MERN Stack Training / Web Full Stack / Javascript Full Stack

Duration: 140 hours

MongoDB: NoSQL Database to maintain the data

Express.js: Backend services

React.js: Frontend

Node.js: Runtime environment

Contents:

* Applications
* Fundamentals of Computer Programming
* Algorithms & Pseudocode
* Linux
* Git & Git Hub
* HTML
* CSS
* Javascript & Typescript
* React.js
* Node.js
* Express.js (REST Api’s)
* MongoDB
* DevOps

Applications: They are computer programs which can perform tasks for the users, there are 2 types of applications

1. Standalone
2. Distributed

Standalone applications: These are applications which you can used only after running in the machine

Two types

* Desktop based: This can be installed on your desktop
* Mobile based: This can be installed on your mobiles

ex: Word document, Mobile based applications

Distributed applications: These are the applications which you can access over the internet

ex: Banking services, Twitter, Facebook, Gmail, Food Ordering Systems, Shopping cart applications

How do we access distributed applications

We can access them using Browser or Client applications

Browser: It is an application which helps to enter URL and use the distributed application

Client applications: These are the applications that can internally use the URL to access distributed application

ex: Mobile applications, ATM machine programs, Swiping Machines programs

URL: Uninform Resource Locator, it is a name given to the applications to easily access it, in real time the applications use IP address & Port numbers, but users can’t remember it hence, URL will be given

Browser:

Browser is an application which helps to access any websites through URL, there are many browsers

1. Google Chrome
2. Edge
3. Mozilla
4. Opera
5. Older browsers like
   1. World wide web browser
   2. Mosaic
   3. Internet explorer

Distributed Applications:

These are the applications that can be accessed over the internet, these applications will be running in some remote machines, there are mainly 2 types of Distributed applications

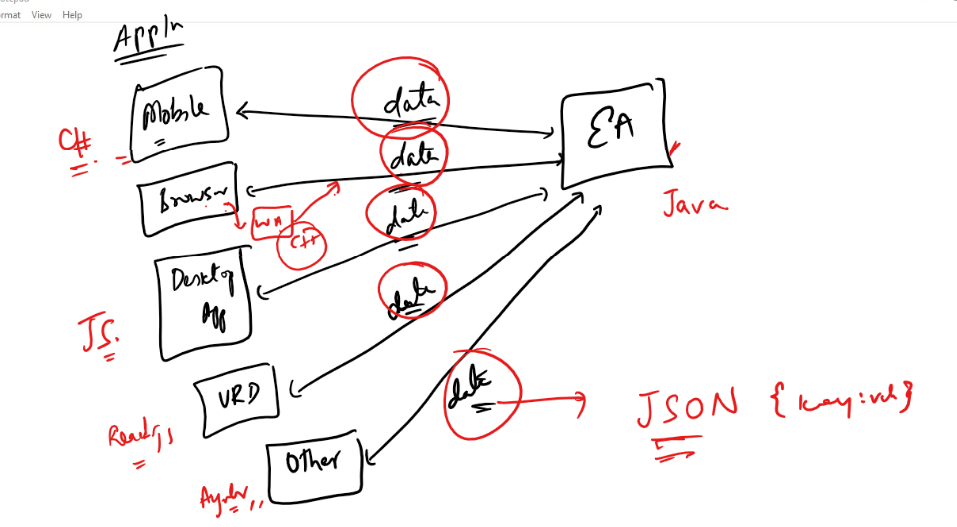
1. Web applications
2. Enterprise wide applications

Web application:

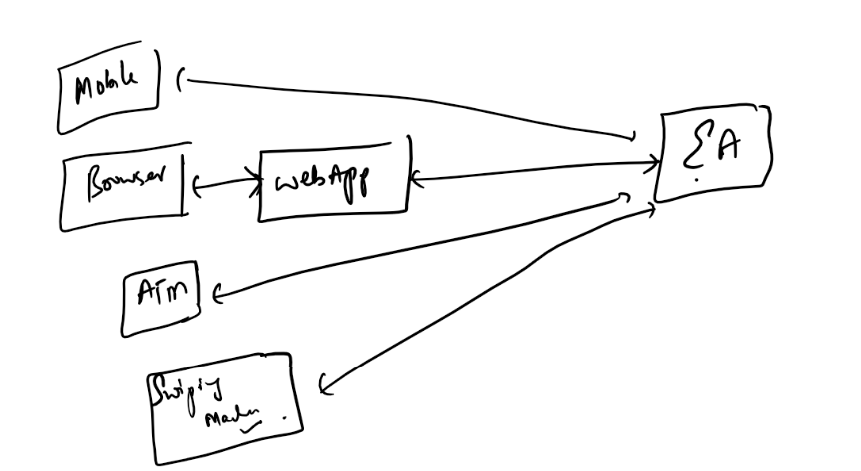
A web application takes request from the browser & returns the response in the format the browser understands i.e., HTML, CSS & Javascript

Enterprise application:

These are business related applications which can serve various types of applications written in different languages.



