SE Bootcamp

Hyperiondev

For Loops

Welcome

Your Lecturer for This Session

Yolandi Viljoen

Objectives

- Learn how to use the for loop to repeat a block of code a specified amount of times.

Loops

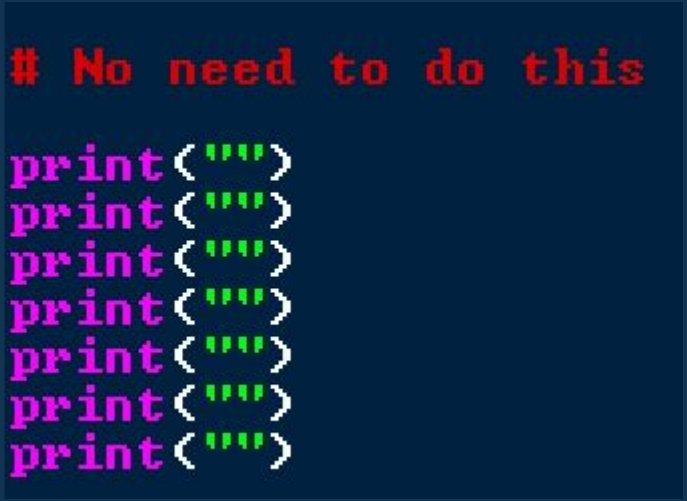
- Loops are used when we need to repeat a certain block of code multiple times.
- There are two types of loops that will be introduced:
  - **while** loops
  - **for** loops

while Loops

- ★ While loops are used in situations when we are not sure how many times we need to repeat the code block.
- ★ Therefore, we can use a while loop to execute a certain condition. While our condition is True, the code within the loop will execute, however, the loop will terminate the moment our condition becomes False.

for Loops

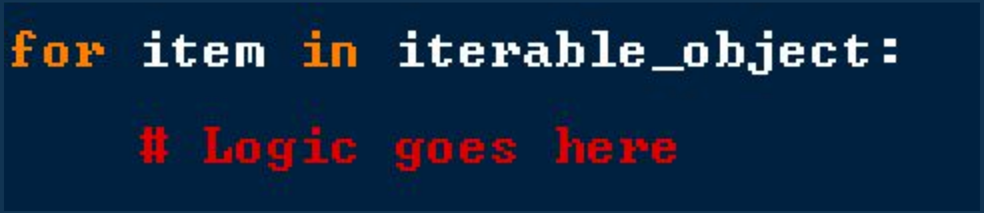
- ★ For loops are used when we need code to run a specified amount of times.
- ★ Think of it making the task of creating ten print statements much easier.



```
# No need to do this

print("")
print("")
print("")
print("")
print("")
print("")
print("")
```

for Loop Syntax

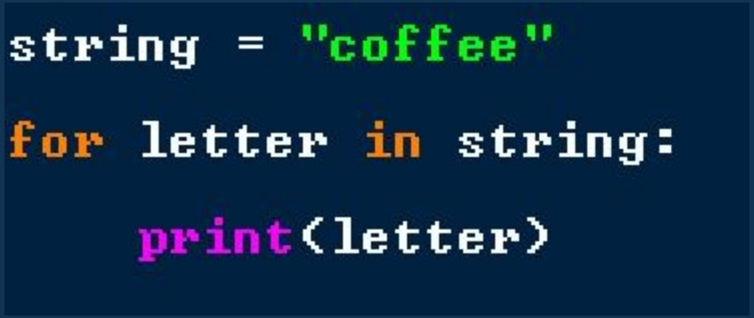


```
for item in iterable_object:
    # Logic goes here
```

★ Iterable_object: a list of numbers, a string of characters, a range etc.

★ Item: temporary variable used inside the for loop to reference the current position of our iterator.

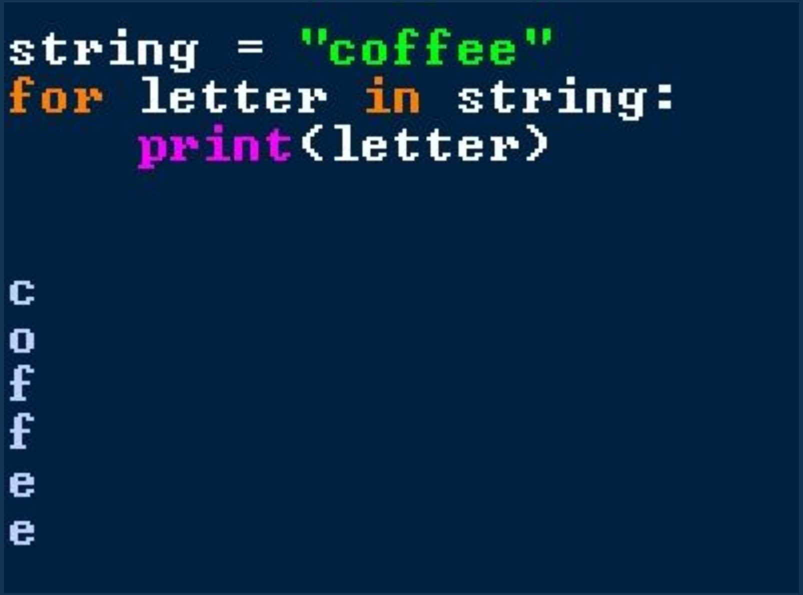
for Loop Example



```
string = "coffee"
for letter in string:
    print(letter)
```

- ★ The above loop will iterate over the string “coffee”.
- ★ This entails that the temporary variable letter will continuously be updated with each letter found in “coffee”.
- ★ Which results in the following output:

for Loop Example Cont.



