Python

- Python is high level language
- Python have OOPS or object
- Python is like English language its easy to read and understand
- Python have intentation in it
- Python have powerfull libraries and frameworks
- Python interpeted language which execute the code line by line
- use friendly, Dynamic typing
- used for web development, data science, AI

- Data type
  - Numeric
  - Boolean
  - set
  - Dictionary
  - Tuple
  - list
  - strings

- Variable :-
  - Variable is used to store and manipulated the data
  - Variable is the name to a memory location that stores a value.

-> list

[ ]

- list have square brackets
- list are mutable which means it can change after its creation.
- list has allocate an extra block of memory for potential growth

list = ["apple", "; ", "banana"]

methods

- pop(), copy(), len()
index(), sort(), insert()
remove(), append(), slicing
indexing()

Tuples

( )

- Tuples have parenthesis
- Tuples are immutable which means they cannot change after creation.
- Tuples memory is efficient it also allocate only the minimum memory required for the data

tuple = ("apple", "; ", "banana")

- len(), index(), count()
in

-> Dictionary

- Dictionary is a ordered collection of key-value
- indexing in dictionary By key-based indexing
- tuples, immutable
- operation of dictionary key based lookup, updates and deletions

my_dict = {'name': 'ram', 'age': 'one'}

print (my_dict)

{'name': 'ram', 'age': 'one'} ordered

Set

- set is a unordered collection of unique element
- set does not support indexing
- set can contain hashable object
- operation in set
- union, intersection

my_set = {1, 1, 3, 4, 5, 6}

print (my_set)

{1, 2, 3, 4, 5, 6}

unordered

How to Remove Duplicates form a list

```
mylist = ["a", "b", "c", "c"]
mylist = list(dict.fromkeys(mylist))
print(mylist)
output :- ["a", "b", "c"]
```

How to reverse a string

As we can reverse a string using negative indexing

```
txt = "Varshith"[::-1]
print(txt)
```

HTIhSRAV

Introduction to web APIs

API :- Application programming interface

it act as a messenger, allowing data to be Exchanged between two or more programs, system or services

web API :- A web API is an interface that enables communication between a web server and client-side applications, through the HTTP protocol

HTTP requests (GET, POST, PUT, DELETE)

Context Model

HTTP : Hypertext transfer protocol is used to exchanging information over the internet

- Request
- Response

HTTP request methods :-

- GET - to Get Data from a resource
- POST - to update at a resource
- PUT - to create data at a resource
- DELETE - to delete data at a resource

Restful api's :- Representational State Transfer is an architectural style for an API that uses HTTP request to access and use data

Client Server Architecture : we distinct roles like Client and Server

Uniform Interface : uses HTTP methods

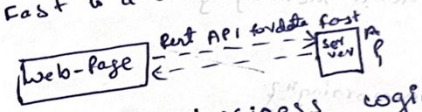
- It provide a flexible, scalable and widely accepted way for different system to communicate and exchange data.

FAST API

FAST API is a web-framework for building modern API with Python

FAST API helps to reduce Bugs, quick & Easy Robust, standards

Fast is a web-frame for building modern RestFul API



Handles all business logic for the application

web framework :- allows a simplified way for rapid development. this include many years of development, which allows you to have a secure and fast application.

Uvicorn :- is an ASGI web server implementation for Python

- a web server. handles network communication.

Path Parameters

- Path Parameters are request parameter that have been attached to the URL
- Path parameter are usually defined as a way to find information based on location.

endpoint :- is a specific location with the API that accepts requests and sends back responses.

Show return Page to the users

The Fastapi in the vscode execution time use this code in terminal

- activate - for activation of virtual environment
- uvicorn main:app --reload
- uvicorn is a server

to get a message in Json formate

return {"message": "Good morning"}

output

{"message": "Good morning"}

for the normal format we use

return 'Good morning'

GET method overview

Path Parameters

In to locate

Path parameters have type validation

- int
- String
- float

Ex:

```
@app.get('/blog/{id}')
def get_blog(id: int):
    return {'message': f'Blog with id {id}'}
```

localhost:8000/blog/3

```
{
  "message": "Blog with id 3"
}
```

Query Parameter

Page and page_size

/all? page=2 & page_size=10

Page and Page_size are Divided the '&'

api. route

Syntax: api_route

to create a single pathway

by using a route api we can save time and by creating only one route we can solve stop creating different methods.

->multiple route, and generick route

Pydantic

Pydantic is a library for datavalidation and modeling

we use pydantic for "body" not for the "header"

curl : curl is an command line tool that enables data transfer over various network protocols communicates with a web or application.

=> The Fast api we use "union" type for accepting both "str" and "int"

syntax : union [str, int]

Panda's -

Dependency injection :- Giving an object its instance variables.

SELECT SQL QUERIES

C - command E - English

1) SELECT * FROM todos; - C

* Select ALL columns and rows

2) SELECT title FROM todos; - C

E - Select just title from colums

3) Select title, description FROM colums - E

C - SELECT title, description FROM todos;

- from Fastapi import FastAPI

This part is importing form FastAPI library

FastAPI is a package

- app = FASTAPI()

This is a variable that holds the instance of FastAPI

creating an instance of FastAPI class

- @app.get("/")

root URL

request

WHERE SQL QUERIES

SELECT * FROM todos WHERE priority = 5;

" * " title = 'Feed dog';

UPDATE

UPDATE todos SET complete = TRUE WHERE id = 5;

DELETE

DELETE FROM todos WHERE id = 5;

DELETE FROM todos WHERE complete = 0;