INTRODUCTION TO PROGRAMMING

PYTHON

Agenda

- Introduction to programming languages
  - What is a programming language
  - Levels of programming languages
  - Parameters to define a good programming language
  - Functional programming
  - Object oriented programming
- What is AI and Data Science
- Programming languages for AI and data science
- Python for AI and data science
- Salient features of Python
- Applications of Python
- External tools & environment connectivity with Python
- Designing and deploying software products with python
- Next steps

INTRODUCTION TO PROGRAMMING LANGUAGES

What is a programming language?

Why do we need programming?

- To communicate with computers
- Language for computers to perform various tasks like printing.
- To write and execute algorithms specified by humans
- To process data and derive inferences

What is a programming language?

Features of programming languages

- Syntax:- grammar for programming languages
- Programming paradigm:- style of programming
- Types:-
  - Functional programming
  - Object oriented programming
  - Procedural Programming Language
  - Scripting Programming Language
  - And many more

Java

Python

C++

Levels of programming languages

- High-level
  - Python
  - Java
- Low-level
  - Assembly language
  - Machine code

```
A = 23
B = 2
def sum(a, b):
        return a+b;
sum(A,B)
```

```
ADD 12
LOAD 1

1001101100000101
```

Functional Programming

- Uses expressions to write programs
- Declarative programming

```
>>> animals = ["ferret", "vole", "dog",
"gecko"]
>>> def reverse_len(s):
        return -(s)
>>> sorted(animals, key=reverse_len)
['ferret', 'gecko', 'vole', 'dog']
```

Object Oriented Programming

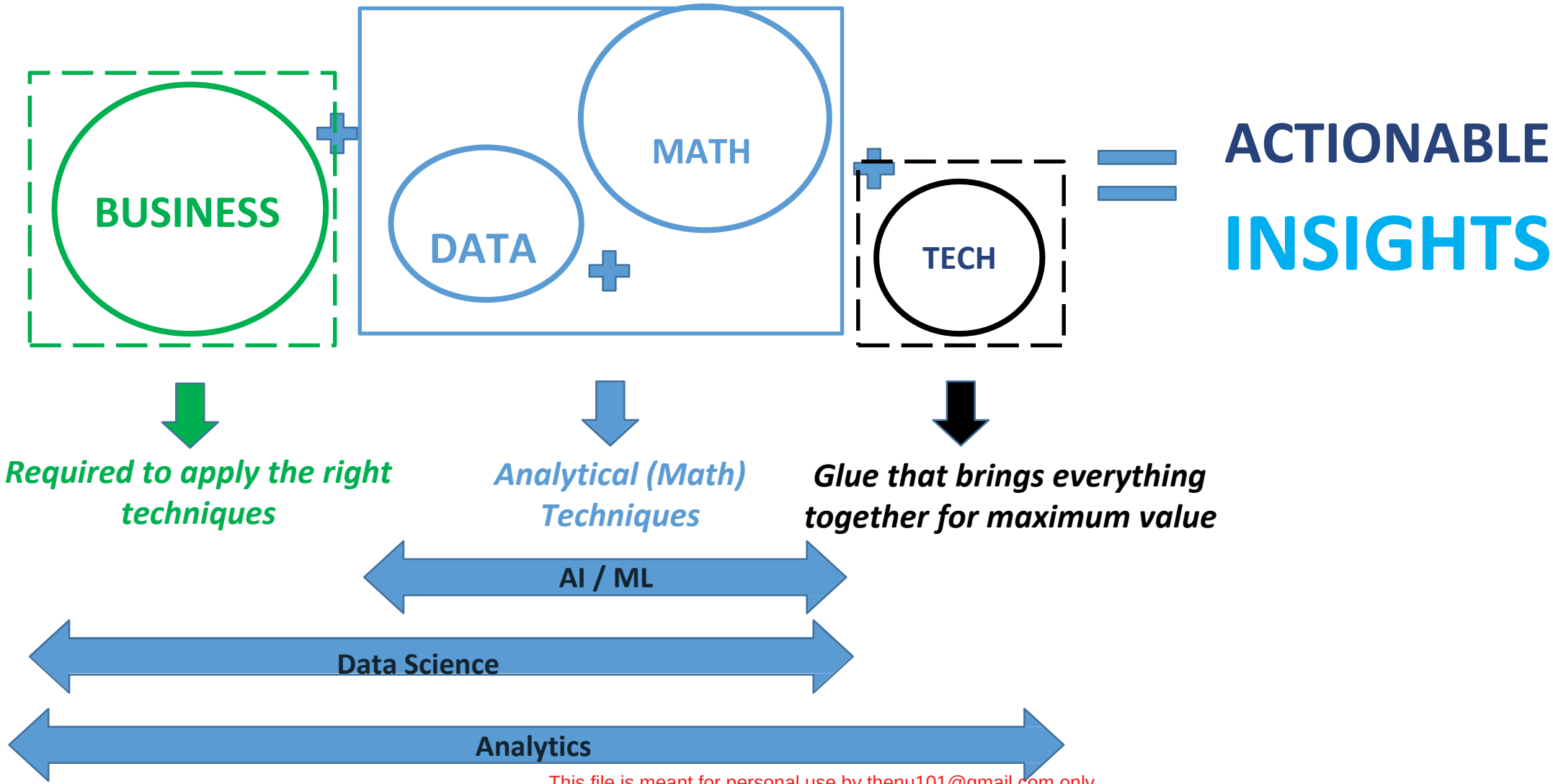
- Uses classes and objects
- Imperative programming

```
class Polynomial:
        # class variable
        Var =1
        def __init__(self, coefficients):
         " input: coefficients are in the form a_n, ...a_1, a_0 "
                self.coefficients =  list(coefficients) # tuple is turned
into a list
        def __repr__(self):
      " method to return the canonical string representation of a
       polynomial. "
                return "Polynomial" +((self.coefficients))
```

