Python

In these days most used language in the world of programming is python since it finds important applications worldwide in many fields.

It is widely used language which is used for high-level programming purposes such as

- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia
- Scientific computing
- Text processing and many more...
- Machine learning

It is an Object-Oriented programming language which means the building block of this programming language is objects.

Python programs are generally shorter than programs in other language such as C, C++, Java etc. which is due to its shorter syntax and lack of braces and also having a vast library of functions help it to be short.

Most of the tech giant companies use python as one of their programming languages such as Google, Facebook, Microsoft etc.

It consists of many libraries such as NumPy, Pandas, Sklearn etc. which are used in machine learning which is one of the hottest fields right now in software industry.

Widely used compilers used for execution of programs in python are:

1. Anaconda

2. Pycharm

Basic Structure of Python for adding two numbers:

“””defining two variables”””

```
A = int(input())

B = int(input())
```

“””Adding two variables “””

C = A+B

“””printing C”””

print(C)

How to take input from user?

We can take in python by defining a variable as:

A = int(input())

Here A is a variable which is taking integer input.

int is the data type of variable. We need to define the data type of variable when we take it as input in python.

Examples:

X= double(input())

Printing a statement in python:

For printing a string in python syntax is as:

print(“Here is the string”)

print(‘Here is the string’)

We can use both “” and ‘’ in python for printing a statement in python.

Conditional statements in python:

if statement

As like in other languages, python has similar conditional statements but there is very much change in syntax for conditional statements in python as:

a = 33
b = 200

```
if b > a:
  print ("b is greater than a")
```

Here a and b are two numbers which are being compared in if statement.

And b>a is the condition for if statements, if the condition is true then it will perform that action defined inside the if statement. In python, {} brackets are not used for closing and opening of if statement. Instead of that in python for if ‘: ‘is used.

Indentation

Python relies on indentation (whitespace at the beginning of a line) to define scope in the code. Other programming languages often use curly brackets for this purpose. Since python does not have any curly braces spacing gets very important in python.

Example:

If statement, without indentation (will raise an error):

```
a = 33
b = 200
if b > a:
print ("b is greater than a") # you will get an error
```

In python, most important thing is the space in next line to if statement as

```
if b > a:
  print ("b is greater than a")
```

If there is no space, then python cannot identify which actions are inside the statement or not and it will not perform the correct action. Instead it will be skipping that part since It’s not in if block.

else if (elif) statement

The elif keyword is python’s way of saying "if the previous conditions were not true, then try this condition". It works in same way as else if statement in other programming languages.

Example

```
a = 33
b = 33
if b > a:
  print ("b is greater than a")
elif a == b:
  print ("a and b are equal")
```

else Statement

The else keyword catches anything which isn't caught by the preceding conditions. It works in same way as else statement.

Example

```
a = 200
b = 33
if b > a:
  print ("b is greater than a")
elif a == b:
  print ("a and b are equal")
else:
  print ("a is greater than b")
```

In this example a is greater than b, so the first condition is not true, also the elif condition is not true, so we go to the else condition and print to screen that "a is greater than b".

Loops in Python:

A **for loop** is used for iterating a sequence as:

Example

Print each fruit in a fruit list:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

It will print:

apple

banana

cherry

Like conditional statements, indentation is important in loops for defining a for loop.

Break Statement:

In python break statement is used for stopping a loop after finishing a certain condition. It breaks the loop and moves onto next blocks of code after loop.

Example
Exit the loop when x is "banana":

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
  if x == "banana":
    break
```

The continue Statement

With the continue statement we can stop the current iteration of the loop, and continue with the next one. It's work is to skip the next lines of code in loop and move to next iteration.

Example
Do not print banana:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  if x == "banana":
    continue
  print(x)
```

The Range () Function:

Range function is used for looping in a certain range of numbers. The range () function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

Example
Using the range () function:

```
for x in range (6):
  print(x)
```

It will print numbers from 0 to 5 not including 6.

