Section 1: Introduction

1. Introduction – What we learn in this course

13 + End to End Programs

1. Python Programming – Basic – Advance
2. NLP in Deep Learning – RNN , LSTM RNN, GRU RNN , Bidrectional RNN, Encoder and Decoder, Attention Mechanism , Transformer, BERT => Basic Building block to understand the LLMS Model -> Generative AI
3. Langchain Ecosystem -> Generative AI framework. => paid LLM , Multimodel , Open Source LLM

Agents, Chat with SQL, Vector Database , Rectrivers , Text Summarization

Chatbots with chatmessage History

{streamlit}

Libraries:

1. Langchaniun core
2. Langchain Community
3. Langchain Tool
4. Deployment Techniques : Free Resouces are available
5. Fine Tuning Techniques -> Google colab <- Quantization LORA and lora -> Fine tune Open Source LLM Models
6. GenAI In AWS: AWS BEDROLK, Lamada function, API Gateway

End to End project -> AWS Sagemaker

1. Nvidia NIM : End to End Project lanchain
2. CREW AI : Multi AI Agent -> Gen AI Apps – Langchain

Crack any job

Startups

LLM and MultiModel

1. Open AI -> Gpt 4-0 , GPT 4 turbo, OpenAI Embedding
2. Google -> Gemma , Gemma 2
3. Meta -> Lamma 3 , Codelama
4. Antropic -> Mistral
5. Hugging Face -> Models

Use all this in the GROQ Infra -> LPU Engine

1. Materials: <https://github.com/krishnaik06/Complete-Python-Bootcamp>
2. Getiing Started with VS Code

When We start a project we need to create a environemt.

Because the packahes can get new features. Which can conflict the code

1. Creating Enviroment:

conda create -p venv python==3.12

syntax:

conda create -p venv python==<version>

1. Some Default packages are installed
2. Activate: conda activate venv
3. New file: app.py
4. python app.py in cmd to execute
5. test.ipynb.

code cell =

Markdown cell =

EXECUTE: SHIFT + ENTER

Options: run all

1. need to pip install ipykernel
2. new file: requriements.txt

add the names of the packages

1. cmd: pip install -r requirements.txt
2. DIfrrent Ways to create the python environments

Way 1: Using the python ways

Create : python -m venv myenv

Activate: myenv\Scripts\activate

Deactivate: deactivate

Which version will be installed ? The installed version will be the

Way 2: pip install virtualenv

Sane as the way 1

Way 3: conda create -p venv python==3.10 -y

Even if the base has the difrrent version

1. Conda not recoginzed: [python - 'Conda' is not recognized as internal or external command - Stack Overflow](https://stackoverflow.com/questions/44515769/conda-is-not-recognized-as-internal-or-external-command)
2. Python Syntax and Semantics
3. Single Line comments and multiline comments
4. Defination of the syntax and sematics
5. Understandinf the symentics in the python
6. Common Errors and how to avoid them
7. Pratical Code Examples

Syntax: Correct arrgment of words and symbols in the code

Sematics: Meaning and purpose of the code

## Basic Syntax Rules in the python – this is the single line comment

## 1. Case Sensitive

‘’’

This is the multiline comment

Welcome to the python course

This not works in the jupier noteowk

‘’’

## Python is case sensitive

Name = “Krish”

name=”Naik”

print(Name)

print(name)

These both variable is difrenr because the python is case sensitive.

## Intentation

## Python uses the indenation to define as the block of statement. Tab

Age = 32

If age>30:

Print(age)

Print(age)

# This is the block of the statemens

## Commetnts

## This is the single line comment

Print(“Hello world”)

## The comments are the quotes used to give the information about the the code

##Line Countation

Total = 1+2+3+4+5+6 +7+\

5+6+10

Print(Total)

#Muliple statements on a single line

X=5; y=10; z=x+y;

Print(z)

# Sematics

# Variable Assignments

Age = 32 # Age is an integer

Name = “Krish” # Name is a string”

## Type Infrence

Type(Age)

## Type Infernce

Variable = 10

Print(type(variable))

Variable = “Krish ”

Print(type(Variable))

Age = 32

If age > 30:

Print(age)

# INTENTATION ERROR

a=b

# Name Error: name b is not defined

## CODE EXAMPLES of indentaion

If True:

Print(“Correct\_Indentation”)

If Flase:

Print(“This not print”)

Print(“This will print”)

Print(“This is the print”

Three times printed

7 Variable

# Declaring and assigning the variable.

a=100

height = 6. 1

name = “Krish Naik”

is\_student = True

##Prinitn the varible

Print(“age: ”, age)

Print(“height”, heigth)

Print(“name: ”, name)

Naming Convertion

1. Variable names should be descriptive
2. They must always start with a letter or a or \_\_
3. They can contain the leyyers , number or a underscore
4. Variables names are case sensitive

## Valid variable name

First\_name = “Krish”

Last\_Name=”Naik”

## Invalid Varibale Name

12age=2

First-name =”Krish”

We will get the syntax error

@name=”Krish”

## Case Sensitivity

Name =”Krish”

Name=”Naik”

Name = name

O/P: False

Variable Types

1. Python is dynamic typed, type of the variable is determined at runtime

Age= 25

The program will understand this is the int in while it execute

Name= “Krish” # Str

Height = 32.34 # flaot

Is\_student=True # bool

Type Checkign

Type(height )

Syntax:

Type(heigth)

Type Convertion

Age = 25

Print(type(age))

<class ‘int’>

Age\_str=str(age)

Type(Age\_str)

<class ‘str’>

Different type convertions:

Str with numbers can be converted to the int

String with etters cannot be converted to the int

ValueError: invalid literal for int() with base 10: ‘Kiish’

Floar-> int : Removthe decimal points

## Dynamic typing:

Varible = 10

Print(Variable , type(var))

10 <class ‘int ’>

## Python allows type of the variable as the program exevutes.

Var = “Hello”

Print(var, type(var))

Hello <class ‘str’>

## Input :

Age = input(“What is the age: ”)

Print(Age, type(age))

Iput: 23

23 <class ‘str’>

## Caluculator

A= flaot(input())

B=float(input())

Sum=A+B

Diff=A-B

Multi= A\*B

Diff=A/B

Print(“SUM”, Sum)

Print(“Diff”, Diff)

Print(“Mulif”, Mulit)

Print(“Diff”, Diff)

Data Types:

1. Definition: Classificarti of data which tells the compiler or intrepretor

It uses for the types of the opertation we need to do.

Examplation:

1. Store Effetively

Int:

Age = 35

Type(35)

Flaot:

Float\_numbr= 23.324

Print(Float\_Number)

Print(type(Float\_Number))

Krish

< Class ‘flaot’>

## Boolanen

Is\_True= True

Type(is\_True)

<class ‘bool’>

Is\_true=bool()

a==b

type(a==b)

True

Commen Erros:

Result = “Hello ” + 5

## TypeError: Can only concardinate

Results = “Hello ” + str(5)

Hello5

Str. <methods>

There are soo many methos here,

Operations:

+ - \* / \*\* // % give definition of all

== !=

>=

<=

!=

>

<

And

Or

Not