**Learn To Build a Real Time Website Like StackOverflow- MERN Stack**

**Full Stack Technology:**

It is the entire depth of a computer system application.

A full stack developer deals with two separate web development domains:

1. Front end
2. Back end

Front end developers work to optimize the visible parts of an application for web browsers and mobile devices.

Back end developers write the scripts for the server-side.

MERN stack 🡪 MongoDB, ExpressJS, ReactJS, NodeJS

Express and Node make up the middle (application) tier, that is the server-side.

ExpressJs is the server-side web framework.

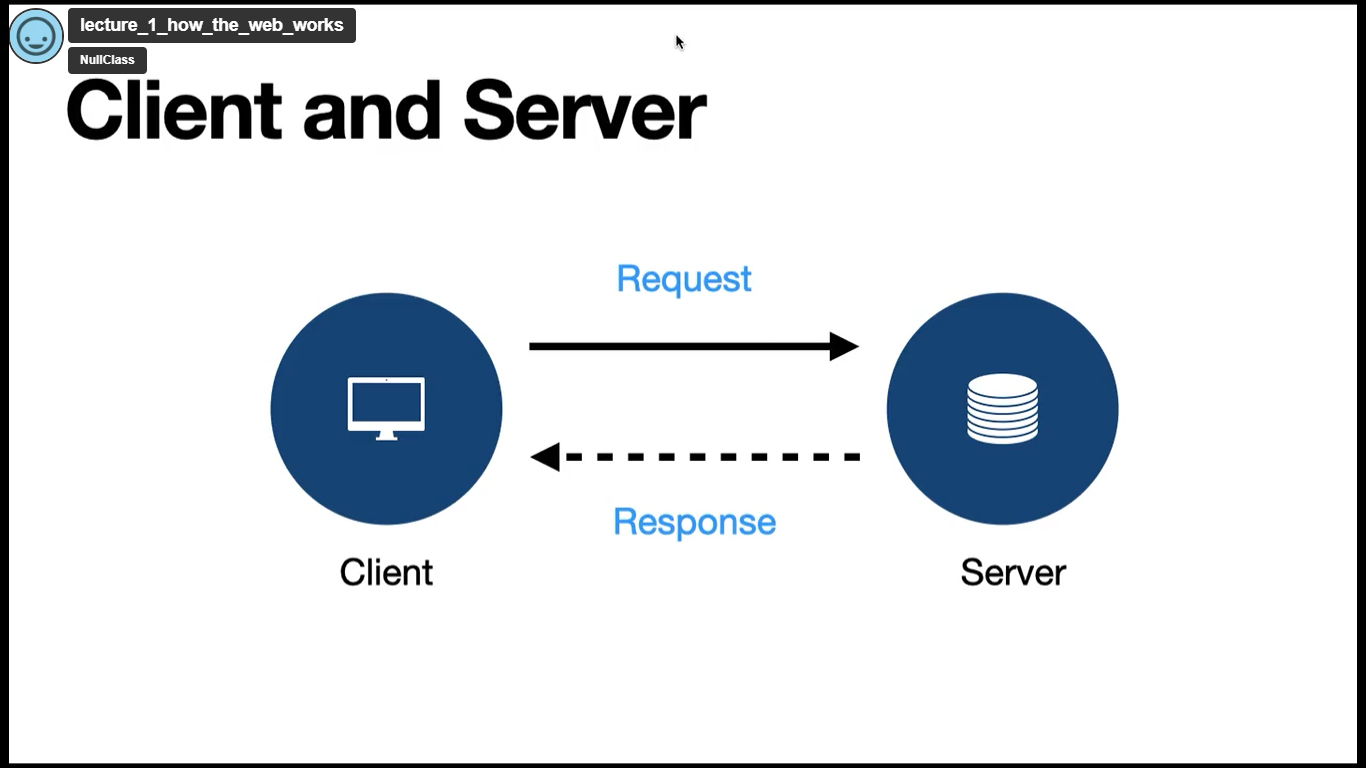
NodeJS is the popular and powerful JavaScript server platform, which is also called as JavaScript Runtime.