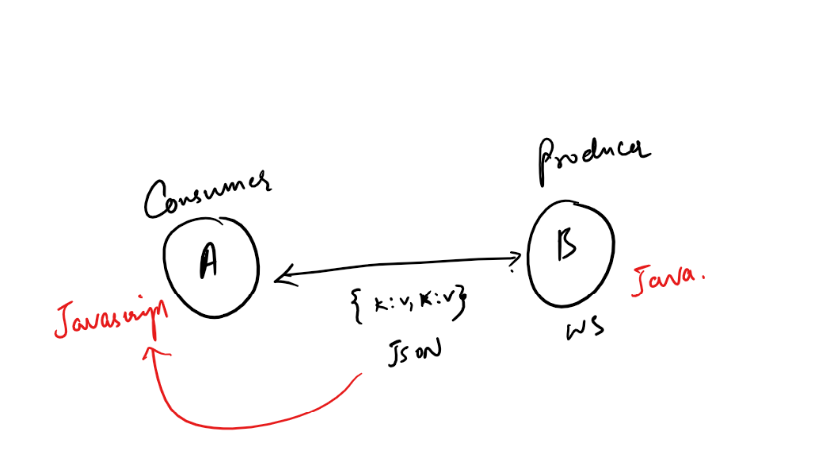
The enterprise application return data in a format which every application can convert to their respective types, suppose Java wants to share data to C#, then Java wouldn’t share data in Java instead it shares in one common format JSON that will be converted to C# by C# applications, same can be converted to Javascript by Javascript applications, same can be converted to Python by Python applications.



How does these enterprise applications share/exchange data to other applications

Enterprise applications will have webservices (ReST based Webservices) to share the data to any applications.

REST WebServices: These are online services or API’s which are made available on the internet so that any applications can use their URLs to consume the data, usually data will be in JSON format which is converted to the format the consuming application is written in



What are the common formats the webservices can use to share the data

1. JSON (More widely used): Easy to understand & maintain hence its used widely almost 99.9% of the time
2. XML
3. TEXT
4. HTML

Web applications: These applications will have web pages, there are 2 types of web pages

1. Static web pages: These are the pages which are having same contents for all the users, it doesn’t change but its common for all the users

ex: Wikipedia

1. Dynamic web pages: These are the pages whose contents are different for different users, these pages change their contents at runtime

ex: Gmail pages, Facebook pages and so on

Technologies/Languages used to create Static & Dynamic web pages

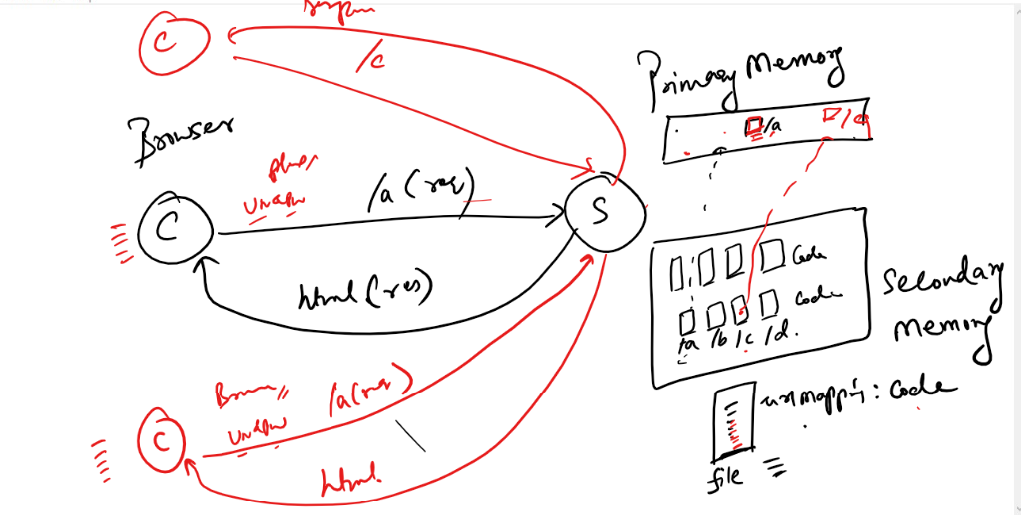
Static pages are created with below languages

