Introduction to Python

A technology awareness session on Python programming language

By Dhruv Desai,
Software Engineer, Narola Infotech

Contents

- Introduction to Python
- Basic programing concepts
- Python with Files & Database
- GUI with Python
- Building a simple website
- Technology trends & Python
- Resources & Materials

What is Python?

Definition: Python is an interpreted, high-level, general-purpose programming language

Just another one those popular & trending programing languages out there.

Actually : A fun tool to play around and make things and explore our technological advancement.

Some background history

Creator: Guido van Rossum

First release: 1991

Python2 and Python3

Python3 release: 2008

Programing in Python

```
a = 2
b = 3
sum = a + b
print(sum)
```

```
im_a_parent:
    im_a_child:
        im_a_grand_child
    im_another_child:
        im_another_grand_child
```

Data Types

1. bool: Boolean (true/false) types.
2. int: Signed integer types.
3. uint: Unsigned integer types.
4. float: Floating point types.
5. complex: Complex number types.
6. string: Raw string types
7. time: Data/time types
8. enum: Enumerated types

...boring.

Just remember these 3

- number
- string
- boolean

Data Structures in Python

[] - List

{} - Dictionary

() - Tuple

same old “if else”

```
if number > 4:
    print("greater than 4")
elif number > 8:
    print("greater than 8")
else:
    do_some_stuff_here()
```

```
some_list = [10, 29, 44, 6]
if 29 in some_list:
    print(True)
else:
    print(False)
```

```
lazy_about_brackets = True
if lazy_about_brackets:
    print("Go Python")
else:
    print("Go Java")
```

```
some_string = "python"
if 'y' in some_string:
    print(True)
else:
    print(False)
```

same old “for loop”

```python
for i in range(1,11):
    print(i)
```

```python
for char in "python":
    print(char)
```

```python
print("Robert Downey Jr. is")
for char in ["Iron Man", "Tony Stark", "Sherlock"]:
    print(char)
```

Oops...Classes & Objects

Yes, Python is an object oriented language means it supports & follows OOP concepts.

- Object
- Class
- Method
- Inheritance
- Polymorphism
- Data Abstraction
- Encapsulation

Simple Stuff we need to know

Not simple stuff ?

Some example of class in Python

```
class Calculator:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def add(self):
        return self.x + self.y

    def subtract(self):
        return self.x - self.y
```

```
calc = Calculator(30,17)
calc.add()
calc1 = Calculator(12,9)
calc1.subtract()
```

Modules & Packages

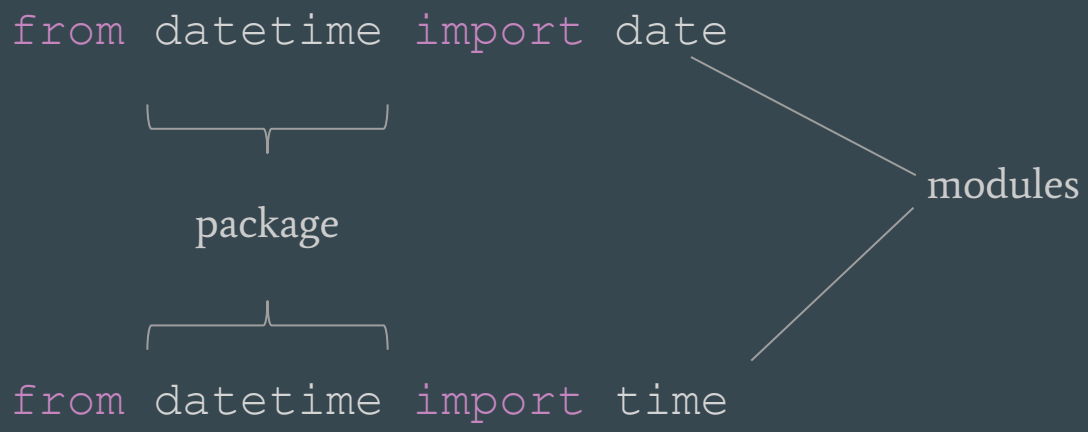
Module:

A Python module is a collection of related classes and functions. We have modules for mathematical calculations, string manipulations, web programming, and many more.

Package:

A package is a collection of related modules. You can either import a package or create your own.

Modules & Packages



Connecting to Database

Default database: **Sqlite**

Python package to use sqlite database: **sqlite3.**

```
>>>import sqlite3
```

Establishing a database connection using **connect()** method of sqlite3.

```
>>>conn = sqlite3.connect('test.db')
```

executing SQL queries

execute() method which takes SQL queries as arguments is used to interact with database.

```
conn.execute('''CREATE TABLE COMPANY
       (ID INT PRIMARY KEY,
       NAME TEXT,
       AGE INT,
       ADDRESS CHAR(50),
       SALARY REAL)'''
   )
conn.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) VALUES (1,
'Paul', 32, 'California', 20000.00 )" )
cursor = conn.execute("SELECT * from COMPANY" )
```

Graphical User Interface with Python

Python offers multiple options for developing GUI (Graphical User Interface).

