

Programming 102

Hands-on Python Syntaxes

Kuldeep Singh Sidhu

kuldeepsinghsidhu.com

By the end of this session

- You will know What Python is!
- You will have Python ready and running
- You will know Python Keywords, Identifiers and Data Types
- You will be able to work with Loops in Python
- Will have done a lot of hands-on coding

What is Python?

Python is a programming language.

Python is used to talk to machines.

Python is a mature programming language (since
February 20, 1991)



If there are other languages, why **Python**?

Python is one of the most easiest and friendly language.

Python has one of the largest developer community.

> **Python** is a multi-purpose language.



> **Python** is highly focused on code readability

The creator of Python Guido van Rossum has stated that code readability was his biggest motivation behind creating Python

Python is a multi-purpose language.

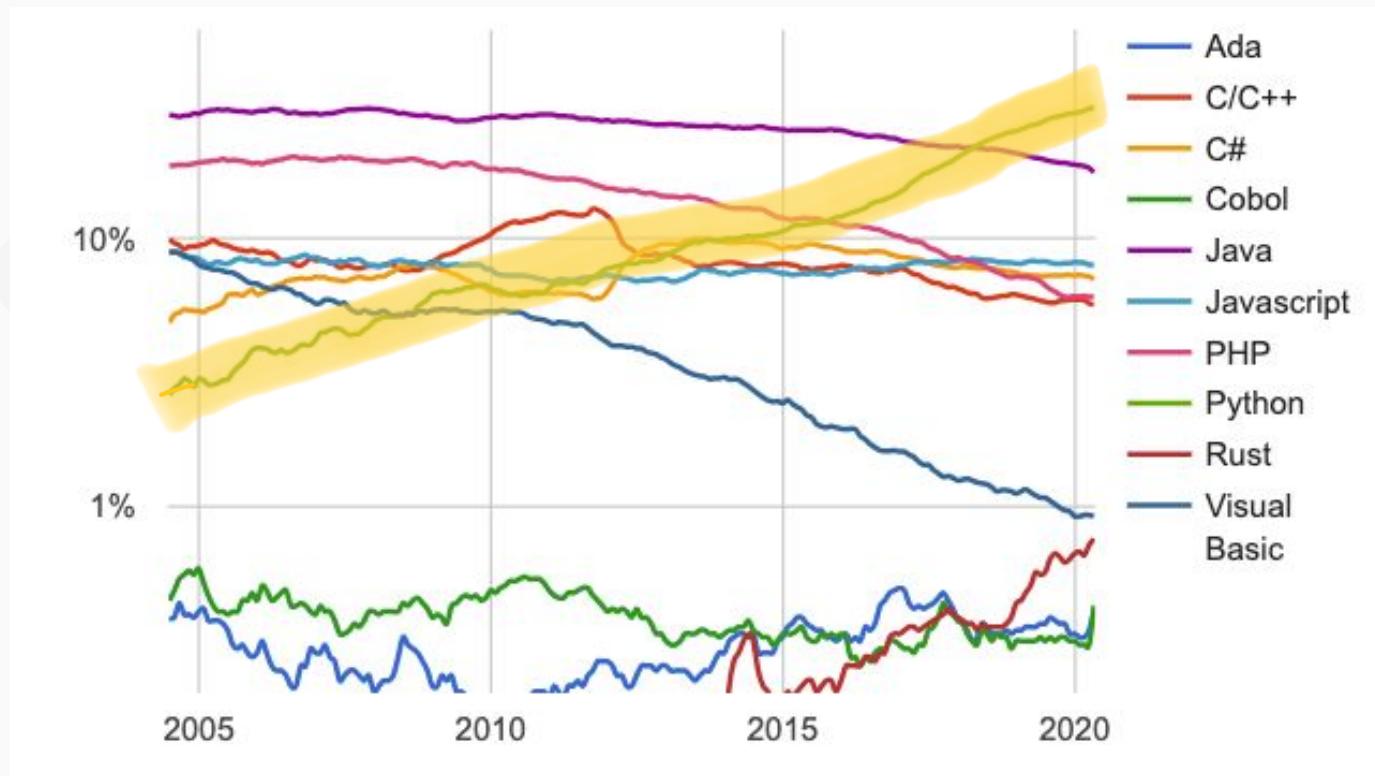
- ✓ Web Scraping
- ✓ Data Science and Visualisation
- ✓ Machine Learning and AI
- ✓ Desktop GUIs
- ✓ Web development
- ✓ Automation

Software development

And much more....

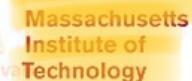
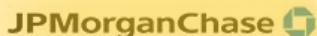


Python is the most popular coding language



Python is used by many top companies

Top Companies Using Python



Why has Python become a first choice for development?