1. HTML
2. CSS
3. Javascript

Dynamic pages are created with various technologies

1. C#: ASP .NET
2. Java: Servlets & JSP
3. Python: Django
4. Javascript: React.js, Angular.js, Ember.js

How the dynamic pages are created



Here when client sends the request to the server, it sends the URL to the server, the server is going first check in the primary memory whether any instance/entity/object can handle the request if the entity is not present then server checks in some configuration file to know which is the entity to handle the URL, it loads that entity to the primary memory & it executes the code to generate response, here response will be HTML because client is browser, however it can generate different HTML response for different clients for same URL.

ex: If you login to gmail then you see different page and others see different page.

What is the language browser understands

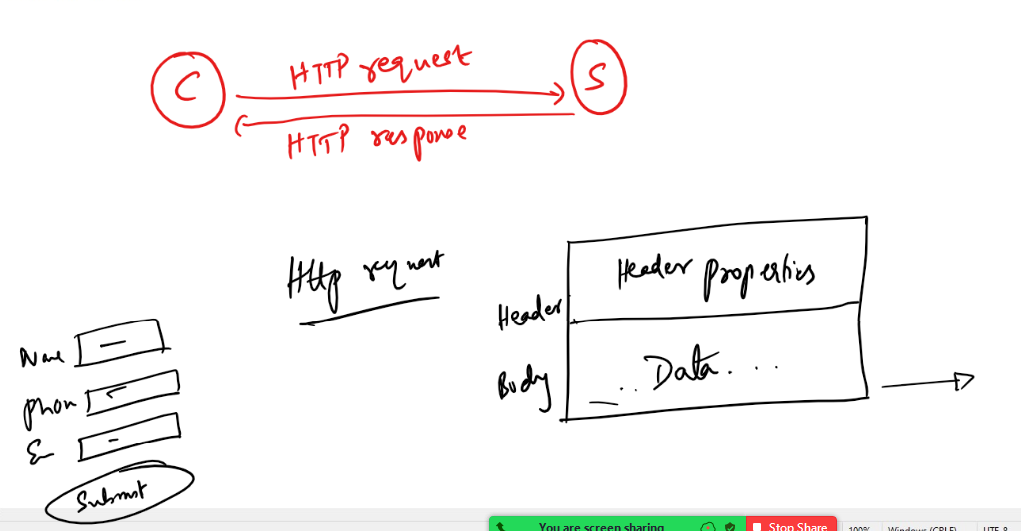
1. HTML
2. CSS
3. Javascript

What is the language client & server uses to communicate

HTTP: Stands for Hyper Text Transfer Protocol, it is the language client & server uses to communicate, they communicate using request & response, since they use HTTP they are also called as HTTP request & HTTP response.

Architecture of HTTP request & HTTP response

HTTP Request



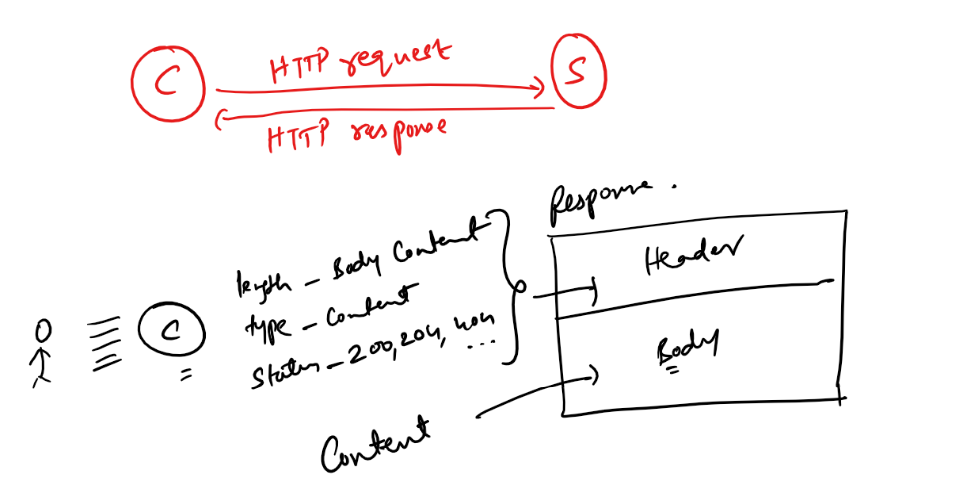
When client sends the request an HTTP request will be sent, the HTTP request will be divided into two sections

