

Introduction to Machine Learning Applications

Spring 2021

Lecture-3

Lydia Manikonda

manikl@rpi.edu



Rensselaer

Today's agenda

- Python Basics – Recap and Loops, Conditionals, Functions
- Including class exercises

Recap on Basics

- Python variables
- Data structures
 - List
 - Dictionary
 - Tuple
 - Set
- Mini class exercise

Mini Quiz

1. Suppose list1 is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after list1.pop(1)?
2. Given a string (Example: "machinElearning") count the number of vowels present in the string.

Python fundamentals

Loops, conditionals, functions

Loops

Loops in Python

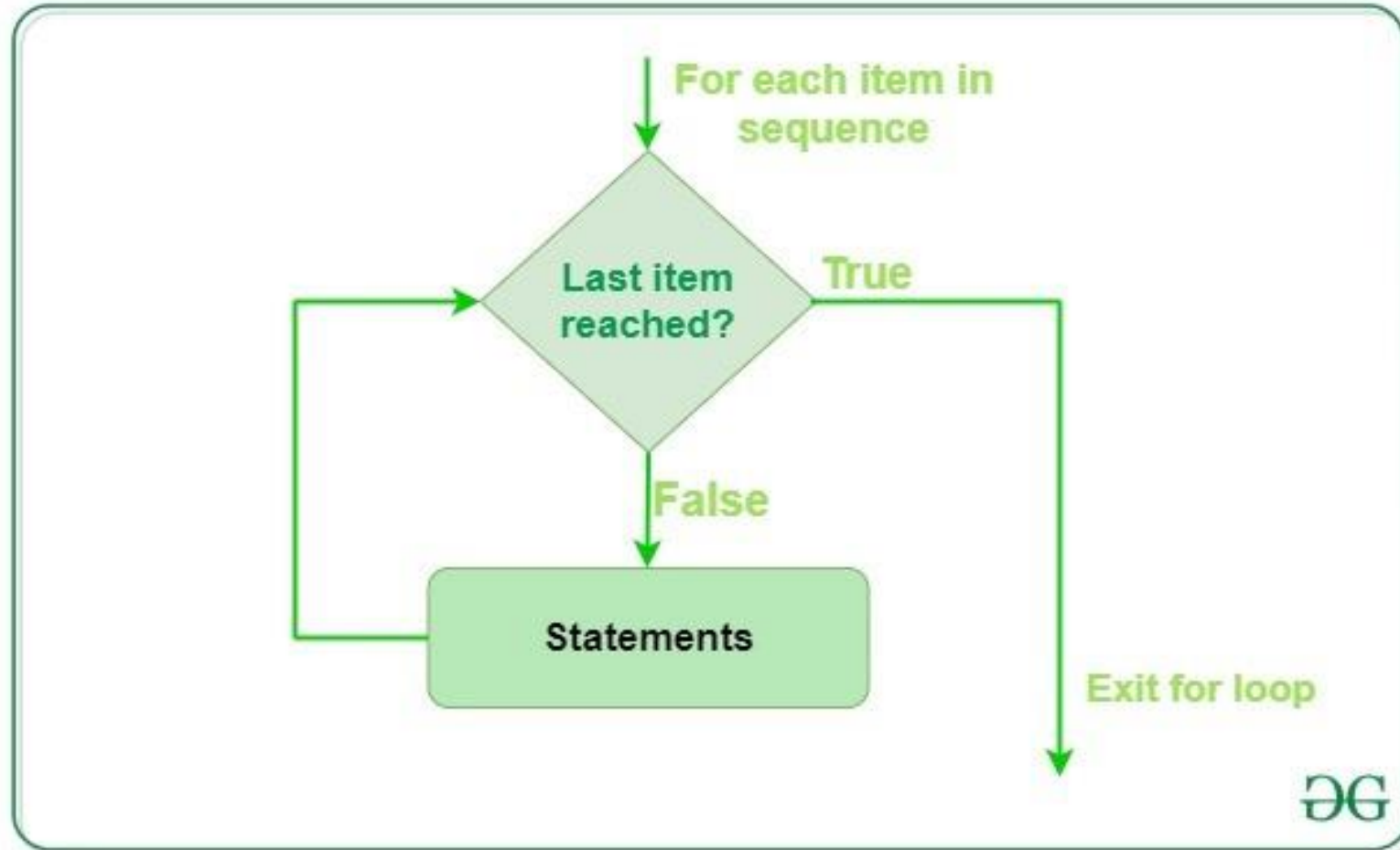
For

```
for iterator_var in sequence:  
    statements(s)
```