Nested Loops

A nested loop is a loop inside a loop.

The "inner loop" will be executed one time for each iteration of the "outer loop":

Example
Print each adjective for every fruit:

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]

for x in adj:
  for y in fruits:
    print (x, y)
```

While loop:

With the while loop we can execute a set of statements if a condition is true.

Example
Print i if i is less than 6:

```
i = 1
while i < 6:
  print(i)
  i += 1
```

Some libraries in python which are used in machine learning are:

1**. NumPy**: this library provides you with an array data structure that holds some benefits over Python lists, such as: being more compact, faster access in reading and writing items, being more convenient and more efficient.

2. **Pandas:** Pandas is an opensource library that allows to you perform data manipulation in **Python**.

3.**Matplotlib** = Matplotlib is a plotting library for the **Python** programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.

4.**scikit-learn** = scikit learn is an open source Python library that implements a range of machine learning, pre-processing, cross-validation and visualization algorithms using a unified interface.

Practice Questions

1) Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.

Solution:

```
name = input("What is your name: ")

age = int(input("How old are you: "))

year = str((2014 - age)+100)

print(name + " will be 100 years old in the year " + year)
```

2) Make a two-player Rock-Paper-Scissors game. (Hint: Ask for player plays (using input), compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game)

Remember the rules:

Rock beats scissors

Scissors beats paper

Paper beats rock

Solution:

```
import sys

user1 = input("What's your name?")

user2 = input("And your name?")

user1_answer = input("%s, do yo want to choose rock, paper or scissors?" % user1)

user2_answer = input("%s, do you want to choose rock, paper or scissors?" % user2)


def compare(u1, u2):
    if u1 == u2:
        return("It's a tie!")
    elif u1 == 'rock':
        if u2 == 'scissors':
            return("Rock wins!")
```

```
        else:
            return("Paper wins!")
    elif u1 == 'scissors':
        if u2 == 'paper':
            return("Scissors win!")
        else:
            return("Rock wins!")
    elif u1 == 'paper':
        if u2 == 'rock':
            return("Paper wins!")
        else:
            return("Scissors win!")
    else:
        return("Invalid input! You have not entered rock, paper or scissors, try again.")
        sys.exit()
print(compare(user1_answer, user2_answer))
```

3) Ask the user for a number. Depending on whether the number is even or odd, print out an appropriate message to the user.

Hint: how does an even / odd number react differently when divided by 2?

Extras:

If the number is a multiple of 4, print out a different message.

Ask the user for two numbers: one number to check (call it num) and one number to divide by (check). If check divides evenly into num, tell that to the user. If not, print a different appropriate message.

Solution:

```
num = input("Enter a number: ")
mod = num % 2
```

```
if mod > 0:
    print("You picked an odd number.")
else:
    print("You picked an even number.")
```

4) Ask the user for a string and print out whether this string is a palindrome or not. (A palindrome is a string that reads the same forwards and backwards.)

Solution:

```
wrd=input("Please enter a word")
wrd=str(wrd)
rvs=wrd[::-1]
print(rvs)
if wrd == rvs:
    print("This word is a palindrome")
else:
    print("This word is not a palindrome")
```

5) Write a program that asks the user how many Fibonacci numbers to generate and then generates them. Take this opportunity to think about how you can use functions. Make sure to ask the user to enter the number of numbers in the sequence to generate.(Hint: The Fibonacci seqence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13, …)

Solution:

```
def gen_fib():
    count = int(input("How many fibonacci numbers would you like to generate? "))
    i = 1
    if count == 0:
        fib = []
```

```
    elif count == 1:
        fib = [1]
    elif count == 2:
        fib = [1,1]
    elif count > 2:
        fib = [1,1]
        while i < (count - 1):
            fib.append(fib[i] + fib[i-1])
            i += 1
    return fib
```

For more practice, visit the unsolved section or practice other language questions using python as questions remain same.