WATCH VIDEOS IN THE PORTAL. TITLE NAME OF VIDEO IS MENTIONED HERE.

Python basics

Watch Video 0: Importance of programming

Python is emerging as the popular language used more in data science applications.

Python is an interpreted, object-oriented, high-level programming language. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

Take the case of the tech giant Google that has created the deep learning framework called tensorflow – Python is the primary language used for creating this framework. Its footprint has continued to increase in the environment promoted by Netflix. Production engineers at Facebook and Khan Academy have for long been using it as a prominent language in their environment.

Python became the most popular language in the data science world. So what are you waiting for? Lets start learning python basic as your first assignment.

Now Lets get started.

Watch Video 1: Print

Watch Video 2: Comments

1. Print Function

In [ ]:
```
# Lets print one sentence("Hello World") using python
```

```
Hello World!!
```

Handling comments

- single line

In [ ]:
```
# This is comment.

print(type(i))
```

```
<class 'int'>
```

- Multi line

In [ ]:
```
'''This is multiline
comments.
lets print
same'''

print(type(i))
```

```
<class 'int'>
```

- There is also different way for printing sentence with variables . We will see going forward. Lets see how we can declare and handle variable.

2.Variable declaration

Watch Video 3: Variables

Watch Video 4: DataTypes

Watch Video 5: TypeCasting

Covers the topic of variable declaration and type of variables.

- reference:-

1+1+1+1+1 = 5 points

Python support different data types like `int` , `float` , `string` and `boolean` .

In [ ]:
```
# lets declare variables int , float and string

#int, # your code below
a=

#string
b=

#float
c=
```

Lets print type of a , b and c

In [ ]:
```
# print type of a
```

```
<class 'int'>
```

In [ ]:
```
# print type of b
```

```
<class 'str'>
```

In [ ]:
```
# print type of c
```

```
<class 'float'>
```

Lets try with a complete sentence

In [ ]:
```
# define a string with sentence  'This is so beautiful'
d=


# print its type
```

```
<class 'str'>
```

Lets try to declare boolean variable

In [ ]:
```
# declare d with True

E=
```

In [ ]:
```
# print type of E

print(type(E))
```

```
<class 'bool'>
```

Cool! so variable declaration and getting it's type was easy.

SWAPPING

Read below cell code for two methods of swapping

In [ ]:
```
'''
# Python program to demonstrate
# swapping of two variables

x = 10
y = 50

# Swapping of two variables
# Using third variable
temp = x
x = y
y = temp

print("Value of x:", x)
print("Value of y:", y)

'''
```

In [ ]:
```
'''
# Python program to demonstrate
# swapping of two variables


x = 10
y = 50
```

```
# Swapping of two variables
# without using third variable
x, y = y, x

print("Value of x:", x)
print("Value of y:", y)

'''
```

In [ ]:
```
# swap a and b
```

Lets confirm the value

In [ ]:
```
# print a and b
```

```
city 5
```

In [ ]:
```
# print with string a and b
```

```
a = city,b = 5
```

3. Arithmetic Operations

Watch Video 6: Arithmatic Operations

Hey novice! this will be easy for you until you know basic arithmetic symbols like +, -, /, * etc.

String formatting reference: https://www.geeksforgeeks.org/string-formatting-in-python/

1 + 7 = 8 points

In [ ]:
```
# declare two variables, a = 24, b = 3

# your code below
a =
b =

print('a = {}, b = {}'.format(a,b)) # this is way of string formatting
```

```
a = 24, b = 3
```

In [ ]:
```
# calculate and print all the possible numerical operations on a and b (There are total 7
```

```
a + b =  27
a - b =  21
a * b =  72
a / b =  8.0
a % b =  0
a // b =  8
a ** b =  13824
```

Watch Video 7: Indexing & Slicing

Watch Video 8: String Operations

4. String slicing

Hey coder, do you know? Python slicing is about obtaining a sub-string from the given string by slicing it respectively from start to end

So be ready to implement it today!