What is a good programming language?

- Good run-time performance
- User friendly
  - Easy to read, write, and implement
  - Easy to maintain
- Reliable
- Extensible
- Community support

What is AI and Data Science



Programming languages for AI and Data Science

Python for AI and Data Science

GLOBAL USERS	LEARNING CURVE	PYTHON LIBRARIES	COMPATIBILITY
~ 8.2 MILLION	General purpose, object-oriented and dynamic programming language.	~ 70000	Easily compatible with web, data base servers, db scripting, virtual machines, visualization tools and other popular data analytics tools and frameworks.

Salient features of Python

- Open Source
- High-level programming language and Easy to code
- Object oriented programming
- Huge library base and online support
- Portability

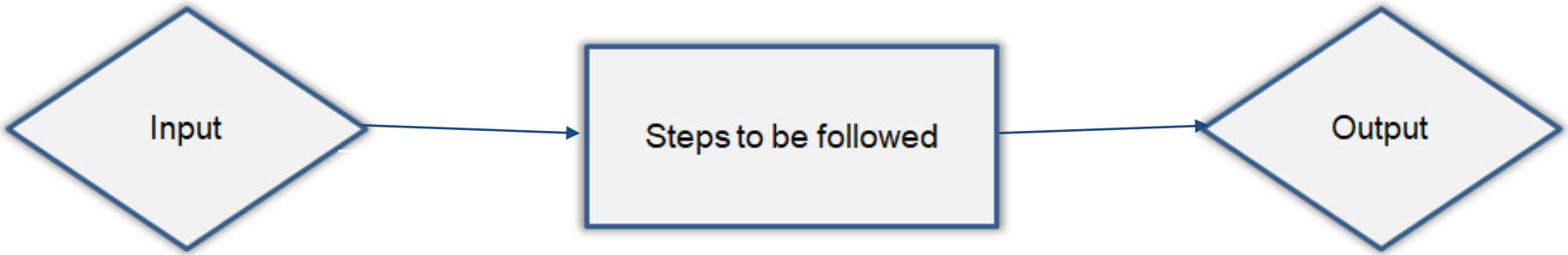
Applications of Python

Python to solve a task

- Using algorithms
- Step-by-step approach

```
A = 23
def isEven(a):
        if(a%2==0):
        print("It is an even number")
    else:
        print("It is an odd number")

isEven(A)
```