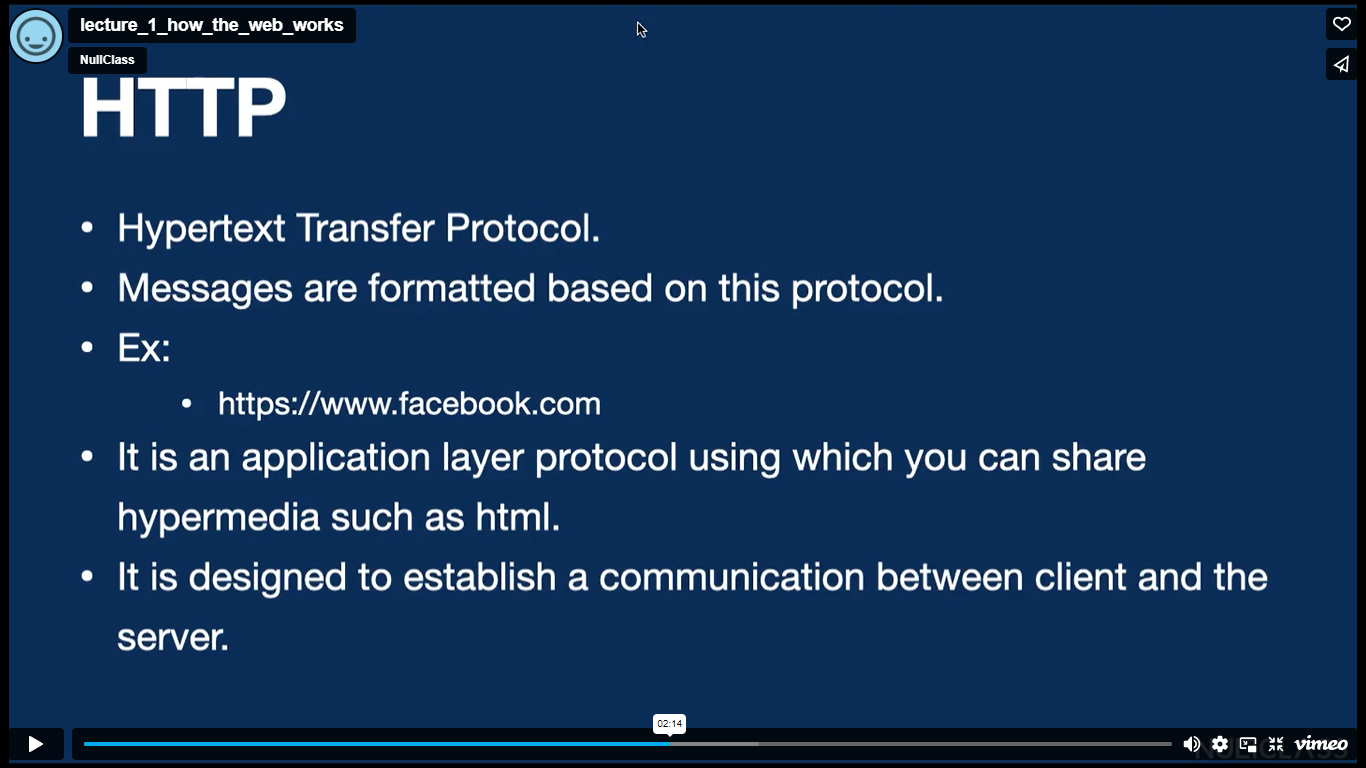
* HTML & CSS 🡪 for structuring and styling the website.
* JavaScript 🡪 makes the website interactive and user-friendly
* ReactJS 🡪 renders the JS and generates HTML at the client-side
* Redux / Redux Thunk 🡪 for statement ajament in ReactJS
* Redux manages the state of the project on the client-side
* Axios 🡪 handles request to the Back end or server-side
* NodeJS 🡪 initiates the server programs by writing scripts
* ExpressJS 🡪 powerful framework of NodeJS, used to get few functions
* Restful API 🡪 sends request and response in the form of JSON
* JSON Web Tokens 🡪 to complete the authentication
* MongooseJS 🡪 to access the database like MongoDB

**How the web works?**

Client sends a request to the server, which is known as HTTP request.