Object Orientation

Open Source

Power Library

Portability and Platform Independence

Multipurpose

And much more...

300k+ active projects

KuldeepSinghSidhu.com

Python Setup

On your personal system

<https://docs.anaconda.com/anaconda/install/>

Using free Online servers

<https://colab.research.google.com/>



ANACONDA.



Python Setup

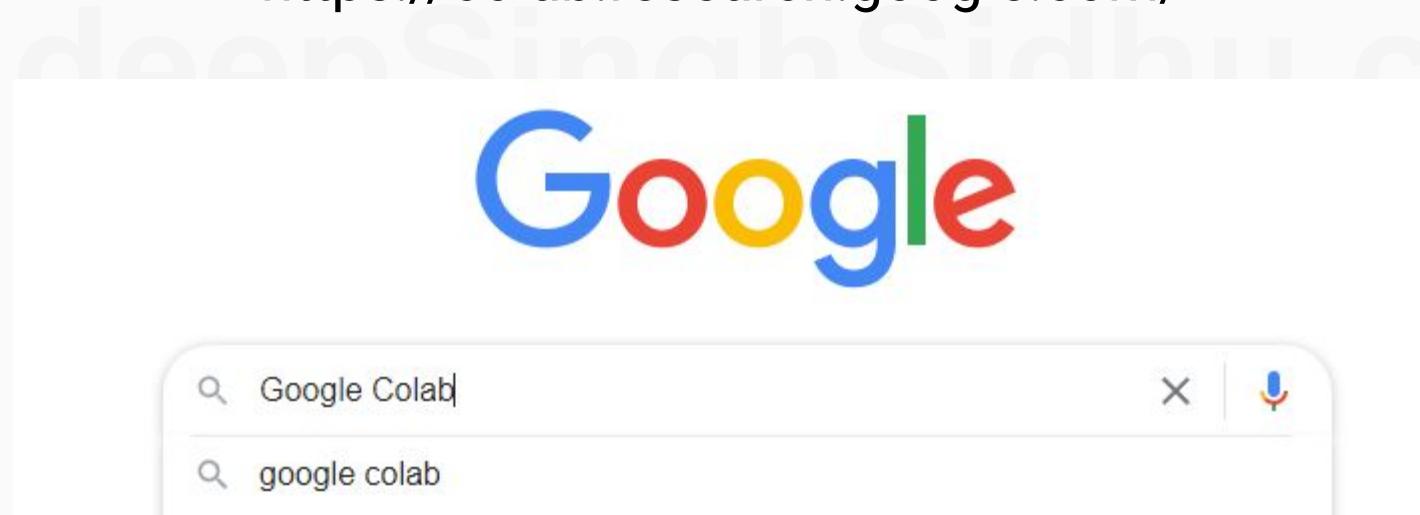
On your personal system	Using Online servers
https://docs.anaconda.com/anaconda/install/	https://colab.research.google.com/
Very popular Python distribution <u> </u>	Popular and Powerful <u>online</u> jupyter environment
Comes with most of the <u>popular libraries</u> pre - installed	Comes with most of the <u>popular libraries</u> pre - installed
IDE (Integrated development environment), <u>Jupyterlab, Spyder, PyCharm</u>	IDE (Integrated development environment) <u>Custom Jupyterlab</u>





Understanding Google Colab

<https://colab.research.google.com/>



Programming language

Any language has 2 things:

- Vocabulary *Apple, day, doctor*
- Grammar *An apple a DAY keeps the doctor away*

Any programming language has 2 things:

- keywords
- syntax

*- Case sensitive
- Precisely typed P#P*

99% of python errors

print ('hi') ✓

print ("hi") ✓

prent ('hi') ✗

Print ('hi') ✗ p ≠ P

print{ 'hi'} ✗ C) ≠ C 3 + C]

print(hi) ✗ (...) C ...)

print('hi') ✗ ' ... ' " ... "

print('hi') ✗ _ " =



```
1. # this is a comment as it starts with a #
2. # and I will be ignored
3. print('This code will be run')
4. print('And this will run') # but this won't as it is
   a comment
5. print("Python #1 and it's popular")
6. this is not a code nor a comment and will give an
   error
7. print("""
8.     This is a multiline string
9.     That goes to multiple lines
0.     """)
```

All functions have () at the end

1. print("called a print funtion")
2. help('keywords') # to print python keywords
3. 2 is 3 # ignore for now

Data Types in Python - Python way of understanding data

1. "Ronaldo" # string
2. 7 #int
3. 3.14 #float
4. True #bool
5. False #bool

Data Types in Python - Python way of understanding data >> **variables**

Variable names cannot start with numbers or have spaces OR be a keyword

1. name='Kuldeep'
2. age=25
3. height=181.5
4. isIndian=True
5. greaterThan6ft=False