Database to dataframe

- Connect with “test_database”
- Select data from the above database
- Fetch the data and store it as dataframe

```
import sqlite3
import pandas as pd
conn = sqlite3.connect('test_database’)
c = conn.cursor()
c.execute(''' SELECT * FROM products '’)
df = pd.DataFrame(c.fetchall(),columns = ['product_id',
'product_name', 'price’])
print(df)
```

Connectivity to Database

- Display number of records fetched

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
          (host = "localhost",\
           user = "root",\
           passwd = "mysql",\
           database="telephonedir")

#creating the cursor object
MyCur = conn.cursor()


qry='Select * from customer'
MyCur.execute(qry)

for x in MyCur: #Fetching result from MyCur one by one
    print(x)
print("Total ",MyCur.rowcount," records present ")
conn.commit()
conn.close()
```

Web Application

- Simple http server
- We need the flask package
- Create a flask app
- Define a function that will be our endpoint
- Add a new route to this function
- Start the server
- Connect to the server through the browser and Python

```
app = Flask(__name__)
def hello():
  return "Hello, World!"
@app.route("/")

FLASK_APP=examples/hello_server.py flask run
```

```
requests.get('http://127.0.0.1:5000/?name=banana').text
```

Machine Learning Application

- Load the libraries required for data preprocessing and model building
- Load the data
- Build a machine learning model
- Train the model
- Make prediction on test data

```
from sklearn.tree import DecisionTreeClassifier

balance_data = pd.read_csv('data',sep= ',', header = None)

clf_gini = DecisionTreeClassifier(criterion = "gini",...)

clf_gini.fit(X_train, y_train)

