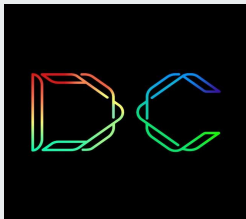




Facebook Dev Circles Meetup - Data Science Track

Python for Data Science

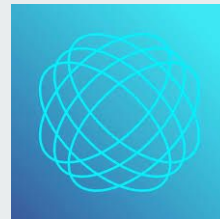
Team Dev Cluster



DataCamp



Dev Circles from Facebook





**Tag us on Facebook, Twitter, Instagram and LinkedIn
@devclustergoa and *use #DevCTraining***

**For any queries or doubts regarding courses of the Data Science track,
*Contact us-***

**Prachi Mahambrey - 8668845314
Vardhana Navelkar - 8329551370**



Lesson Plan

- Python Basics
- Python Variables and their types
- Python Lists
- Python Dictionaries
- Python Functions, Methods
- Numpy
- Pandas
- Matplotlib



Python Basics

`print()`

`Basic Arithmetic`

`Comments`



print()

- Used to print all sorts of data in python.
- Syntax: print(x)

- **Examples-**

I/P:

```
print(5 + 5)
```

O/P:

```
<script.py> output:  
10
```

I/P:

```
savings = 100  
print(savings)
```

O/P:

```
<script.py> output:  
100
```



Basic Arithmetic

Python is perfectly suited to do basic calculations. Apart from addition, subtraction, multiplication and division, there is also support for more advanced operations such as:

- Exponentiation: `**`
- Modulo: `%`



Basic Arithmetic

Example-

I/P:

script.py

```
1 print(5 + 5)
2 print(5 - 5)
3 print(3 * 5)
4 print(10 / 2)
5 print(18 % 7)
6 print(4 ** 2)
```

O/P:

```
<script.py> output:
10
0
15
5.0
4
16
```



Comments

- To add comments to your Python script, you can use the # tag.

Examples-

```
# Division  
print(5 / 8)
```

```
# Addition  
print(7 + 10)
```

```
# Create a variable savings  
savings = 100  
print(savings)
```




Python Variables

- **Variable Assignment & Calculations with Variables**
- **Variable Types**
- **Operations with Variables**
- **Type Conversion**



Variable Assignment & Calculations with Variables

- To create a variable, use '='
- *Example-*
 $x = 5$
- Use the name of this variable, `x`, instead of the actual value, `5`.



Variable Types

- `int`, or integer
- `float`, or floating point
- `str`, or string
- `bool`, or boolean



Operations with Variables

- Different types behave differently in Python.

Note: Suppose you've defined a variable `a`, but you forgot the type of this variable. To determine the type of `a`, simply execute: `type(a)`



Type Conversion

- *Example-*

```
print("I started with $" + savings + " and now have $" + result + ". Awesome!")
```

This will not work as you cannot simply sum strings and floats.

To make it work:

- Use *str(savings)* instead.



Logic, Control Flow, Filtering and Loops

- **Comparison operators**
- **Boolean operators**
- **if, elif, else**
- **while loop**
- **for loop**



Comparison Operators

- *Equality: ==*
- *Not equal to: !=*
- *Greater than: >*
- *Less than: <*
- *Greater than or equal to: >=* (and not =<)
- *Less than or equal to: <=*



Boolean Operators

- *and*
- *or*
- *not*

*Note: Notice that **not** has a higher priority than **and** and **or**, it is executed first.*



if-elif-else

- ***Syntax-***

if **condition 1** :
 statements inside if

elif **condition 2** :
 statements inside elif

else
 statements inside else



While loop

- *while loop* is like a **repeated if loop**.

- ***Syntax-***

```
while condition :  
    Expression
```

- ***Example-***

```
x = 1  
while x < 4 :  
    print(x)  
    x = x + 1
```



For loop

- *Let us understand for loop with the following example.*
- *Example-*

```
# areas list  
areas = [11.25, 18.0, 20.0, 10.75, 9.50]
```

```
# Code the for loop  
for element in areas:  
    print(element)
```



Python Lists

- **What are Lists? Create a List.**
- **List with different types**
- **List of Lists**
- **Subsetting Lists**
- **Slicing and Dicing**
- **Subsetting Lists of Lists**
- **List Element Manipulations:
Replace, Extend a List, Delete**
- **Inner Working of Lists**



What are Lists? Create a List

- A *list* is a **compound data type**.
- It can group values together.