2 points

In [ ]:
```
#Create variable a = “pineapple”

a =
```

```
pineapple
```

In [ ]:
```
# Using index slicing, print the words ‘pine’ and “apple”.

# HINT - a[:]
```

```
pine
apple
```

Great job! Slicing is fun.

Watch Video 9: Input Output

Assign value to variable using user input

Reference for input() method: https://www.w3schools.com/python/ref_func_input.asp

In [ ]:
```
# take value from user and print it back

i=
```

```
2
```

In [ ]:
```
i
```

Out[ ]:
```
'2'
```

In [ ]:
```
#check type
```

```
<class 'str'>
```

In [ ]:
```
# take value from user, type cast it to int and print it back.

print("Please enter value: ")

i=

print("Hello , You have entered : ",i)
```

```
Please enter value:
40
Hello , You have entered :  40
```

In [ ]:

```
# check type
```

```
<class 'int'>
```

5. Math Functions

Watch Video 10: Math Function

Lets do some simple mathematical operations by using math module. Math module contains various mathematical functions. So here we go!

10 points

In [ ]:
```
#import math
```

In [ ]:
```
# declare a variable x = 2 and y=5.32167

x=
y=
```

In [ ]:
```
# Print the maximum value between x and y
```

Out[ ]:
```
5.32167
```

In [ ]:
```
# Print the minimum value between x and y
```

Out[ ]:
```
2
```

In [ ]:
```
# Print square and cube of x and store in a and b variable

a =
b =


# print a

# print b
```

```
4
8
```

In [ ]:
```
#print log of x and store it in variable c

c =

# print c
```

```
0.693
```

In [ ]:
```
#print minimum value between x,y,a, b and c
```

Out[ ]:
```
0.693
```

In [ ]:
```
#print maximum value between x,y,a, b and c
```

Out[ ]:
```
8
```

In [ ]:
```
#print the largest integer that is smaller than or equal to c. ie.round down
```

```
0
```

In [ ]:
```
#print the smallest integer that is greater than or equal to c ie. round up
```

```
1
```

6. Logical operations

You must be aware of some logical operations you did in college, like 'and', 'or', 'not' etc. These operations becomes even more easy to implement when you have friend like python.

1 + 1 + 1 + 6 = 9 points

Watch Video 11: Binary Operator(optional)

Watch Video 12: Python Operator

In [ ]:
```
# declare two variables, a = True, b = False

a =
b =

# print a and b
```

```
True
False
```

In [ ]:
```
# print type of a and type of b
```

```
<class 'bool'>
<class 'bool'>
```

A fun fact for you ! The Python Boolean type is one of Python's built-in data types. It's used to represent the truth value of an expression

In [ ]:
```
# print int(a) and int(b)
```

```
int of a is  1
int of b is  0
```

We are sure you must have seen that, values for 'True' is 1 and for 'False' is 0.

In [ ]:
```
# Find and print the values of not a, not b, a and b, a or b, a and not b
```

```
not a =  False
not b =  True
a and b =  False
a or b =  True
a and not b =  True
```

We hope you tried to undertand the above output!

6. Data Structures in Python

Python has implicit support for Data Structures which enable you to store and access data. These structures are `List` , `Dictionary` , `Tuple and Set` .

- For now , we will only look into List to get basic understanding. Later we will discuss about other data structure.

Watch Video 13: Data Structures Introduction

Watch Video 14.1,14.2,14.3: List

List

A list is collection of items which can be change/ modify at any point of time. You can create list that contains the string, float, integer and boolean. In python, square brackets ([]) represents lists.