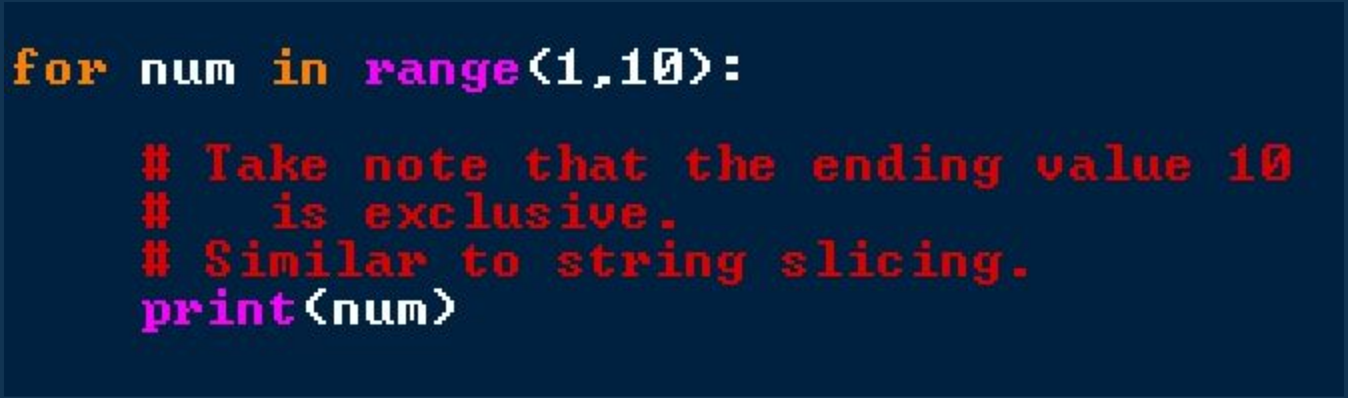
```
string = "coffee"
for letter in string:
    print(letter)
```

```
c
o
f
f
e
e
```

Since letter will iterate over every instance of string, we get the output of “coffee” spelt on separate lines.

for Loops and Range

★ With for loops we can also get a range of numbers from a starting value to an ending value.



```
for num in range(1,10):

    # Take note that the ending value 10
    #    is exclusive.
    # Similar to string slicing.
    print(num)
```

The output here will be all values from 1 to 9.

Range

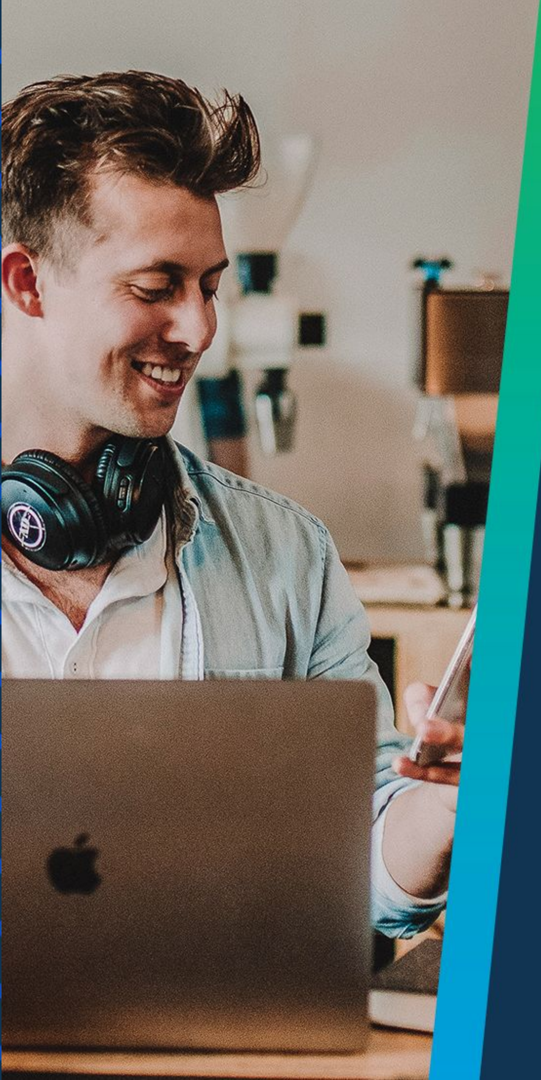
- ★ Range allows us to run a block of code a specified amount of times.

Range	Description	Additional Info
range(10)	Outputs integers from 0 through 9	Range will always start from 0
range(1, 10)	Outputs integers from 1 to 9	Parameters(start, end)
range(1, 10, 2)	Outputs odd numbers from 1 to 10	Third available parameter is “step” (how many to skip)
range(10, 0, -1)	Outputs integers from 10 to 1	Negative counter that skips backwards

Hyperiondev

Q & A Section

Please use this time to ask any questions relating to the topic, should you have any.



Hyperiondev

Thank You for Joining Us