1. Header
2. Body

Header: It will have header properties like URL of the server resource, content type like text, json, xml and so on

Body: it will have content/data

HTTP Response



The Response will have 2 sections header & body

Header of the response will have header properties of response like content type, content length, time take to produce response, status codes like (200, 404, 405) and so on

Note: status code plays an important role for the client programs, because client programs can understand whether the request is succeeded or failed.

If status codes are in 2xx series then its success

If status codes are in 4xx series then its failed

HTTP status codes

These are some standard codes for client & servers used while communicating so that they can make decisions based on the status code

200: Ok

201: Created

404: Resource not found

401: Unauthorized access

400: Bad request

Application design patterns

Design Pattern:

It is a solution to the commonly occurring problems in application development, there are many design patterns in the IT industry which suits for certain kind of requirements, some of the most widely used design patterns are:

1. Singleton pattern
2. MVC pattern

Singleton pattern: It is a design pattern which creates one & only instance to serve some tasks, even if you try to create multiple instances it gives you single instance

ex: Task manager of an OS doesn’t create multiple copies of it, instead same copy of task manager is used

ex: Database Connections can be singleton so that multiple instances of the application can use same database connections

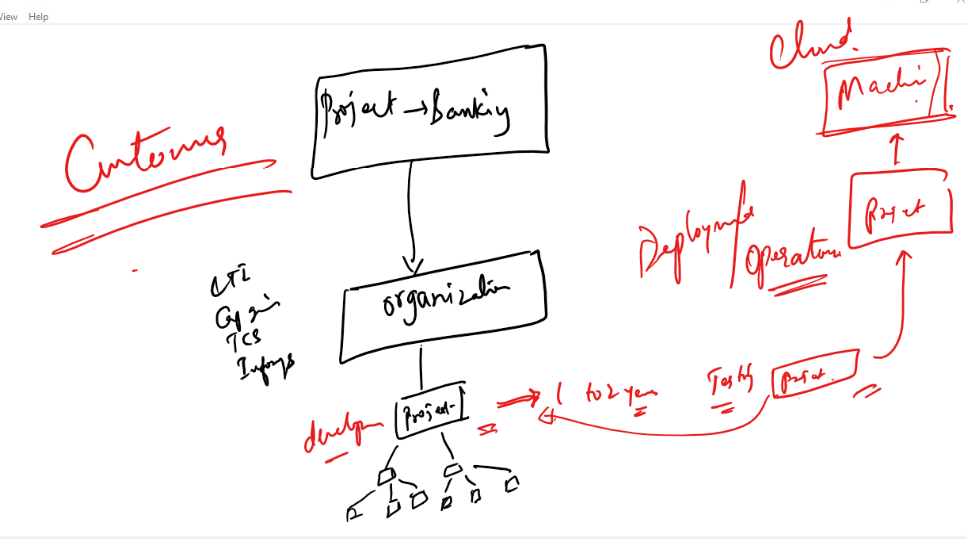
MVC pattern: MVC stands for Model View Controller, it is one of most widely used design pattern in many of the applications almost all the applications follow it, because it separates the roles of the code so that it will be easy to maintain and develop.

This makes application to easily adapt for the changes and development will be faster because you will have different teams for different roles like one team for only view layer, one team for only controller layer and another team for only model layer.

What type of application we will develop in Full Stack

A fullstack developer needs to know all the phases (developing, testing, deploying) of the application development, full stack deals with two types of applications

1. Front end applications
2. Back end applications



Backend applications:

These are the backend services which are available online to access from any kind of applications normally they are RESTful webservices, they will have database connectivity, business logics, data processing logics and so on.

