

Iterations and More Control Flow

IT Adventures: Smart IT

Reading Assignment

- [How to Think Like a Computer Scientist: Interactive Edition](#)
- Required: 4.1 - 4.5
- Recommended: 4.1 - 4.9

What is a Loop?

- A loop is a concept in programming that allows us to run a command, or a set of commands, multiple times to complete a goal.
- In python there are 2 kinds of loops, the for loop, and the while loop
- A loop is made up of 2 components, an initialization statement that sets the iteration value and the termination condition of the loop, and the instructions that will be repeated.
- Conditionals are always checked at the beginning of the loop.
- The iteration value differs between the two types of loops and are best explained on their own.

The For Loop

- A for loop is a loop that will run a number of times until the iteration value goes over the range dictated in the termination statement.
- The initialization statement will take the form of
 - for [variable] in [condition]:
 - [variable] will be the iteration variable, the iteration variable will take on the value of the condition variable corresponding to what iteration the loop is in
 - [condition] can take the form of an expression using the range() function, or a list
 - The line must end in a colon
 - Finally, the instructions that will be repeated during iterations of the loop must be indented
- Used when you want a known number of iterations

How to use Range

- The range() function will return a sequence of number in accordance with the arguments it's given
- range(x) where x is an integer will return a list of 0 to x-1
 - range(4) = 0,1,2,3
- range(x,y) where x > y will return a list of x to y-1
 - range(2,5) = 2,3,4
 - Cases where x<=y will return nothing
- range(x,y,z) will return a list of x to y-1 with an interval of z
 - range(2,9,2) = 2,4,6,8
 - Can also use this to iterate backwards

Examples

script.py

```
1 # Prints out the numbers 0,1,2,3,4
2 for x in range(5):
3     print(x)
4
5 # Prints out 3,4,5
6 for x in range(3, 6):
7     print(x)
8
9 # Prints out 3,5,7
10 for x in range(3, 8, 2):
11     print(x)
```

Run



IPython Shell



```
0
1
2
3
4
3
4
5
3
5
7
```

In [1]: |

The While Loop

- A while loop is a loop that will iterate through its instructions until its iteration conditional is made false.
- The conditional can be any boolean statement, and so long as it remains true the loop will continue to iterate.
- The initialization statement will take the form of
 - while [conditional]:
 - [conditional] being literally any boolean statement
 - Finally, as with the for loop, the instructions that will be repeated during iterations of the loop must be indented
- What would happen if the initialization condition was
 - while true:
 - This is called an infinite loop, you would require a break statement to exit it
 - They can cause memory leaks so be careful
- Used when you don't know exactly how many time the loop will run, like with programs that run until closed by the user.

Examples

script.py

```
1 # Prints out 0,1,2,3,4
2
3 count = 0
4 while count < 5:
5     print(count)
6     count += 1 # This is the same as count = count + 1
```

IPython Shell



```
0
1
2
3
4
```

In [1]: |

Run



Break And Continue

- break is used to exit any loop from exactly where the break statement is called regardless of the status of the conditional.
- continue is used to immediately move from one iteration of the loop to the next without finishing the rest of the instructions

Examples

script.py

```
1 # Prints out 0,1,2,3,4
2
3 count = 0
4 while True:
5     print(count)
6     count += 1
7     if count >= 5:
8         break
9
10 # Prints out only odd numbers - 1,3,5,7,9
11 for x in range(10):
12     # Check if x is even
13     if x % 2 == 0:
14         continue
15     print(x)
```

IPython Shell



```
0
1
2
3
4
1
3
5
7
9
```

In [1]: |

Run



Technical Challenge

Expand even further upon your calculator program and make it work for multiple numbers and operations!

Your program should:

- Continue to ask for an operation, numbers, and print the result until “end” is entered when asked for an operation.
- Use a loop to make sure that the operation entered is valid (if the operation is not valid, notify the user and ask again)
- BONUS: Use a loop to make sure that the numbers entered are valid (similarly to above)
- Watch the YouTube link to see a step-by-step solution
 - <https://youtu.be/JS9YoIOZTWE>

Example Input and Output

```
User Input
Program Output

What operation do you want to do? (Add/Subtract/Multiply/Divide)
Operation: Add
Number 1: 10
Number 2: 5
10 + 5 = 15

What operation do you want to do? (Add/Subtract/Multiply/Divide)
Operation: Divide
Number 1: 10
Number 2: 5
10 / 5 = 2

What operation do you want to do? (Add/Subtract/Multiply/Divide)
Operation: Muultiply
Sorry, that is not a correct operation.

What operation do you want to do? (Add/Subtract/Multiply/Divide)
Operation: Multiply
Number 1: 10
Number 2: 5
10 * 5 = 50

What operation do you want to do? (Add/Subtract/Multiply/Divide)
Operation: end
Goodbye!
```