COMP2152 LAB MANUAL

OPEN SOURCE DEVELOPMENT

This booklet will help the reader understand the concepts, principles, and implementation of the Python programming language. By the end of the booklet, the reader will be able to code comfortably in Python.

Prepared by Ben Blanc

TABLE OF CONTENTS

CHAPTER 3

ITERATION

FOR LOOP

The structure for the for loop is the following:

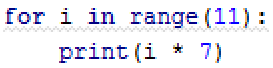
for variable_identifier in collection_of_data :

statement(s)

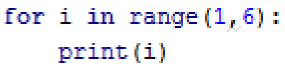
The variable_indentifier is a variable that will represent each individual value in your collection_of_data.

Take the following examples

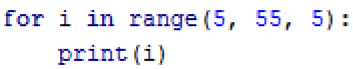
Here are some examples of a for loop:



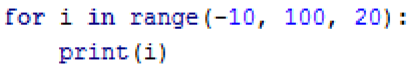
```
for i in range(11):
    print(i * 7)
```



```
for i in range(1,6):
    print(i)
```



```
for i in range(5, 55, 5):
    print(i)
```



```
for i in range(-10, 100, 20):
    print(i)
```

The **range()** function takes three paramters:

range(start, end (exclusive), incrimental_value).

It can also take one argument

range(end (exclusive)

The starting number will be 0 and the incremental value will be 1.

Or two values

range(start, end (exclusive)).

The incremental value will be 1.

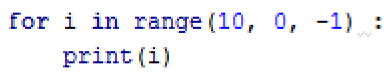
CHAPTER 3 AT A GLANCE

In this chapter you will learn how to code repeated instructions. You will be introduced to the following loops:

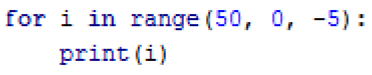
- for loop
- while loop

You will also learn how to code conditions and update values in these iteration structures.

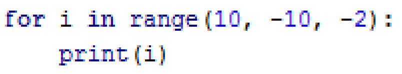
We could also decrement in our loops



```
for i in range(10, 0, -1) :
    print(i)
```



```
for i in range(50, 0, -5):
    print(i)
```



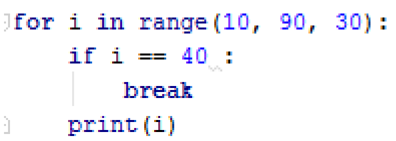
```
for i in range(10, -10, -2):
    print(i)
```

BREAK AND CONTINUE

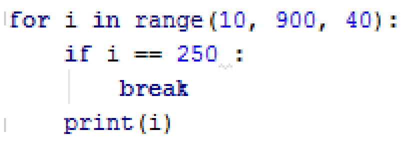
In an iteration structure, you can use the keywords **break** and **continue** to control the iteration structure.

BREAK STATEMENT

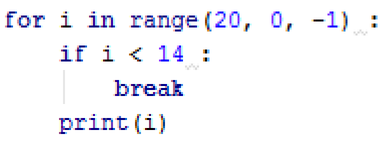
The break statement in the body of a loop immediately escapes the loop.



```
for i in range(10, 90, 30):
    if i == 40 :
        break
    print(i)
```



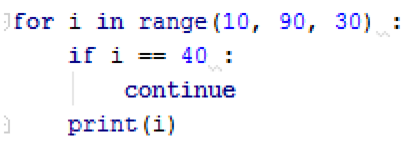
```
for i in range(10, 900, 40):
    if i == 250 :
        break
    print(i)
```



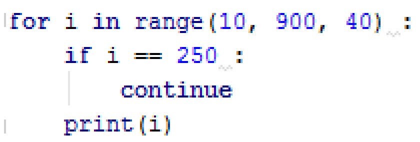
```
for i in range(20, 0, -1) :
    if i < 14 :
        break
    print(i)
```

CONTINUE STATEMENT

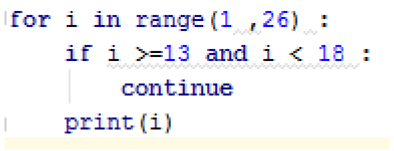
The continue statement in the body of a loop immediately stops processing the loop (at that point) and jumps to the next iteration.



```
for i in range(10, 90, 30) :
    if i == 40 :
        continue
    print(i)
```



