General:

* Browser makes request to the server and receive HTML,Css, Javascript
* To get the IP a site Command Tracert Name of the site
* In Development Tool : Element = Html, Style = Css , Console= Js

HTML

# Tag:

* <h1>Header1h1>
* <p>paragraph tag</p> : paragraph will start every paragraph in new line
* <b>Bold</b> : it will make the Font Bold
* <strong>Bold also </strong> : it will make also bold
* <ol>OrderList </ol> : order list
* <ul> unorder list </ul> : unorder list
* <br> : inserting Break
* <hr> : horiziontal Break
* <a href=““ alt=““ width=““ lenght=““>senstnse </a>
* Form Tag

## In general all the attribute should have name and value

## Method: Form could be get means the data will be written in the url or post which mean the data will be hidden

## Action: decide which function will be invoked when the submit button us clicked

## Radio: the name will be the same for all the radio button but the value will be different

## Select: will have name in the select and value for each option

## Check box: does not have name

CSS:

# General:

* The file should be .css
* It will be in the head part : <Link type=text/css rel:”stylesheet” href=””
* CSS: Cascade styple sheet
* Recommended sources : scc.tricks website
* If u add 2 Classes to the same element which matter is the order of the class in the css file not in the html file it has no effect which is right and which is left

# Selectors:

* .class : to select class
* #id: to select id
* H1,p : to select all header and paragraph
* Elemnet1 element2: all elements 2 inside element 1
* Element1 > element2: all element 2 which has element 1 as parent
* Element1 + element2 : element 2 has immediate neighbour

# Properties‑:

* Color : red;
* Text-alighn: center;
* List-style: none ; -> delete the point
* Display : block or in-lineblock
* Color: rgba(255,255,255,1) the last element is transparent
* Cursor: pointer
* Pixel: 20 px or 2em -> em is referring to the container element
* Image= ur(“”)
* For image u can use float so u can write some words near the image
* Hover: what will happen when the mouse reaches the element

# Flexbox:

* To activate flexbox choice display :flex
* Define container:

## Flex-wrap:wrap to make the website responsivce

## Justify-content:center,space-between,space-around

## Align-item:center, flix-end, flix-start

* Item : flex-schrink and flex-grow

# Bootstrap:

* Reference the bootstrap through link from bootstrap website
* Link the css ain the header and the css in the bottom
* U can adjust the element by adding attribute of the last element

JavaScript:

# Dom:

* Dom is object element model

# Selectors:

* Document.getElementByid()
* Document.GetElementByclass()
* Document.getElementByClass()
* Document.parentElement
* Document.childElement
* U can get the attribute on a specific element by classing getattribute on the element
* U can set the attribute by calling set attribute
* to set a class call className on the element
* to add a class call classlist.add or remover or toggle

Creating Element:

* document.CreateElement(“li”)
* document.CreateTextNode(“fsdfs”)
* li.AppendChild()

# EvevntListeneer:

* it is easy there is not a lot to say

# Promise

* to create promise u have to initialize the promise using the class new promise((res,rej)=>{}
* the output of the promise can be sent by passing it to res or rej
* to fire the promise call .then on the instance of the promise
* in case of failure call .catch on the instance
* promise can be used in the cases where u usually implement call back function
* **it is very important to know that the promise will always return promise so u can nested the promise so easily**

# HTTP:

* request

## get: the data will be in the url

## post : the data will be in the body of the request

## put: update the data

## delete: delete the data

* Response:

## 304: not changed

## 404 : Not found

## 200: ok

Node express:

NPM:

* The first step is to call npm init to create package.json
* Install nodemon as Dev depenancy : npm install nodemon –save-dev
* In the package.jspn write start: nodemon app.js
* If u have the package and u want to install all the depenacy u can write npm install and the depancay will be installed

# Server:

* To create server u have to import http model
* On http instance call createServer function which takes response and and request , the createServer function should be assigned to a const which is server
* On sever object call listen function and pass the port number the second argument can be the host name the default is localhost

# Express:

* Import express using require after installing it
* Invoked express in einem variables
* Calling listen on this variable

# Exporting module:

* It is important to export using module.exports.router=router
* To send file u use path.join\_\_ dirname which to refers to the file in which the routing happeninh
* U can use path.dirname(process.mainmodule.filename) to refer to the app and then call all the file from this point.