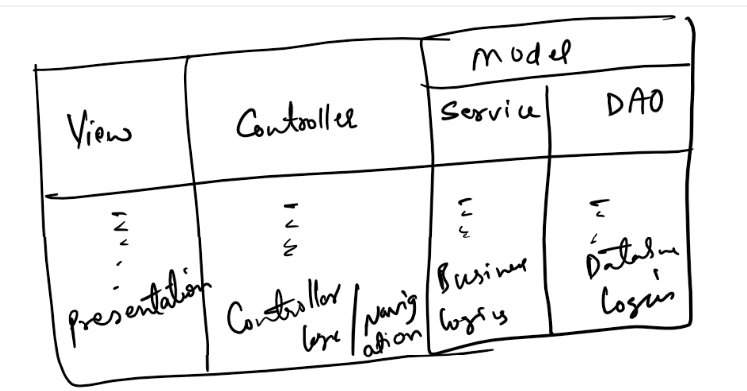
Database logics will take care of performing operations on the database, normally these operations are called as CRUD operations(Create Retrieve Update Delete), However business logics will take care of performing business specific actions.

Business logics:

These are the logics which are performed before database logics, like

1. checking for balance before debiting the amount, else respond with insufficient amount
2. Checking for ID card validity to process the user details
3. In Facebook you need to wait for 24 hours to again delete your account if in case previously you have deleted and cancelled it

When we follow MVC architecture you will keep in business logics & database logics in model layer as below:-



We have many technologies to implement the backend services like

* Java
* C#
* Python
* Javascript (node.js & express.js)

Front End applications:

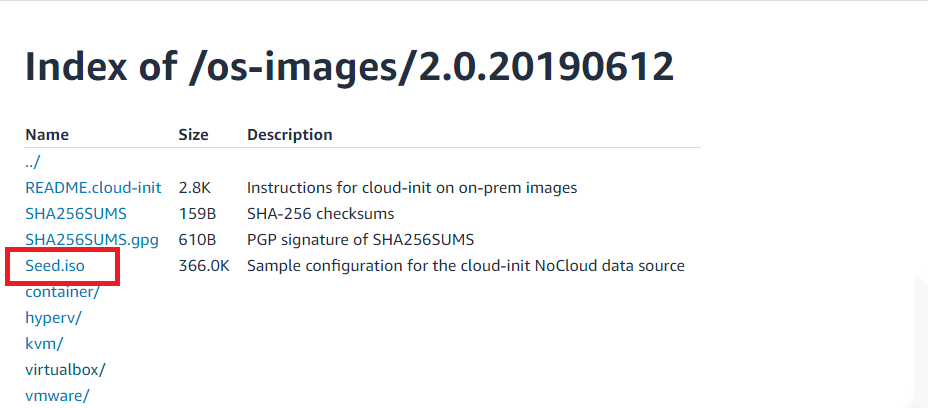
These are the applications which can be run on different devices they will have only UI’s it could be Mobile, Web Browser, Different devices like Watch, VRH, GPS devices, swiping machines, they can be developed in various technologies like

* Java for android devices
* React.js for many mobile applications
* Angular Framework
* Ember.js
* Javascript

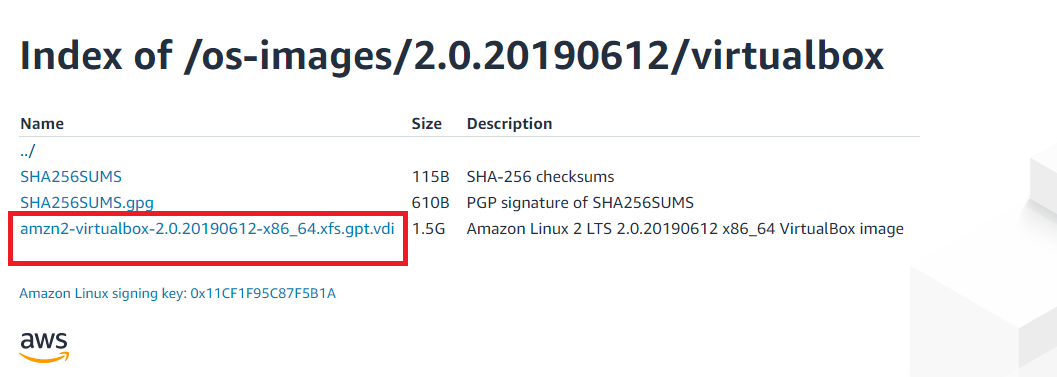
Installing Linux

* You can install it using virtual box, it is not a base machine
* You need to get two files from the below website