In [ ]:
```
# list that contains numbers.

l =
```

```
[9, 1, 4, 5, 10]
```

In [ ]:
```
# print type of l
```

```
<class 'list'>
```

In [ ]:
```
# create list that contains one string, float and integer values.

l1 =
```

```
['Cat', 15, 2.97]
```

In [ ]:
```
# lets create list and try to access each element with indexing

l2 =

# print value at index 0


# print value at index 2


# print last item from the list


# print second last item from list using -2
```

```
11
43
65
40
```

In [ ]:
```
# print the length of list
```

```
5
```

In [ ]:
```
# sort in ascending order and print l2



# sort in descending order and then print l2
```

```
[11, 40, 43, 65, 85]
[85, 65, 43, 40, 11]
```

Lets try to append , insert , delete item from list

In [ ]:
```
# add item 99 in list and print l2



# insert item 500 in list at index 0
# we are going to use insert(index, item) method.


# remove items of index 1 from list



# remove last element from the list using pop() method


# remove item by value
```

```
[85, 65, 43, 40, 11, 99]
[500, 85, 65, 43, 40, 11, 99]
[500, 65, 43, 40, 11, 99]
[500, 65, 43, 40, 11]
[500, 65, 43, 11]
```

7. String value and operators

1 + 1 + 1 + 1 + 1 = 5 points

Characters in strings

Watch Video 15,15.1: If else and statement, indentation

In [ ]:
```
# declare the variables x = cat and y = dog and print it

x =
y =
```

```
cat
dog
```

In [ ]:
```

```

```
#Print first character of both x and y.
```

```
First ch of x =  c
First ch of y =  d
```

In [ ]:
```
#Print last character of both x and y.
```

```
Last ch of x =  t
Last ch of y =  g
```

In [ ]:
```
#Print len of both x and y
```

```
Lenght of x is  3
Length of y is  3
```

In [ ]:
```
#concatenate both x and y and store it in variable z and print z

z =
```

```
catdog
```

In [ ]:
```
#check if x is part of z
```

Out[ ]:
```
True
```

8. Dealing with Sentences

1+1+1+1+1+1 = 6 points

In [ ]:
```
# Declare a sentence of 'I am flying to London' and print it

sent =
```

```
I am flying to London
```

In [ ]:
```
#Print length of the sentence
```

```
Len of sentence is  21
```

In [ ]:
```
#Using list indexing, to print the word London
```

Out[ ]:
```
'London'
```

In [ ]:
```
# Use .split to get list of individual words in the sentence
```

Out[ ]:
```
['I', 'am', 'flying', 'to', 'London']
```

In [ ]:
```
# Convert sentence in lower case
```

Out[ ]:
```
'i am flying to london'
```

In [ ]:
```
# Convert sentence in upper case
```

Out[ ]:
```
'I AM FLYING TO LONDON'
```

In [ ]:
```
# Convert sentence in Camel case
```

Out[ ]:
```
'I Am Flying To London'
```

Lets try to print two string with inserting new line and tab

In [ ]:
```
# create two string varibles jack , hills

first_name=
last_name=
```

In [ ]:
```
# lets add tab space between first and last name

full_name1 =

print(full_name1)

# lets add new line between first and last name

full_name2 =

print(full_name2)
```

```
jack    hills
jack
hills
```

Lets Remove (Strip) white space from string

In [ ]:
```
s = ' This string contains white space. '

print(s)

# remove white space from both side.


# remove white space from left side.


# remove white space from right side.
```

```
 This string contains white space.
This string contains white space.
This string contains white space.
 This string contains white space.
```

In [ ]:
```
#Print length and verify
```

```
35
33
34
34
```

9. Loops and Iterations

If statement

1 + 2 + 3 = 6 points

In [ ]:
```
#Declare a variable x equal to a number of your choice.
#your code here

x = 3 # add a number of your choice (mine is 3 :) )
```

```
3
```

In [ ]:
```
#Using an if statement, print whether x is zero, positive or negative.
```

```
0 is zero
```

Watch Video 16: Loops

Watch Video 17: Loop Problem Statements in Python , Nested Loop Examples

For loop

2 points

In [ ]:
```
#Create a list x, such that it contains the elements - 'India', 'Israel','Canada'.

x =
```

In [ ]:
```
#Using For loop, iterate over the list and print the elements.
```

```
India
Israel
Canada
```

In [ ]:
```
print(range(1,11))
```

```
range(1, 11)
```

2 points

In [ ]:
```
# Using For loop, print the table of 2.
```

```
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
```

For a given list, x = [21, 'hello,'cream', 20,19,'village'], using for loop to iterate over the items and using - if statement print the items that are of type int

In [ ]:
```
#code here
```

```
[21, 20, 19]
```

Using For loop, lets create a pattern.