1. # these will give errors
2. 2ab=1
3. my name is='Kuldeep'

1. print(name)
2. print(age)
3. print(height)
4. print(isIndian)
5. print(greaterThan6ft)
6. print(name, age, height, isIndian, greaterThan6ft)#printing in the same line, seperated

Alphabetical/alpha-numeric data cannot be without quotations

Data Types in Python - Python way of understanding data >> The **type()** function

```
1. "Ronaldo" # string  
2. 7 #int  
3. 3.14 #float  
4. True #bool  
5. False #bool  
6. # get data type  
7. print(type("Ronaldo"))  
8. print(type(7))  
9. print(type(3.14))  
10. print(type(True))  
11. print('CR7') #what is the data type for 'CR7'
```

Data Types in Python - Python way of understanding data >> The **type()** function

```
1. name='Kuldeep'  
2. age=25  
3. height=181.5  
4. isIndian=True  
5. greaterThan6ft=False  
6. print(name, type(name))  
7. print(age, type(age))  
8. print(height, type(height))  
9. print(isIndian, type(isIndian))  
10. print(greaterThan6ft, type(greaterThan6ft))
```

Getting data from the user>> The **input()** function

```
1. # Asking user for input
2. # input() creates a text box
3. myName = input()
4. print('My name is', myName)
5. city = input("enter your city")
6. food = input('enter your food')
7. print(city, food)
8. age = input('Enter you age ')
9. print('Your age is', age, 'Age is of type', type(age))
```

Anything accepted by **input()** is string by default even numbers will be treated as a character data and not numeric
Int, bool, float >> anything entered is considered as str as input() only return str

Getting data from the user>> Type Conversion

```
1. age = input('Enter your age ')
2. print('Your age is', age, 'Age is of type', type(age))
3. #gives error
4. # print(age+1)
5.
6. # type conversion
7. print(int(age) + 1)
```

Issue with this: You cannot do mathematical operations eg: age+1

Anything accepted by `input()` is string by default even numbers will be treated as a character data and not numeric

Getting data from the user>> Type Conversion

```
age = int(input('Enter your age '))  
print('Your age is', age, 'Age is of type', type(age))  
print(age+1)
```

KuldeepSinghSidhu.com

Issue with this: You cannot do mathematical operations eg: age+1

Anything accepted by `input()` is string by default even numbers will be treated as a character data and not numeric

Getting data from the user>> Understanding step by step

```
1 #converting kg to grams  
2 kg=int(input('Enter weight in kg: '))  
3 gms=kg*1000  
4 print('weight in grams', gms)
```

Enter weight in kg:

Getting data from the user>> **Understanding step by step**

```
1. # average of 2 numbers
2. first = int(input('Enter first number: '))
3. second = int(input('Enter second number: '))
4. total = first+second
5. avg=total/2
6. print('Average is', avg)
```

Division always returns float

RECAP

1. Python is amazing 😊
2. Setting up and using Python
3. Python is case sensitive and like any programming language it's precisely typed
4. We looked at the common errors
5. Comments start with #
6. Functions end with ()
7. Using print() function to display output
8. Str, int, float, bool data types
9. Storing data in variables
10. Variable names cannot start with numbers or have spaces OR be a keyword >>>
`help('keywords')`
11. Alphabetical/alpha-numeric data cannot be without quotations
12. Accepting data from user using input()
13. Converting data types
14. Converting str to int to do mathematical operations
15. Looked at step by step look at the code
16. Division always returns float

Coding Exercise: Write a python program to accept Principle, Rate of interest and Time from the user and display the simple interest (this is an integer)

```
1. # conversion tips  
2. ## converts to int  
3. var = int(input())  
4. ## converts to float  
5. var = float(input())  
6.  
7. # simple interest formula  
8. si = (p*r*t)/100
```

Coding Exercise: Write a python program to accept Principle, Rate of interest and Time from the user and display the simple interest (this is an integer)

```
1. p = int(input('Enter principle: '))
2. r = float(input('Enter rate: '))
3. t = int(input('Enter time: '))
4. si = int((p*r*t)/100)
5.
6. print(f"""
7. Principle {p}
8. Rate {r}
9. Time {t}
10.
11. Simple Interest is {si}
12. """)
```

String formatting >>> using **f strings**

```
1. flower = 'Rose'  
2. color = 'Red'  
3. print(f'The {flower} are {color}')  
4.  
5. # f-strings work with all data types  
6. canVote = True  
7. age = 25  
8. print(f'It is {canVote}, you can vote as you are  
{age} yrs old')
```

F-strings provide a way to embed expressions inside string literals, using a minimal syntax.
-Python Org Docs

Syntax: Enclose variables in {} and add f

Coding Exercise: Write a python program to accept name, age and city and use f-string to print them