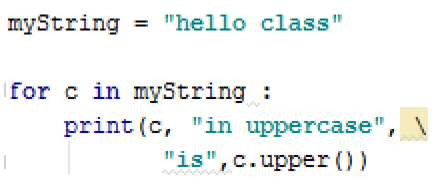
```
for i in range(10, 900, 40) :
    if i == 250 :
        continue
    print(i)
```



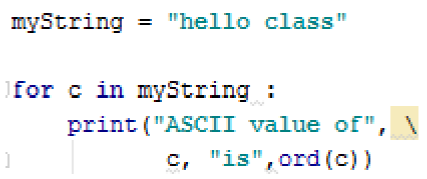
```
for i in range(1 ,26) :
    if i >=13 and i < 18 :
        continue
    print(i)
```

ITERATING A COLLECTION

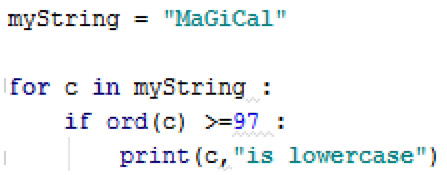
A string can be considered as a collection or characters. Therefore, the following is possible;



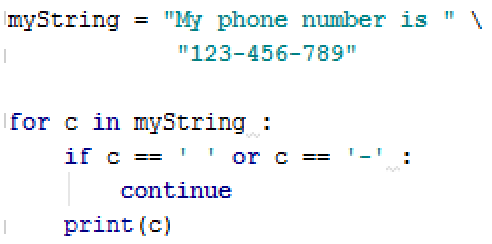
```
myString = "hello class"

for c in myString :
    print(c, "in uppercase", \
          "is",c.upper())
```



```
myString = "hello class"

for c in myString :
    print("ASCII value of", \
          c, "is",ord(c))
```



```
myString = "MaGiCal"

for c in myString :
    if ord(c) >=97 :
        print(c,"is lowercase")
```



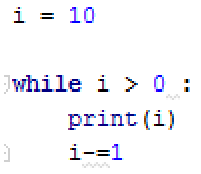
```
myString = "My phone number is " \
           "123-456-789"

for c in myString :
    if c == ' ' or c == '-' :
        continue
    print(c)
```

WHILE LOOP

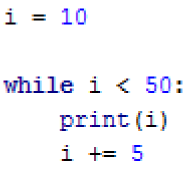
The structure for the while loop is the following:

while(conditional_expression)
{
    statement(s)
}

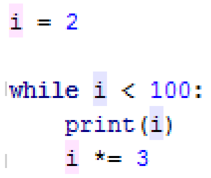
As you can see, the initialization statement and update statement is not included in the structure. It is up to the programmer to input them. You input the initialization statement before the while structure and you put the update statement within the loop body.



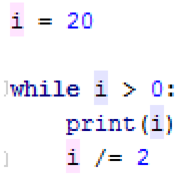
```
i = 10

while i > 0 :
    print(i)
    i-=1
```



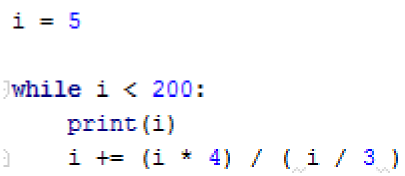
```
i = 10

while i < 50:
    print(i)
    i += 5
```



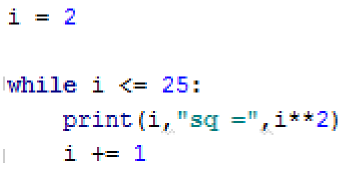
```
i = 2

while i < 100:
    print(i)
    i *= 3
```



```
i = 20

while i > 0:
    print(i)
    i /= 2
```



```
i = 5

while i < 200:
    print(i)
    i += (i * 4) / ( i / 3 )
```



```
i = 2

while i <= 25:
    print(i,"sq =",i**2)
    i += 1
```