While

```
while expression:  
    statement(s)
```

for



for

```
>> print("List Iteration")
>> list1 = ["hello", "world"]
>> for i in list1:
    print(i)

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

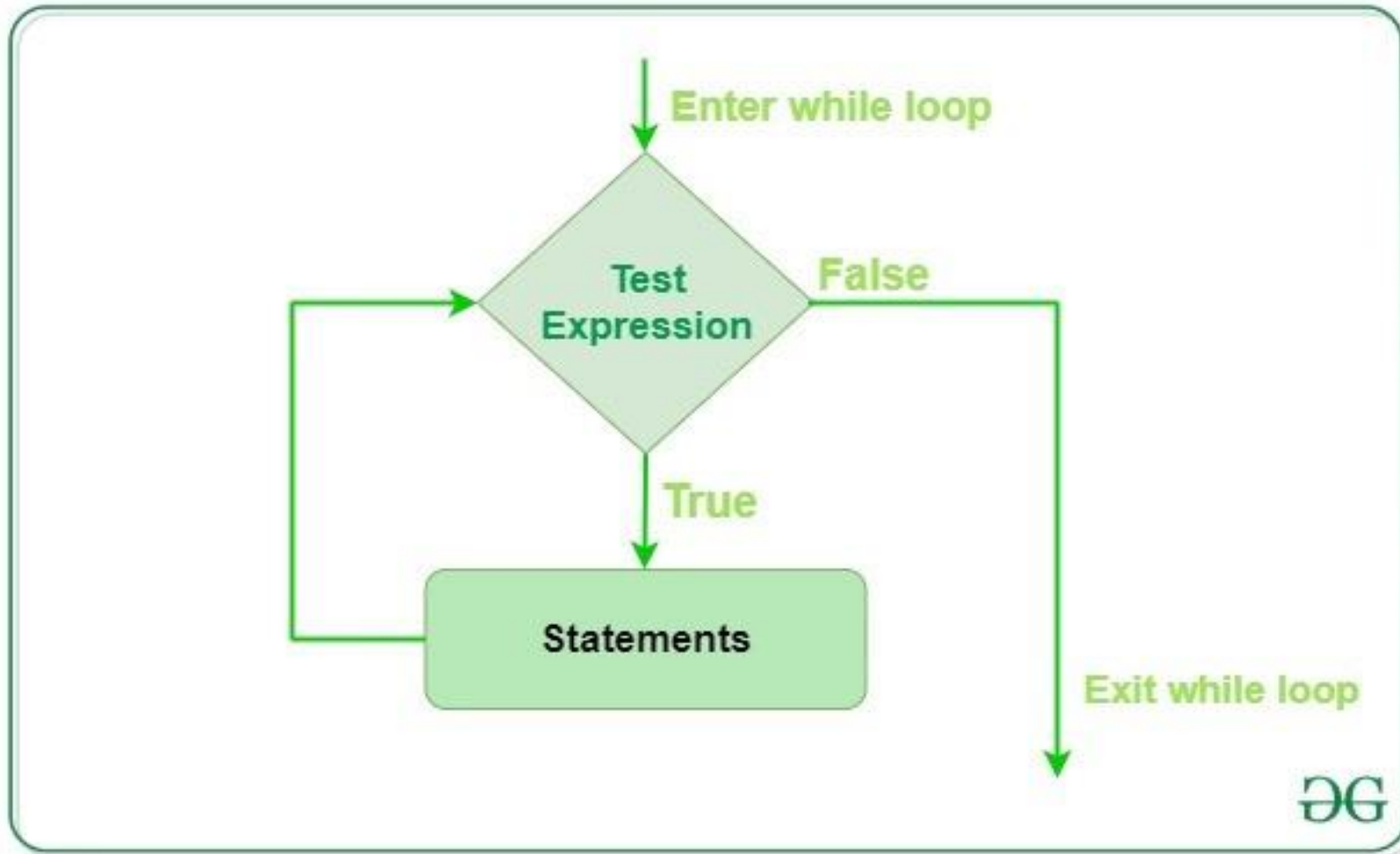
>> for letter in 'machinelearning':
    if letter == 'e' or letter == 's':
        continue
    print('Current Letter :', letter)
```