Syntax: Enclose variables in {} and add f

KuldeepSinghSidhu.com

Decision making in python >>> if-else

```
1. # program to find if a non-zero number is positive or negative
2.
3. num = int(input('Enter any number:\n'))
4. #\n adds a line
5.
6. if num>0:
7.     print('+ve')
8. else:
9.     print('-ve')
```

Look at :, it is used to declare a block. Putting : IDE will indent the cursor automatically
Indent helps to tell which code is under influence of the if condition

Decision making in python >>> **if-else** understanding influence of if statements because of indentation

```
1. myBirthday=True
2.
3. if myBirthday:
4.     print('Happy birthday to me')
5.     print('Birthday Party')
6.
7. if myBirthday:
8.     print('Happy birthday to me')
9. print('Birthday Party')
10. # will run even if not a birthday
```

Look at :, it is used to declare a block. Putting : IDE will indent the cursor automatically
Indent helps to tell which code is under influence of the if condition

Coding Exercise:

Write code to ask the user their age and check if they can vote or not.

KuldeepSinghSidhu.com

Decision making in python >>> MULTIPLE **if-else**

```
1. # As per the rank of a student display the division
2. rank = int(input('What is your rank: '))
3. if rank==1:
4.     print('First Division')
5. elif rank==2:
6.     print('Second Division')
7. elif rank==3:
8.     print('Third Division')
9. else:
10.    # this executes when all above conditions are False
11.    # that is value is not 1,2 or 3
12.    print('No division')
```

== is used to check if 2 values are equal
We use **elif** to make another decision when the above decision fails

Coding Exercise:

Write a program that accepts a number and determines if it's +ive, -ive, neutral (zero)

KuldeepSinghSidhu.com

Loops in python (used for solving repetitive tasks) **Understanding the flow of execution**
You can see loops in music players, YouTube even websites will looping slideshows.

```
1. meter=1
2. while(meter<=5):
3.     print(meter)
4.     meter=meter+1
```

memory
(mutation)

1
2
3
4
5
6

Display:

1
2
3
4
5

Python has 2 loops **for** and **while**

Loops in python (used for solving repetitive tasks) **Understanding the flow of execution**
You can see loops in music players, YouTube even websites will looping slideshows.

```
1. meter=1  
2. while (meter<=5) :  
3.     print(meter)  
4.     meter=meter+1
```

Exercise:

1. Can you print from 1 to 10?
2. Can you print from 2 to 5?
3. Can you print from 2 to 10 but only even numbers?
4. Can you print from 5 to 50 but only multiples of 5?

Python has 2 loops **for** and **while**

Loops in python (used for solving repetitive tasks) **Understanding the flow of execution**
You can see loops in music players, YouTube even websites will looping slideshows.

```
1. meter=5
2. while (meter<=50) :
3.     print (meter)
4.     meter+=5
5.     # this is short for meter=meter+5
```

Exercise:

1. Can you print from 1 to 10?
2. Can you print from 2 to 5?
3. Can you print from 2 to 10 but only even numbers?
4. **Can you print from 5 to 50 but only multiples of 5?**

Python has 2 loops **for** and **while**

Loops in python (used for solving repetitive tasks) **Understanding the flow of execution**
You can see loops in music players, YouTube even websites will looping slideshows.

Display:

```
milk  
butter  
bread
```

```
1. for bag in ['milk', 'butter', 'bread']:  
2.     print(bag)  
3.     # bag is a temporary variable  
4.     # all elements will be assigned to bag 1 by 1  
5.     # When the list finishes, it auto terminates
```

Python has 2 loops **for** and **while**

Loops in python (*used for solving repetitive tasks*) **Difference between for and while**

You can see loops in music players, YouTube even websites will looping slideshows.

- `for` has no condition
- `for` has no increments/decrements
- Only `while` loops work when we need condition execution and not sequential

Python has 2 loops **for** and **while**

Loops in python (*used for solving repetitive tasks*) **Cannot do this using for loop**

You can see loops in music players, YouTube even websites will looping slideshows.

```
1. choice='y'  
2. while choice=='y':  
3.     choice = input('Do you want to continue: ')
```

Python has 2 loops **for** and **while**

Fundamentals of Python Programming

pdf ebook tutorial on 669 pages
by Richard L. Halterman.

 DOWNLOAD

PDF

Fundamentals of Python Programming

DRAFT

Richard L. Halterman
Southern Adventist University

July 26, 2018

Link:<https://www.computer-pdf.com/programming/804-tutorial-fundamentals-of-python-programming.html>

<https://gist.github.com/singhsidhukuldeep/6618b9aea2c6bdb86aaf59c2799c009b>

Google Search:
fundamentals of python programming pdf



KuldeepSinghSidhu.com