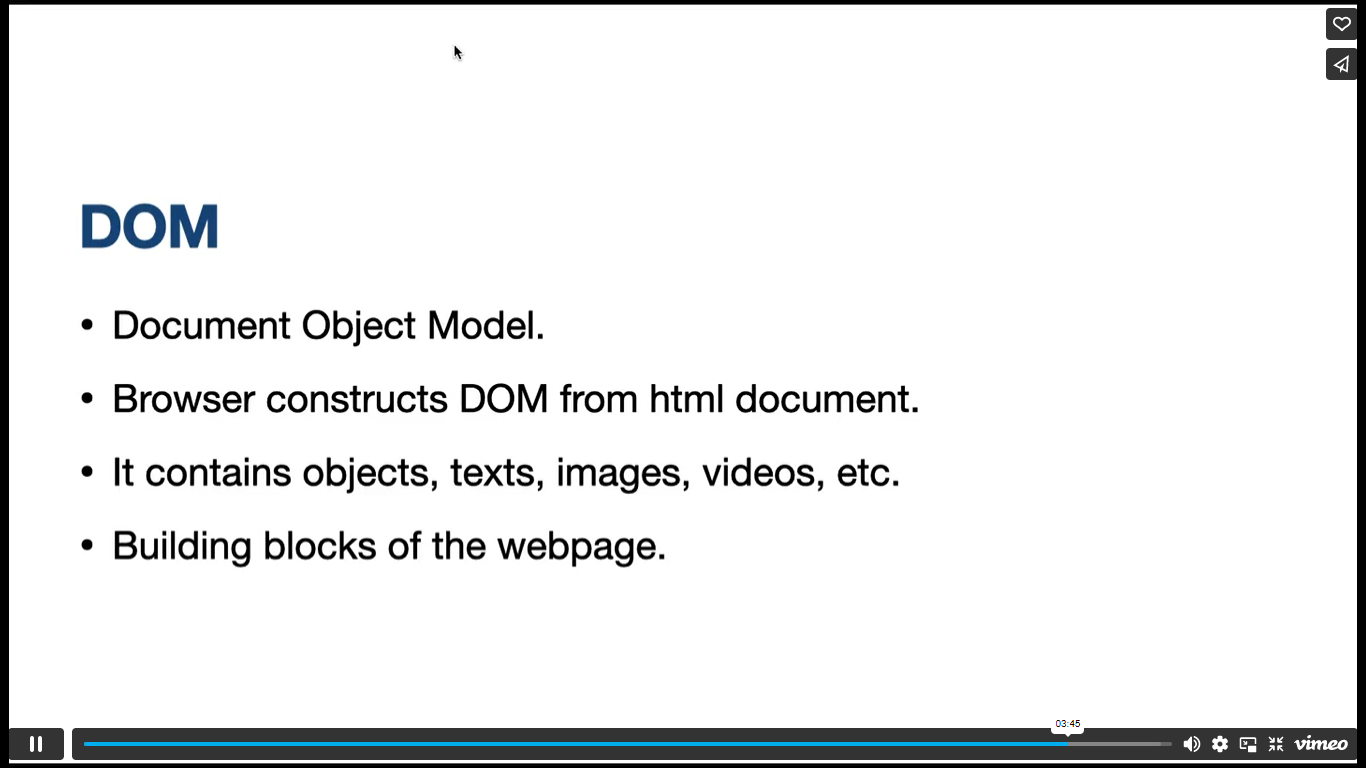
Server responds to it by sending a response, which is known as HTTP response.





https 🡪 http + encryption

The response from the server contains the html document of the web page, which matches the exact url entered by the client in the browser.