<https://cdn.amazonlinux.com/os-images/2.0.20190612/>



Click on the virtualbox folder



The above highlighted is an image for Virtual box that helps us to create a virtual machine to install linux OS

Lastly you need to download & install Virtual Box software

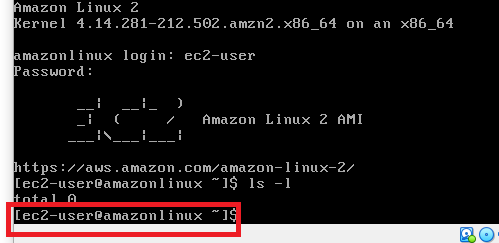


Steps to Install Linux in base machine using virtual box

1. Firstly we need to Virtual Box
2. Secondly we need to open Virtual Box & install Linux OS

Once you login to the linux machine you will see below message as EC2 Amaxon Linux 2

The highlighted part is the location of the linux command prompt, i.e.,   
[ec2-user@amazonlin]$



Linux commands plays an important role in our course

* Git
* React & Nod.js: When you use terminals to launch react & node applications
* Deployment

Evolution Programming Languages

Computer understands Machine Level Language, when it comes to language evolution we have

1. Machine Level Language: Hard to understand
2. Assembly Level Language: Works only with certain processors
3. C: Procedure Oriented Programming Language
4. C++: Object Oriented Programming Language & Platform dependent
5. Java: Object Oriented Programming Language & Platform Independent
6. Javascript: We can use it to develop both frontend and backend applications

How to create applications

Applications are created with programs, these programs will have lot of rules in the form syntax to make compilers & machines to understand, we need to follow those rules to make the application work, else we get errors

We must able to write the code properly, but for that as a beginner we must know 3 things before writing the code

1. Algorithms
2. Flowchart
3. Pseudocode

Before writing the code / program we must know how to solve the problems, hence we have algorithms, flowcharts & pseudocode

All these three are independent of the programming languages as they are not understood by any compilers or machines, they are steps given for humans to understand what is the solution for the problem & use those steps they can write the program.

Algorithms:

* It is independent from the language
* It is a step by step procedure to solve a given problem to get the desired result
* It is not understood by computers
* It gives steps in English statements

Flowcharts:

* It is independent from the language
* It is to understand the flow of the problem solution to better understand the Algorithms
* It gives us pictorial representations of the algorithms
* It is not understood by computers

Pseudocode:

* It is independent form the language
* It mimics programming languages
* It gives us the idea what all the common programming constructs you have to use to solve the problem like conditions, loops
* It is not understood by computers

Note: There is no standard rules about how we should write Algorithms/FlowCharts/Pseudocode, we can write them however we want to understand properly

Note: Once we become good at coding, we may not need these Algorithms, Flowcharts, Pseudocode

Let us write all 3 to solve a simple problem

Problem: Find the area of a circle & print the area

Algorithms

Step1: Start

Step2: Take the radius

Step3: Take the PI

Step4: Multiply radius with radius once

Step5: Multiply PI with the result you get from Step4

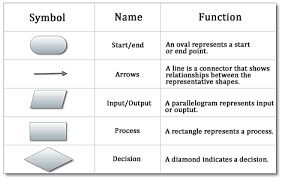
Step6: Take the result you get from Step5 and store in a variable area

Step7: Print the area

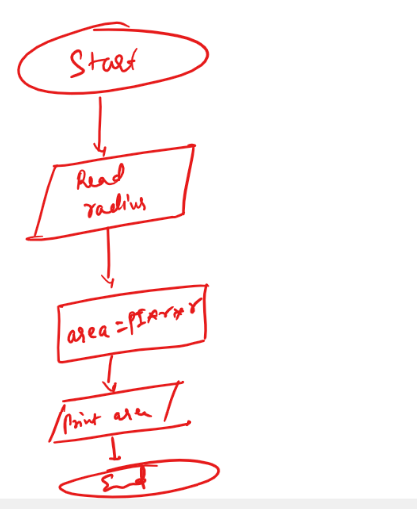
Step8: Stop

Flow charts

Flow chart has some defined symbols



Flowchart for the area of circle



Pseudocode:

It mimics the programming language, the pseudocode for the area of circle

begin

read radius, area

set radius=value

calculate area = pi \* radius \* radius

print area;

end

Simple Activity:

Find the highest number in the two numbers

i.e., which is greater number

Write Algorithms, Flowcharts & Pseudocode for it.

Duration: 10 mins