3. for loop -- Example

Using the *for* loop print a new list as an output with all the squares of the elements in a given list [1, 2, 3, 4, 5].

Output: [1, 4, 9, 16, 25]

while



while

```
>> count = 0
```

```
>> while (count < 3):
```

```
    count = count + 1
```

```
    print("Hello world!")
```

While

```
>> i = 0
```

```
>> a = 'machinelearning'
```

```
>> while i < len(a):
```

```
    if a[i] == 'e' or a[i] == 's':
```

```
        i += 1
```

```
        continue
```

```
    print('Current Letter :', a[i])
```

```
    i += 1
```

4. while loop – Example

Write a program to print this format when given a string str1="machine" **using while loop**:

machine

machin

machi

mach

mac

ma

m

Conditionals

If condition

```
if condition:
```

```
    statement1
```

```
    statement2
```

```
# Statements to execute if condition is true
```

```
if condition:
```

```
    statement1
```

```
statement2
```

```
# Here if the condition is true, if block will consider only statement1
```


Example

```
i = 10
```

```
if (i > 15):
```

```
    print ("10 is less than 15")
```

```
print ("I am Not in if")
```

If-else

if (condition):

 # Executes this block if condition is true

else:

 # Executes this block if condition is false

Example

```
i = 20
```

```
if (i < 15):
```

```
    print ("i is smaller than 15")
```

```
    print ("i'm in if Block")
```

```
else:
```

```
    print ("i is greater than 15")
```

```
    print ("i'm in else Block")
```

```
print ("i'm not in if and not in else Block")
```

If-elif-else

```
if (condition):  
    statement  
elif (condition):  
    statement  
..  
else:  
    statement
```

if-else-if

```
>> num1 = 4
```

```
>> if(num1%2 == 0):
```

```
    print("Num1 is even")
```

```
>> elif(num1%2==1):
```

```
    print("Num1 is odd")
```

```
>> else:
```

```
    print("It never comes to this section")
```

5. Class Exercise

Given a list `l1= [1,2,3,4,5,6,7,8,9,10]`, print the maximum value till a given index (from the start of the list) using a *for* loop and *if* statement. (Assume that all the values in this list are positive integers)

Output: `[1,2,3,4,5,6,7,8,9,10]`

=====

Input: `[10,9,8,7,6,5,4,3,2,1]`

Output: `[10, 10, 10, 10, 10, 10, 10, 10, 10, 10]`

Functions

Functions

- Set of statements that take inputs and perform certain computations

```
>> def FindEven( x ):
    if (x % 2 == 0):
        print "even"
    else:
        print "odd"
```

```
>> FindEven (2)
```

```
>> FindEven (3)
```


Pass by Reference

- When we pass a reference and change the received reference to something else, the connection between passed and received parameter is broken.

Pass by reference and pass by value

- In Python it utilizes “pass by assignment”.
- Pass by value means the actual value is being passed to the function
- Pass by reference means the address of that argument is being passed to the function
- When you pass immutable objects as arguments to a function:
 - Such as integers, strings or tuples
 - These arguments will act as call by value as you cannot change the value of these immutable objects.
 - So when you change the value in the function and return to the location where you made the function call, the value will be same as before making the function call.