**HTML Basics:**

Hyper Text Markup Language

It is a markup language for creating web pages.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

</body>

</html>

Meta tags are used for SEO purpose and responsiveness of the web page.

Go to file, enable auto save to stop manually saving the html file.

Open a file with live server, which prevents us from refreshing the browser every time we make changes to the html file.

<pre></pre> tag preformats the code and can be used to display poems

Ex. <pre>Lorem ipsum dolor sit amet consecutur

Adipisicing elit. Aperiam obaecati earum<pre>

o/p: Lorem ipsum dolor sit amet consecutur

adipisicing elit. Aperiam obaecati earum

This pre tag displays the font as it is in the html file.

<!—code 🡪 this is the comment

<hr> tag is used to get lines.

Ex. <h1>Hello World</h1>

<hr>

<h2>Welcome to HTML tutorial</h2>

o/p: Hello World

Welcome to HTML tutorial

<a> tag can be used to add links.

Ex. <a href=”https:facebook.com” style=”background-color: aqua; color: green;”>Link to Facebook</a>

Refer to HTML Layout in w3 schools.

**CSS:**

Cascading Style Sheets

It describes how html elements are to be displayed on the screen.

Using CSS, we can control style of multiple web pages simultaneously.

3 ways of using CSS:

1. Inline CSS
2. Internal CSS
3. External CSS

Inline CSS overwrites the internal CSS.

 a{

                background-color: blue;

                color: aqua;

            }

.button{

                background-color: aqua;

                color:green;

            }

background-color and color are properties, blue and aqua are values.

:, ;, {} are syntax.

a tag is a selector. s//y, button is also a selector, but it is selected using a class, so dot is used before it.

/\*

Code

\*/

--- this is the way of commenting in CSS.

Margin property adds space outside the element.

Padding property adds space inside the element.

padding: 10px 20px 30px 40px;

--- here 10px is for top, 20px is for right, 30px is for bottom, 40px is for left.

padding: 20px 40px;

--- here 20px is for top and bottom, 40px is for left and right.

For li tag, we have display property.

Ex: li{

display: block;

}

--- this makes all the li content to appear line by line

li{

display: inline;

}

--- this makes all the li content to appear in the same line side by side

display: flex; --- also makes the content to be side by side, but it works only for limited html elements like div

flex-direction: column; --- makes the content to appear one below the other

justify-content: center; --- puts the content horizontally center

align-items: center; --- puts the content vertically center

justify-content: space-between; --- puts space between the elements

justify-content: space-evenly; --- puts even space around left and right of each element(including the edges)

justify-content: space-around; --- puts uneven space around left and right of the elements(including the edges)

min-height: 100vh;

--- vh here means view height

--- this makes the screen to be of size 100

section p{

    font-family: cursive;

}

In the above code, section is the selector, space is a combinator

The above code changes the content in p tag of section element has to become cursive.

section hr + p {

    font-family: 'Times New Roman', Times, serif;

}

In the above code, + is a combinator that means the adjacent sibling.

The above code changes the content in the p tag next to the hr tag.

section hr ~ p {

    font-family: 'Times New Roman', Times, serif;

}

Here ~ is a combinator that means all the siblings

The above code changes the content in all the p tags next to hr tag.

section > p {

    font-family: 'Times New Roman', Times, serif;

}

In the above code, > is a combinator that means all the particular children tags

The above code changes the content of all the p tags of section element

.button:hover{…}

.button:hover, button:active{…}

.flex-container .flex-elements:nth-child(1){…}

.flex-container .flex-elements:first-child{…}

.flex-container .flex-elements:last-child{…)

In the above code, hover, active, nth-child, first-child, last-child are pseudo classes.

Anything after (:) is a pseudo class.

--- Refer w3 schools or developer.mozilla.org for pseudo classes ---

Type div.box-container and press enter,

--- this in VS Code auto completes as

<div class="box-container"></div>

By pressing **alt + shift + down arrow** by placing the cursor after </div> tag will paste the

<div class=”box-container”></div> code in the next line.

