

04_Iteration

January 19, 2018

1 Iteration

1.0.1 The While Loop

The while loop is used to repeating identical or similar tasks. The following chunk of code is a program that repeats "Hello World" five times.

```
In [1]: print("Hello World!")
        print("Hello World!")
        print("Hello World!")
        print("Hello World!")
        print("Hello World!")
```

```
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
```

Since the above task was repetitive, we can use a while loop to do it:

```
In [2]: count = 1 #initialize the count to 1

        while count <=5 :
            print("Hello World!")
            count = count + 1 #increase the count by 1
```

```
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
```

The above loop execution will have 5 iterations and with each iteration, the execution works as follows:

1. Evaluates the condition (count <=5), yielding True or False

2. If the condition (count <=5) is false, Python will exit the while loop, and continues execution at the next statement
3. If the condition (count <=5) is true, execute the statements in the loop and then go back up to the beginning of the while statement.

Notice that the "count" variable is there to prevent the loop from running indefinitely (i.e. infinite loop). It is initialized to 1, and with each iteration, it is incremented by 1 using the statement (count = count + 1). It will keep incrementing until it reaches 6 in this case, and thus it will have the condition (count <=5) be false, so the loop will stop.

Write a program that counts down from five and then says "Hello World" once it is done from the countdown.

```
In [3]: count = 5 # start at 5
```

```
while count >=1:
    print (count)
    count = count - 1

print("Hello World!")
```

```
5
4
3
2
1
Hello World!
```

1.0.2 Infinite Loops and the break Statement

The break statement is used to jump out of a loop.

Write a program that keeps taking input from a user, prints it, until the user inputs the magic input "MSBA". Once the user input "MSBA", then the program prints out "You are a winner!", and stops.

```
In [4]: while True:
        line = input("> ")
        if line == "MSBA":
            print("You are a winner!")
            break
        print(line)
```

```
> Hello!
Hello!
> Canada
Canada
> MSBA
You are a winner!
```

1.0.3 Finishing Iterations with `continue`

Sometimes you are in an iteration of a loop and want to finish the current iteration and immediately jump to the next iteration. In that case you can use the `continue` statement to skip the next iteration without finishing the body of the loop for the current statement.

Update the previous code so that the program keeps taking input from the user until they get the input "MSBA", however, it treats lines that start with the hash character (#) as lines not to be printed (kind of like Python comments)

```
In [5]: while True:
        line = input("> ")
        if line == "MSBA":
            print("You are a winner!")
            break
        if line[0] == "#":
            continue
        print(line)
```

```
> Hello!
Hello!
> # I am a comment.. don't print me
> Canda
Canda
> MSBA
You are a winner!
```

1.0.4 Using `for` Loops

The following `for` loop prints "Hello World!" 5 times. The `range()` function creates number of integers (whole numbers) to generate, starting from zero. eg. `range(3) == [0, 1, 2]`.

```
In [6]: for count in range(5):
        print("Hello, World!")
```

```
Hello, World!
Hello, World!
Hello, World!
Hello, World!
Hello, World!
```

The following chunk of code wishes happy new year to 3 of my friends:

```
In [7]: friends = ["Hilary", "Robertas", "Lily"] #list

        for name in friends: #looping through the list of 3 people.
            print("Happy New Year,", name)
```

Happy New Year, Hilary
Happy New Year, Robertas
Happy New Year, Lily

Write a function `repeat(count)`, that takes an argument `count` from the user and prints "Hello, World" `count` times.

```
In [8]: def repeat(count):  
        for i in range(count):  
            print("Hello, world!")
```

```
In [9]: count = int(input())  
        repeat(count)
```

```
4  
Hello, world!  
Hello, world!  
Hello, world!  
Hello, world!
```

Write a program that takes numeric input from the user, and prints the sum of all previous input until the user input is 999, then the program stops. e.g. >1

```
1  
>2  
3  
>4  
7  
>999
```

```
In [10]: sum = 0  
        while True:  
            value = float(input())  
            if value == 999:  
                break  
            sum = sum + value  
            print(sum)
```

```
5  
5.0  
4  
9.0  
6  
15.0  
999
```

Write a program that returns the minimum of a given list of numbers. Don't use Python's built-in functions.

```
In [11]: mylist = [4, 6, 2, 90, 19]
```

```
    smallest = mylist[0] # initialize the minimum to be the first element in the list for

    for num in mylist:
        if num < smallest:
            smallest = num

    print("The minimum is: ", smallest)
```

```
The minimum is:  2
```