**PYTHON:**

Is high level ,dynamic programming language. It is easy and simple to learn and also interpreted and prototyping. Python is an imperative language, is basically open source. It has objects, modules, threads, exceptions and automatic memory management.

**Pickling and Unpickling:**

In python data is in the form of modules called pickles. Pickle accepts any python object and converts into a string representation and dumps into a file using dump function, this process is known as pickling and where the process of retrieving objects from string representation is unpickling.

**How Pyhton is Interpreted?**

Python program runs directly from the source code. It converts source code to intermediate language then translated to machine language has to be executed.

**Memory management**

In python memory is managed by private heaps. All objects and data structures are located in private heap space.

Space allocation for objects in heap will take care by memory manager. It has inbuilt garbage collector, which recycle all the unused memory and makes it available for heap.

**Decorators:**

Decorators change syntax to alter functions. We can extend a function without modifying the behavior of current function.

**Data Types in Python:**

Mutable -Lists,Sets,Dictionaries.

Immutable – Strings,Tuples,numbers.

Operators: Arithematic,Relational, Logical , Assignment , Membership and Identical Operators.

Web Development:

Git is a tool, Github manages our source code.

Git is a version control system which enables us to track changes to fields.

Git creates .git folder to store the details of the file system.

Frameworks:

Django vs Flask

Flask provides simplicity and flexibility it supports all packages to build applications.

While Django has its own object relational mapping (ORM), it is used for complex and end to end applications.

**PIP:** puthon package index

It is a python package manager which is used to install all new packages.

**HTML**

HTML is the standard markup language for creating Web pages.

HTML describes the structure of Web pages using markup

HTML elements are the building blocks of HTML pages

HTML elements are represented by tags

<head><body><title>

CSS:

**CSS** stands for **C**ascading **S**tyle **S**heets

CSS describes **how HTML elements are to be displayed on screen**

CSS is used to define styles for your web pages, including the design, layout

and variations in display for different devices and screen sizes.

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

SQL:

SQL stands for Structured Query Language

SQL lets you access and manipulate databases

SQL can execute queries against a database, retrieve data from a database , insert records in a database,update records in a database, delete records from a database, create new databases and can create new tables in a database.

SQL commands:

Select,update,delete,insert into,create database,alter database,order by etc.,

SQL Joins: inner join,outer join,left join,right join

Client – Server Architecture:

Consumer who wants to consume a specific information called as client where as provider who provides the informations is known as server.

HTTP Protocol:

A medium for communication ,specifically a protocol for two systems to interact.

A protocol to ask for the required details from the server,this could be in any form of formatted data most commonly used formats are XML and JSON

HTTP is a required response protocol to communicate asynchronously between client and server.

Mostly in HTTP a browser acts as a client and web-server like Apache acts as server

Examples: GET,POST,PUT,PATCH,DELETE

API(Application Programming Interfsce)

Is a Software intermediary that allows two applications to talk to each other.

**SOAP&REST:**

SOAP(Simple Object Access Protocol) and REST(Representational State Transfer) are Web services access protocols.

SOAP is a standards-based Web services access protocol, where REST is seeks to fix the problems with SOAP and provide a truly simple method of accessing Web services.

SOAP vs REST:

**SOAP:**

It is definitely the heavyweight choice for Web service access.

Language, platform, and transport independent where REST requires use of HTTP.

Works well in distributed enterprise environments where REST assumes direct point-to-point communication.

It is Standardized.

It Provides significant pre-build extensibility in the form of the WS\* standards

Built-in error handling

**REST:**

No expensive tools require to interact with the Web service

Smaller learning curve.

Efficient where SOAP uses XML for all messages, REST can use smaller message formats.

Fast (no extensive processing required)

Closer to other Web technologies in design philosophy

REST is easier to use for the most part and is more flexible.