A position property in CSS allows you to navigate a particular container throughout the web page.

 position: static;

this makes no difference to the element containing it. The element will be displayed as it is.

position: relative;

top: 10%;

left: 30%;

this means 10% space from relatively top and 30% space from relatively left.

If position: relative; is not there then top and left will not work.

 position: absolute;

this makes the element above other elements in the web page.

 position: absolute;

 z-index: -2;

this makes the element to be below the second element, since the z-index of second element is assumed to be 0

.box-container:nth-child(2){

    background-color: blue;

    position: fixed;

}

This is sllr to absolute, by making the element above other elements. Also, though you scroll the page up or down the element will remain stable in its original position.

 position: sticky;

top: 30%;

left: 70%;

this means that when the element reaches 30% from the top while scrolling the page up or down, the element remains fixed there itself, it will not move anymore.

transform: translate(50%,50%);

in the above code first 50% is in X-axis(left side) and the second 50% is in Y-axis(top).

transform: translate(100px, 500px) rotate(40deg) skew(20deg, 10deg);

the above code rotates our element in 40 degrees and skews our element 20 degrees in X-axis and 10 degrees in Y-axis.

scale(n) value for transform property makes the size of the element n times bigger than the original.

To change the color of a link slowly when hovered, we can use transition property.

@keyframes identifier{

from{

}

to{

}

}

--- here identifier is the animation-name, from and to are the positions.

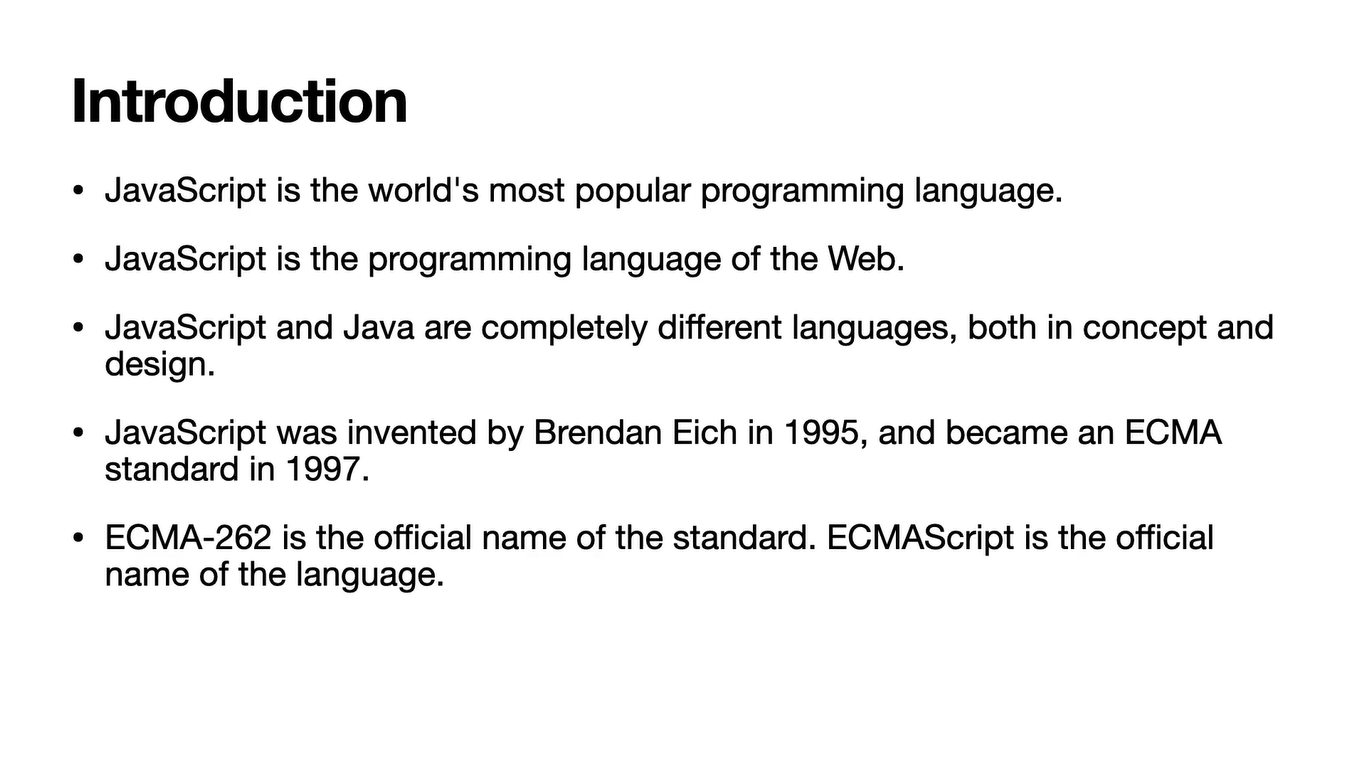
animation-delay: 5s;

