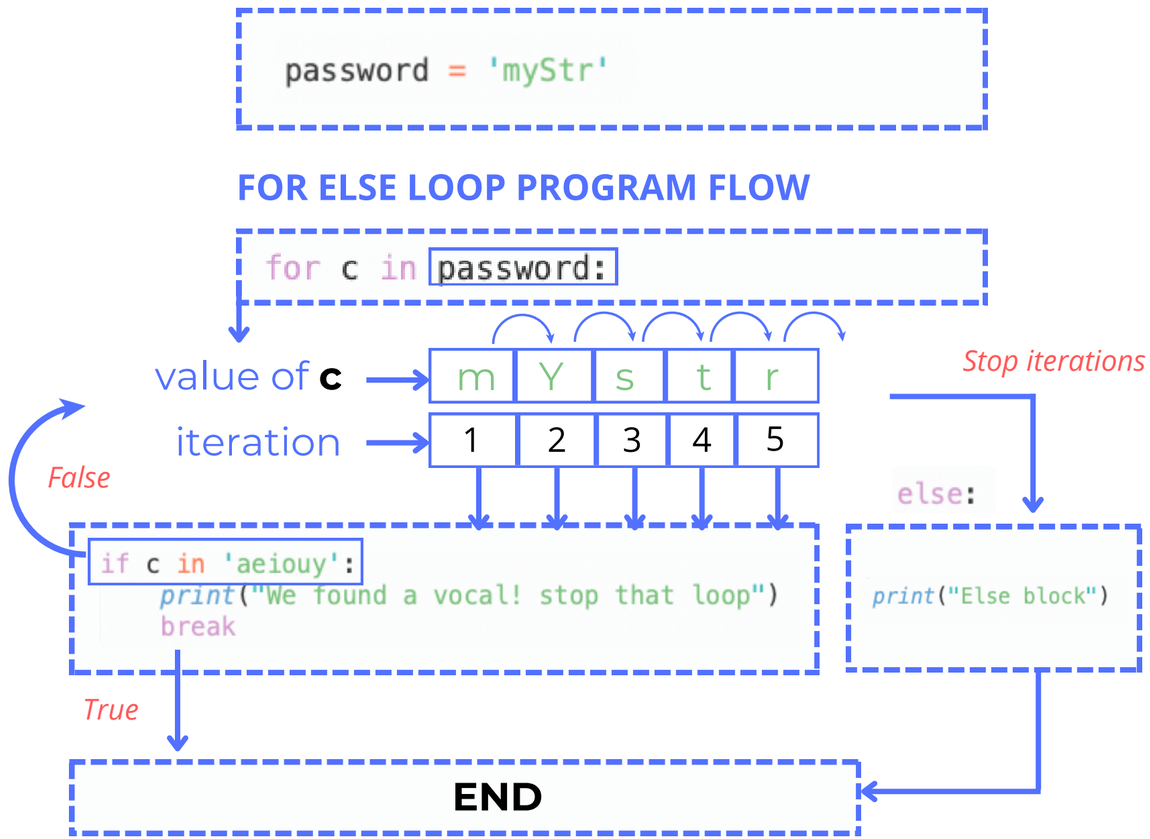
```
> python3 while_else.py
0
1
2
3
Loop exited, counter is 4, and is higher than 3
```

PYTHON

LOOPS: FOR-ELSE

Used to handle the False branch of the program flow in while loops





LOOPS: FOR-ELSE

```
password = 'myStr'

for c in password:
    if c in 'aeiouy':
        print("We found a vocal! stop that loop")
        break
else:
    print("Sorry man, we found no vocals in your pass")
```

True for **c = ‘y’ at second iteration**

exiting with **break** instruction

```
> python3 for_else.py
We found a vocal! stop that loop
```

LOOPS: FOR-ELSE

```
password = 'mnStr'

for c in password:
    if c in 'aeiouy':
        print("We found a vocal! stop that loop")
        break
else:
    print("Sorry man, we found no vocals in your pass")
```

False for every value of c

exiting for **terminating iterations**, executing *else* block

```
> python3 for_else.py
Sorry man, we found no vocals in your pass
```

QUESTION TIME

PYTHON

QUESTION 1:

What will be the value of the i variable when the while else loop finishes its execution?

```
i = 0
while i != 0:
    i = i-1
else:
    i = i+1
```

- 1
- 0
- 2
- the variable becomes unavailable

PYTHON

QUESTION 2:

What is the last value of variable i to be printed:

```
for i in range(10):
    print(i)
```

- 11
- 9
- 10
- variable becomes unavailable

PYTHON

QUESTION 3:

What is the last value of variable i to be printed:

```
i = 5
while i>0:
    i=i//2
    if i % 2=0:
        break
else:
    i+=1
print(i)
```

- 3
- code is erroneous
- 7
- 15

PYTHON

QUESTION 4:

How many lines does the following code produces as output?:

```
for i in range (1, 3):
    print("*", end= "")
else:
    print("*")
```

- three
- one
- two
- four

PYTHON

QUESTION 5:

What is the expected output of the following code:

```
a=2
if a>0:
    a+=1
else:
    a-=1
print(a)
```

- 3
- 1
- 2
- Erroneous code

PYTHON

QUESTION 6:

Assuming that my_var is a six or more letters long string, the following slice is shorter than the original string by how many characters?

```
my_var[1:-2]
```

- four chars
- three chars
- one char
- two chars

EXERCISES