# Day 1

**Basics:**

* Like window, there is global object in node called ‘global’.
* \_\_dirname: it give the absolute path of current file without a file name
* \_\_filename: it give the absolute path of current file with the file name at the end.

**Module and require**

* To export something to be used in another file we need to export which furthur get attached to the variable in which that was imported to
  + module.exports = { name, ages };
* we can import the module using
  + const varname=require(path)
* Or for importing ant module we can make use of destructuring functionality of the javascript.

**The File system**

* node provide a inbuilt module for file system called “fs”.
* Const fs=require(“fs”) #importing fs module
* For reading file:
  + Use readFile(path,callback(err,data))

//readFile method is async function

fs.readFile("blogs/blog1.txt", (err, data) => {

  if (err) {

    console.log(err);

  }

  //toString method is used to extract actual data from the buffer

  console.log(data.toString());

});

* For writing a file we make use of writeFile method:
  + writeFile(path,dataToWrite,callback(err))

//if we are writing into the file which does not exist then it will crate a new file and write the data into automatically.

fs.writeFile("blogs/blog1.txt", "hello everyone", (err) => {

  if (err) console.log(err);

  console.log("written succesfully");

});

* **delete a file**

//to delete the file we can use unlink method

if (fs.existsSync("./blogs/deleteme.txt")) {

    fs.unlink("./blogs/deleteme.txt", (err) => {

      if (err) console.log(err);

      else console.log("file deleted");

    });

  }

* **create and delete directory**

//Before creating any folder we check whether that concern folder already exists or not

if (!fs.existsSync("./assets")) {

  //we can create any folder using mkdir method

  //if that directory already exists then it will throw some error

  fs.mkdir("./assets", (err) => {

    if (err) console.log(err);

    else console.log("folder created");

  });

} else {

  //To delete any directory, we can use rmdir method

  fs.rmdir("./assets", (err) => {

    if (err) console.log(err);

    else console.log("Folder deleted");

  });

}

* **Streams**
  + **Using streams we can pass the data like a small buffers.**

//creating a readstream

const readstream = fs.createReadStream("./blogs/blog2.txt", {

  encoding: "utf8",

});

//creating a write stream

//this will create new file if that file does not exist.

const writestream = fs.createWriteStream("./blogs/blogs3.txt");

//reading data using read stream

//on is event listener which get triggered everytime new cgunks of data will arrived

readstream.on("data", (chunk) => {

  console.log("------New Chunks---------");

  console.log(chunk);

  //write file using write stream

  writestream.write(chunk);

});

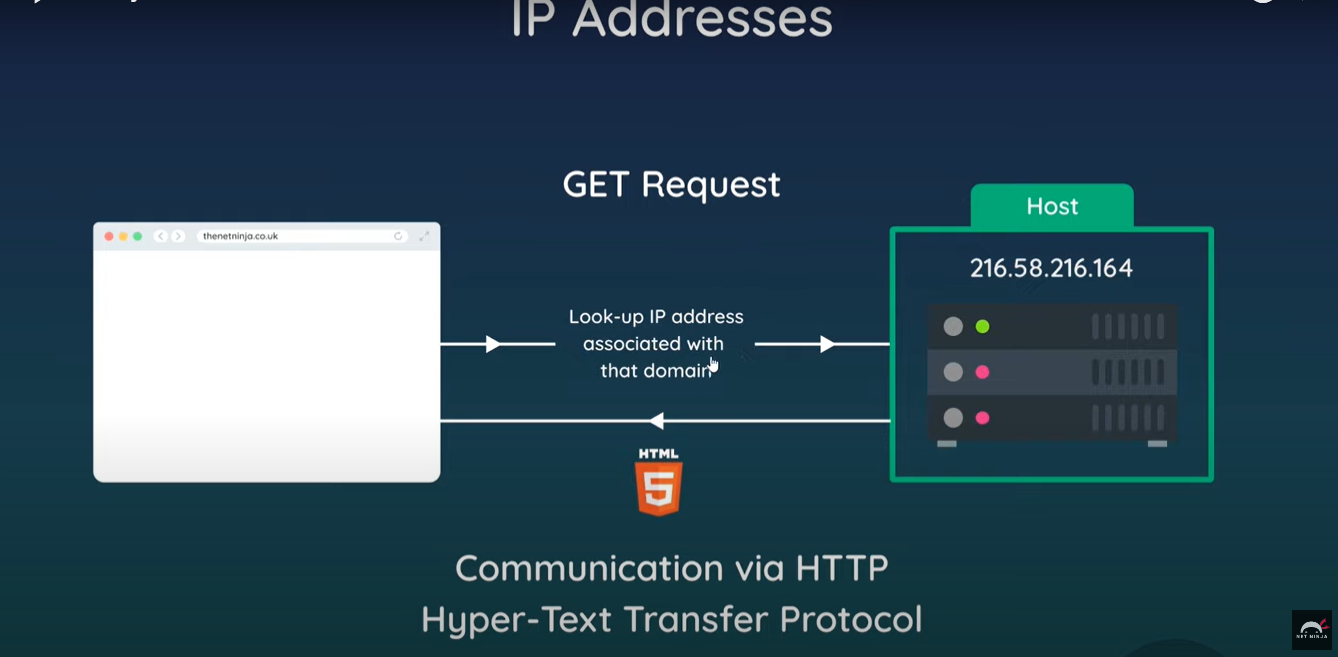
//Piping

//we can copy data from readstream to write stream using pipe method

readstream.pipe(writestream);

**HTTP module**

* How website works?



* **Creating and starting a server**

const http = require("http");

//Creating Server

//createServer method is used create server and takes two arguments reques and response

//request: gives the information about the URL from where the request has been made

//respose: what reponse to give back to that URL

const server = http.createServer((req, res) => {

  console.log("request made");

});

//To start the server we have to start listening to the request,

//we can do that by using listen method come with the server object

//listen method takes in 3 arguments portnumber,hostname,callbackfunction

//callback function get fired when we first stred listening to the request

server.listen(3000, "localhost", () => {

  console.log("litening to the localhost on port 3000");

});

* **Sending response back to front end**

const server = http.createServer((req, res) => {

  //sending back the res

  //1. set the header of the reposnse which tell the browser which type of data is being sent

  res.setHeader("Content-Type", "text/plain");

  //2.write the reposne object with our content

  res.write("Hello from server");

  //3. ending the message to send the data

  res.end();

});

* **Routing**