--- our animation starts our 5 seconds.

animation-timing-function: ease-in-out;

--- different speeds for our animation in different sides.

**JavaScript:**



JavaScript is a scripting language, that is a programming language designed for system to automate the execution of tasks.

Latest version of JavaScript is es6.

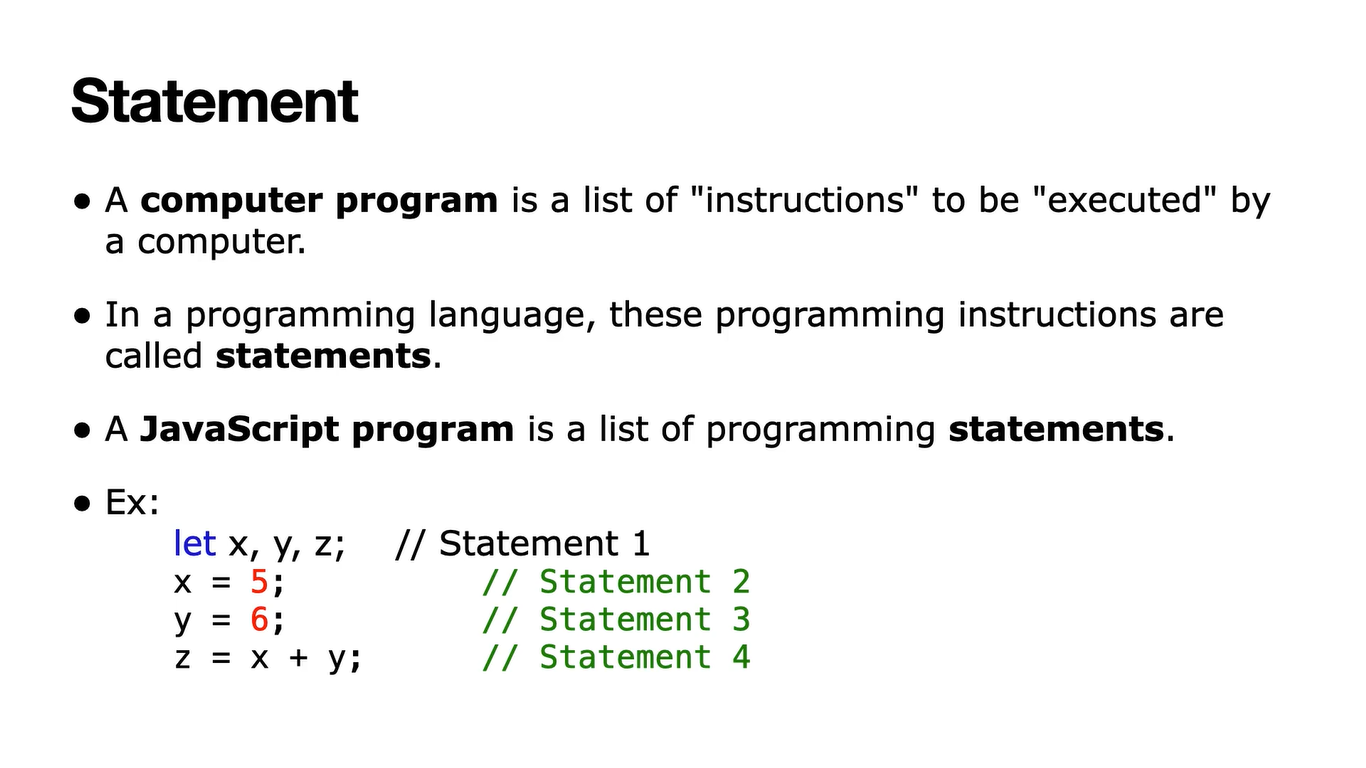
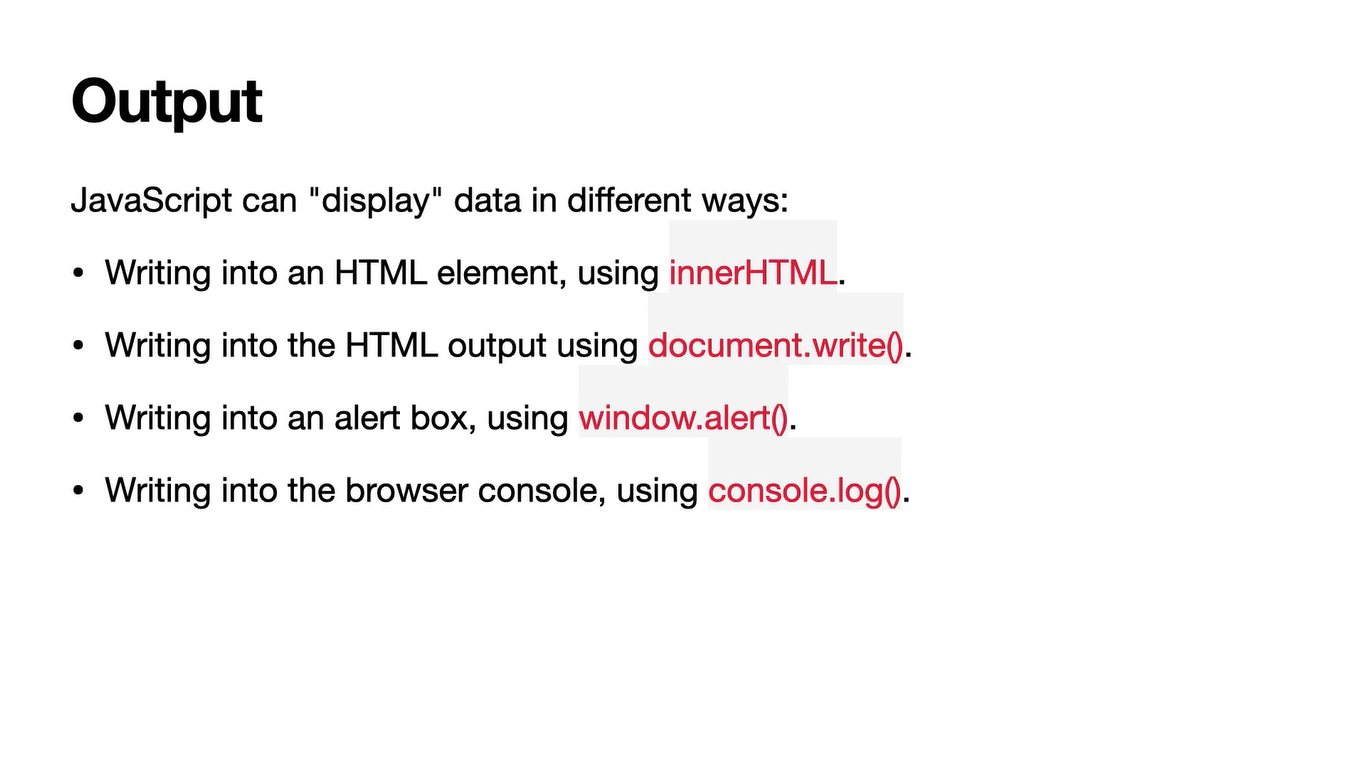
There are 3 ways to declare a JavaScript variable:

1. Using var
2. Using let
3. Using const

All JS variables must be identified with unique names which are called identifiers.