y_pred = clf_object.predict(X_test)
```

Calling JAVA from Python

```
>>> from jnius import autoclass
>>> autoclass('java.lang.System').out.println('Hello world’)
Hello world
>>> Stack = autoclass('java.util.Stack’)
>>> stack = Stack()
>>> stack.push('hello’)
>>> stack.push('world’)
>>> print stack.pop()
world
>>> print stack.pop()
hello
```

External tools and connectivity with Python

- Database
- Cloud
- Distributed systems
- Web-services
- ML Pipeline

Designing and deploying software products

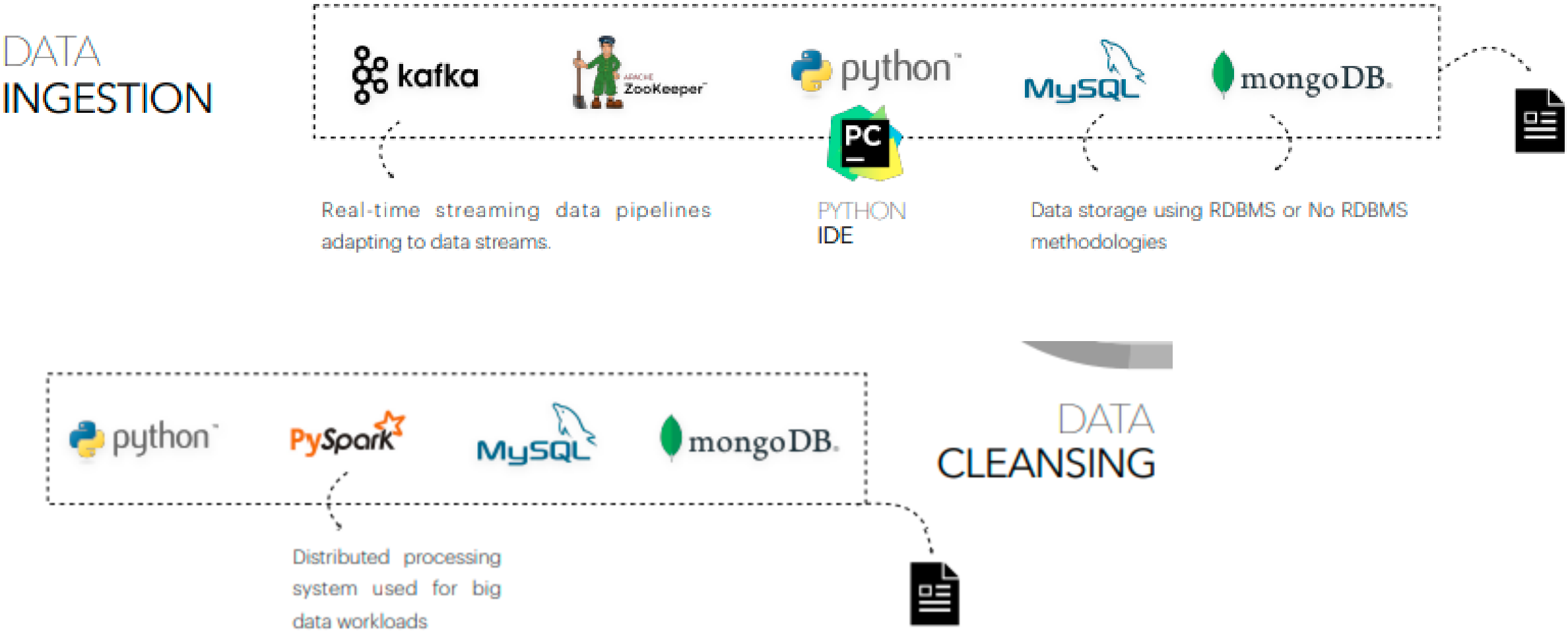
PROBLEM **STATEMENT**

Classify News Articles into categories - With information overload today users are inundated with news articles of all topics, even the ones which may not be relevant to users. Design a system which can classify incoming news articles and appropriately tag the corresponding category. Develop a data pipeline which includes the all the following stages of Machine Learning Project Life Cycle –

1. Data Ingestion
2. Data Preparation
3. Data segregation & Model Training
4. Model Deployment
5. Model Prediction

EXPECTED **OUTCOME**

Designing and deploying software products



Designing and deploying software products

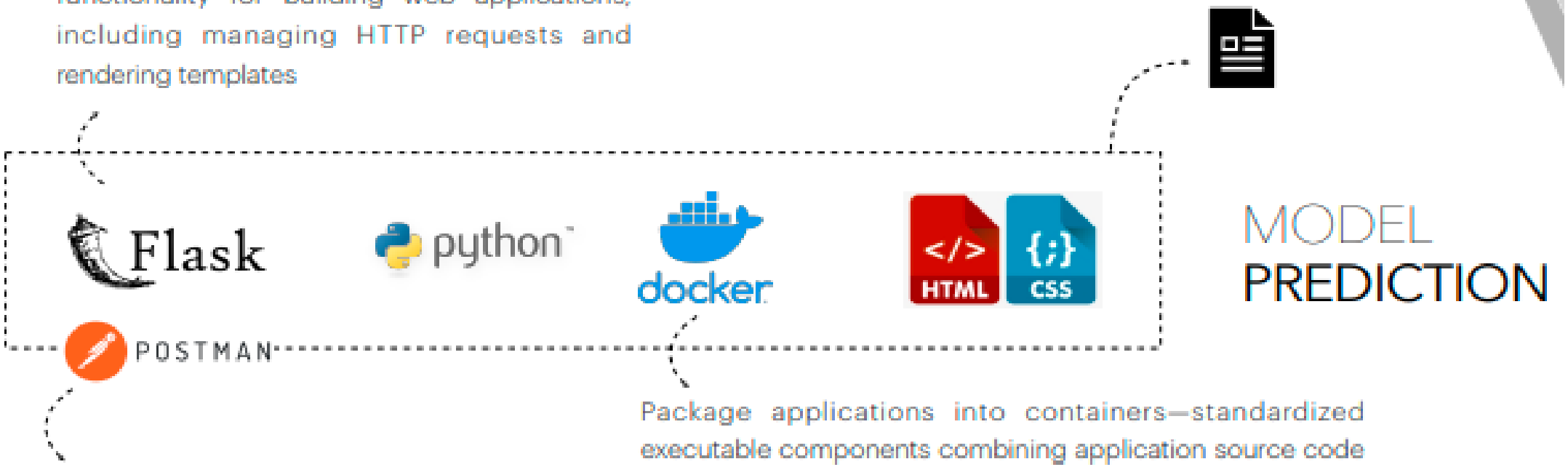
MODEL TRAINING

Building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout.