Example-2 this will act like pass by value

```
def myFunc(x):
```

```
    x = 20
```

```
x = 10
```

```
myFunc(x)
```

```
print(x)
```

Example-3 acts like a pass by value

```
def swap(x, y):  
    temp = x;  
    x = y;  
    y = temp;  
# Driver code  
x = 2  
y = 3  
swap(x, y)  
print(x)  
print(y)
```

Pass by reference and pass by value

- When you pass a mutable object such as list, it will be considered as pass by reference as long as the values are altered inside the function and they will be reflected even after exiting the function
- However, even the mutable object if you rebind the reference which means assigning the entire list to a new set of values as a list the pass by reference will be broken.

Functions Example – acts as pass by reference

```
def myFun(x):  
    x[0] = 20
```

```
lst = [10, 11, 12, 13, 14, 15]  
myFun(lst)  
print(lst)
```

Example-1 acts as pass by reference but.. Its broken due to rebinding the variable again

```
def myFun(x):
```

```
    x = [20, 30, 40]
```

```
lst = [10, 11, 12, 13, 14, 15]
```

```
myFun(lst)
```

```
print(lst)
```

Default Arguments

```
def myFun(x, y=50):  
    print("x: ", x)  
    print("y: ", y)
```

```
myFun(10)
```


Keyword arguments

```
def student(firstname, lastname):  
    print(firstname, lastname)
```

```
student(firstname = 'John', lastname = 'Smith')
```

```
student(lastname = 'Smith', firstname = 'John')
```

Variable length Arguments

```
def myFun(*argv):  
    for arg in argv:  
        print (arg)
```

```
myFun('Hello', 'Welcome', 'to', 'GeeksforGeeks')
```

Lambda Functions – Anonymous functions

- ***lambda arguments: expression***

```
>> def square(y):
```

```
    return y**2
```

```
>> g = lambda x: x**2
```

```
>> print(g(7))
```

```
>> print(cube(5))
```

Example – Intersection of 2 lists

```
>> def ArrIntersect(a1, a2):  
    result = list(filter(lambda x: x in a1, a2))  
    print ("Intersection : ",result)
```

```
>> arr1 = [1, 3, 4, 5, 7]
```

```
>> arr2 = [2, 3, 5, 6]
```

```
>> ArrIntersect (arr1,arr2)
```

Class exercises

- Python notebook

Functions examples

6. Write a function Square that takes an integer argument and outputs the square value of this argument. For example, if the input is 3, output should be 9.
7.

```
y = 8  
z = lambda x : x * y  
print z(6)
```

Exercise

8. Given a list of keywords, create a dictionary:
Keywords as the keys and their frequencies as the values
Remove all the keys that appear only once.

Input: Keywords = ['hello', 'I', 'am', 'fine', 'but', 'fine', 'is', 'fine', 'hello', 'to', 'you', 'fine']

Dictionary: {'hello': 2, 'I':1, 'am':1, 'fine':4, 'but':1, 'is':1, 'to':1, 'you':1 }

FinalDictionary: {'hello': 2, 'fine':4 }

del Dictionary[<key>]

```
keywords = ['hello', 'i', 'am', 'fine', 'but', 'fine', 'is', 'fine', 'hello', 'to', 'you', 'fine']
```

```
diction={}
```

```
for eachword in keywords:
```

```
    if eachword in diction.keys(): #try
```

```
        diction[eachword]+=1
```

```
    else: #except
```

```
        diction[eachword]=1
```

```
print(diction)
```

```
allkeys=list(diction.keys())
```

```
for eachkey in allkeys:
```

```
    if diction[eachkey]==1:
```

```
        del diction[eachkey]
```

```
print(diction)
```


Exercises

9. Write a comprehensive code to initialize a dictionary where values are squares of keys – keys from 1 to 10.
10. Write a lambda function to multiply two numbers X and Y.
11. Find all the duplicate characters in a given string “machinelearning”
12. Convert all the repeated characters in the above string to uppercase – “mAchiNEIEArNINg”
13. Write a program to generate the first set of ‘ n ’ values in the Fibonacci series 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ... using conditionals and while statement. For example, $n=3$ it will be 0, 1, 1