- *Example-*

```
a = "is"
```

```
b = "nice"
```

```
my_list = ["my", "list", a, b]
```



Lists with different types

- A list can contain any Python type.
- A list can also contain a mix of Python types including **strings**, **floats**, **booleans**, etc.
- *Example-*

```
['hallway', 11.25, 'kitchen', 18.0, 'living room', 20.0, 'bedroom', 10.75, 'bathroom', 9.5]
```



List of Lists

- Instead of creating a flat list containing strings and floats, you can also create a ***list of lists***.
- ***Example-***

```
[['hallway', 11.25], ['kitchen', 18.0], ['living room', 20.0],  
 ['bedroom', 10.75], ['bathroom', 9.5]]
```



Subsetting Lists

- Consider the list-
`x = ["a", "b", "c", "d"]`
- To access ***b***,
`x[1]` or `x[-3]`
- `print(x[1] + x[3])` gives output as **bd**



Subsetting Lists

- To select multiple elements from the list, use-

```
my_list[start:end]
```

- ***start*** is included, ***end*** is **not** included.

- ***Example-***

```
In x = ["a", "b", "c", "d"]
```

```
x[1:3] gives output ['b', 'c']
```



Subsetting Lists

Examples-

In `x = ["a", "b", "c", "d"]`

`x[:2]` gives output `['a', 'b']`

`x[2:]` gives output `['c', 'd']`

`x[:]` gives output `['a', 'b', 'c', 'd']`



Subsetting Lists of Lists

- *Example-*

```
x = [ ["a", "b", "c"],  
      ["d", "e", "f"],  
      ["g", "h", "i"] ]
```

- **x[2][0]** gives output 'g'
- **x[2][:2]** gives output ['g', 'h']



List Manipulations

For the list : `x = ["a", "b", "c", "d"]`

- ***Replace list Elements***

`x[1] = "r"` or `x[2:] = ["s", "t"]`

- ***Extend a list***

`y = x + ["e", "f"]`

- ***Delete list elements***

`del(x[1])`



Dictionaries

- **Why Dictionaries ?**
- **Definition**
- **Create, Update & Delete**
- **Dictionary of dictionaries**



Why dictionaries?

- *Not convenient*
- *Not intuitive*



Definition

- ***Syntax***

```
Dict_name = { key:value,....}
```

- ***Accessing a value***

```
Dict_name[key]  
result = value
```

- ***Keys have to be immutable***

```
In [5]: {0:"hello", True:"dear", "two":"world"}  
Out[5]: {0: 'hello', True: 'dear', 'two': 'world'}
```

```
In [6]: {["just", "to", "test"]: "value"}  
TypeError: unhashable type: 'list'
```



List vs Dictionaries

List	Dictionary
Select, update and remove: []	Select, update and remove: {}
Indexed by range of numbers	Indexed by unique keys
Collection of values order matters select entire subsets	Lookup table with unique keys



Dictnariception !

- *Dictionary of dictionaries*

```
Dict_name = { key:{key:value,....}, key  
: {key:value,....},..... }
```



Functions & Methods

- **Familiar functions**
- **Functions with multiple arguments**
- **Methods**



What are functions?

- *Nothing new!*
- *Piece of reusable code*
- *Solves particular task*
- *Call function instead of writing code yourself*



How to find functions?

- *Standard task -> probably function exists!*
- *The internet is your friend*



Methods?

- *Functions that belong to objects*
- *Object have methods associated, depending on type*



Functions vs Methods?

- ***Functions***

```
In [11]: type(fam)
Out[11]: list
```

- ***Methods***

```
In [12]: fam.index("dad")
Out[12]: 6
```



Packages

- What are packages ?
- Installing a package



What are packages?

- *Directory of Python Scripts*
- *Specify functions, methods, types*



Installing a package

- *Download get-pip.py*
- *Terminal:*
python3 get-pip.py
pip3 install numpy



NumPy

- **Introduction**
- **NumPy Arrays**



Introduction

- *Numeric Python*
- *Alternative to Python List: NumPy Array*
- *Calculations over entire arrays*



NumPy Arrays

- *Element-wise calculations*
- *Contain only one type*



Matplotlib

- Basic Plots - **Line plot, Scatter plot, Histogram**
- Labels
- Ticks
- Sizes
- Colors



Line Plot

- *Syntax-*

```
import matplotlib.pyplot as plt  
plt.plot(x,y)  
plt.show()
```



Scatter Plot

- *Syntax-*

```
import matplotlib.pyplot as plt  
plt.scatter(x,y)  
plt.show()
```



Scatter Plot

- *Syntax-*

```
import matplotlib.pyplot as plt  
plt.scatter(x,y)  
plt.show()
```




Histogram

Refer examples for syntax.



Labels

- To add **axis labels** and **title** to the plot.
- You can use **xlabel()**, **ylabel()** and **title()**.



Ticks

- To label or name the values in the plot.
- You can use **`xticks()`**.



Sizes and colors

Refer examples for syntax.

Thank you :)