10 points

****

In [ ]:
```
# print the following pattern
```

```
*
* *
* * *
* * * *
* * * * *
```

Lets remove duplicates from list names=['A','B','C','A','D','E','F','G','H','E','D']

In [ ]:
```
name=['A','B','C','A','D','E','F','G','H','E','D']
```

```
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']
```

Good! Shall we try solving above problem in shorter way using list comprehension? It will be amazing. Here we go!

10. List Comprehensions

Hey buddy! Python has really amazing thing called list comprehension. List comprehension offers a shorter syntax when you want to create a new list based on the values of an existing list. Cool right? Let's dive into this more by doing it.

10 + 10 + 10= 30 points

In [ ]:
```

```

```
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']
```

In [ ]:
```
# Using list comprehension, create a list containing numbers from 1 to 20.
# Hint: You can use range function of python to do this.
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```

In [ ]:
```
# Using list comprehension, create a list containing numbers from 1 to 20 that are even
```

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

Create a list of all the fruits you like. Using list comprehension, create another list which contains the total number of charachters for each corresponding fruits.

- Eg. fruits = [‘apple’,’kiwi’,’orange’], then output would be - [5,4,6]

In [ ]:
```
fruits = ['apple','kiwi','orange'] # your code here (create a list of your favourite fruit
```

```
['apple', 'kiwi', 'orange']
[5, 4, 6]
```

The above output is shown if fruits = ['apple','kiwi,'orange'], it can vary for different values of fruits

Amazing right?

Write the code for the following, what is the output obtained?

- Set a variable flag = True, num = 5
- Run a while loop till flag = True
- Inside the while loop -
  - Increment the value of num by 5
  - If value of num is greater then 50, set flag to be False

In [ ]:
```
# your code here
```

```
5
10
15
20
25
30
35
40
45
50
```

Watch Video 18: Functions in Python

Watch Video 19: Top 5 Functions

11. Functions

Declare a function named square, which takes an input as x, and returns the square of x.

5 + 3 = 8 points

In [ ]:
```
#define function

def square(x):
    '''
        add your code
    '''
    num=x**2
    return num
```

Call the function to find the value of square of 9.

In [ ]:
```
# find the square of 9 using the function square you have just created.
```

```
Square of 9 is 81
```

Wohoo! see, defining function was simple

Define another function which checks whether a given is number is odd or even for a given input and it should print whether the number is odd or even as an output

5 + 3 = 8 points

In [ ]:
```
# your code here

def check_odd_even(x):
    # code below
```

In [ ]:
```
 # your code here
```

```
3 is odd
6 is even
21 is odd
```

define a funtion 'occurence' which takes 2 inputs, one list 'lst' and another one a number 'n' and count the number of occurrences of n in your list lst.

In [ ]:
```
#your code
def occurence(lst, n):
    # code below



#lets use your function
```

Out[ ]:
```
3
```

Using while loop make a function that prints all the values between 1 to n.

10 points

In [ ]:
```
#define counter

def counter(n):
   # code below
```

In [ ]:
```
#this should print numbers from 1 to 10
```

```
1 2 3 4 5 6 7 8 9 10
```

In [ ]:
```
# Now Define a function, which takes input from the user and prints whether the given inpu

def check_num(x):
    # code below
```

```
3 is positive
-1 is negative
0 is zero
```

hey hey! You did a great code.

Smile, you completed the first milestone :)!!

FeedBack

We hope you've enjoyed this course so far. We're committed to help you use "AI for All" course to its full potential, so that you have a great learning experience. And that's why we need your help in form of a feedback here.

Please fill this feedback form
https://docs.google.com/forms/d/e/1FAIpQLSfjBmH0yJSSA34IhSVx4h2eDMgOAeG4Dk-yHid__NMTk3Hq5g/viewform