Framework that supports the machine learning lifecycle. This means that it has components to monitor your model during training and running, ability to store models, load the model in production code and create a pipeline.

Designing and deploying software products

Web framework for Python which provides functionality for building web applications, including managing HTTP requests and rendering templates

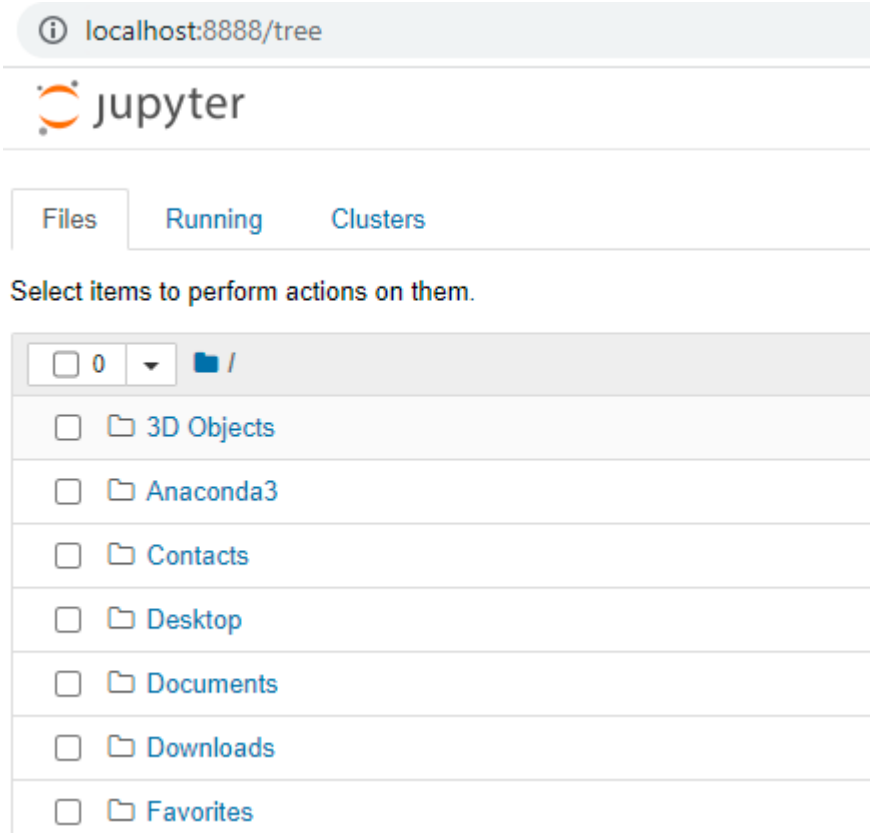


Application used for API testing. It is an HTTP client that tests HTTP requests, utilizing a graphical user interface, through which we obtain different types of responses that need to be subsequently validated.

Package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment.

Next Steps

- Install Python
- Learn basics of python:- variables, data types, data structures, operators, etc
- Learn programming concept:- conditional statements, loops, functions, etc
- Learn common packages
- Explore and experiment
- Play with data and make business decisions



Summary

- Why do we need programming
- Different types of programming languages
- Programming languages for AI & Data Science
- Python for AI & Data Science
- Salient features of Python
- Applications of Python
- External tools & environment
- Python for designing and deploying software products with python

Thank you!

Happy Learning :)