- tkinter
- wxPython
- PyQt
- Kivy

Out of all the GUI libraries, **tkinter** is most basic & commonly used.

```
import tkinter as tk
```

GUI is made up of combinations of windows.

Main element is a window, which contains widgets.

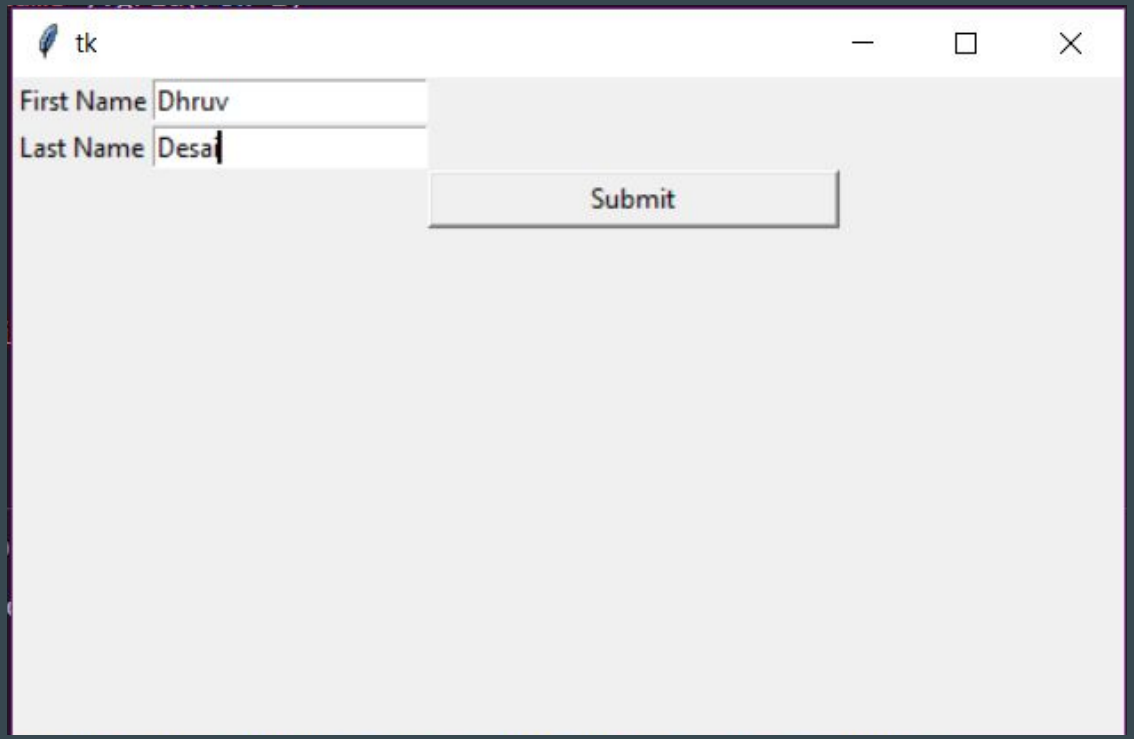
Some basic widgets are

- Button
- Checkbox
- Entry
- Label
- Frame
- Menu

GUI cont.

- tk.Tk()
  - `master = Tk()`
- tk.Button()
  - `Button(master, text='Submit', width=25, command=fetch_value)`
- tk.Label()
  - `Button(master, text='Submit', width=25, command=fetch_value)`
- tk.Entry()
  - `Button(master, text='Submit', width=25, command=fetch_value)`

A simple GUI window



Building a website with Python

Web Frameworks:

- **Django:**
  - Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.
- **Flask:**
  - Flask is a Python web framework built with a small core and easy-to-extend philosophy.

Building Website with Flask

Flask:

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.

Installing Flask in Python:

```
pip install Flask
```

Let’s see some Flask code

app.py

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello():
    return 'Hello, World!'
```

Technology trends & Python

- Web Development
- Internet of Things (IOT)
- Machine Learning & AI
- Data Analysis
- Data Science
- Big Data

Reference & materials

Tkinter demos to get started

- Basic: https://github.com/emailman/tkinter_demos
- Some advance demos: https://github.com/errnox/tkinter-gui-demos
- https://www.tutorialspoint.com/python3/python_gui_programming.htm
- https://github.com/Miraj50/Awesome-Tkinter-Apps

For Flask

- https://realpython.com/tutorials/flask/

Github Repository

- https://github.com/dhruv-3d/PySession

Thank You