If you declare a variable with const, you must assign a value to it in the beginning and it cannot be changed anytime in future.

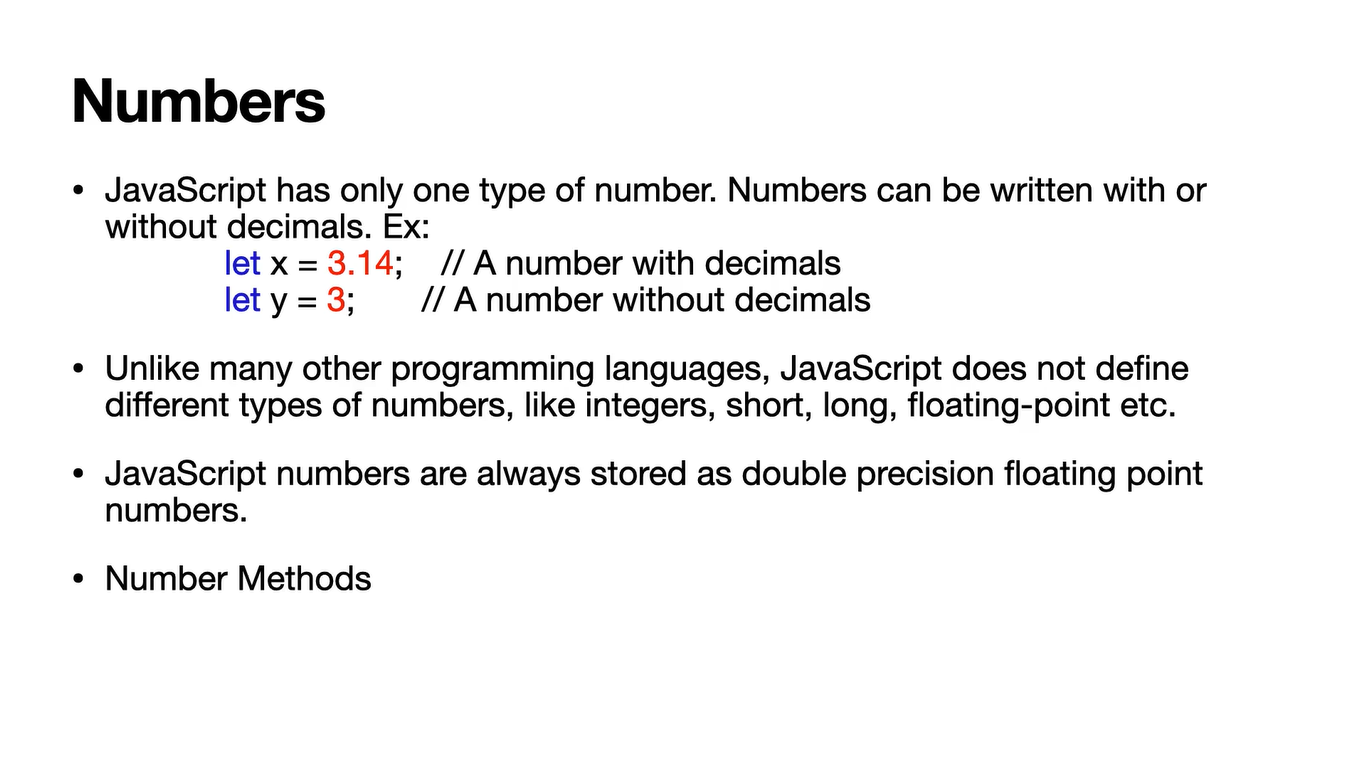
Ex. const name = “Pranavi”;

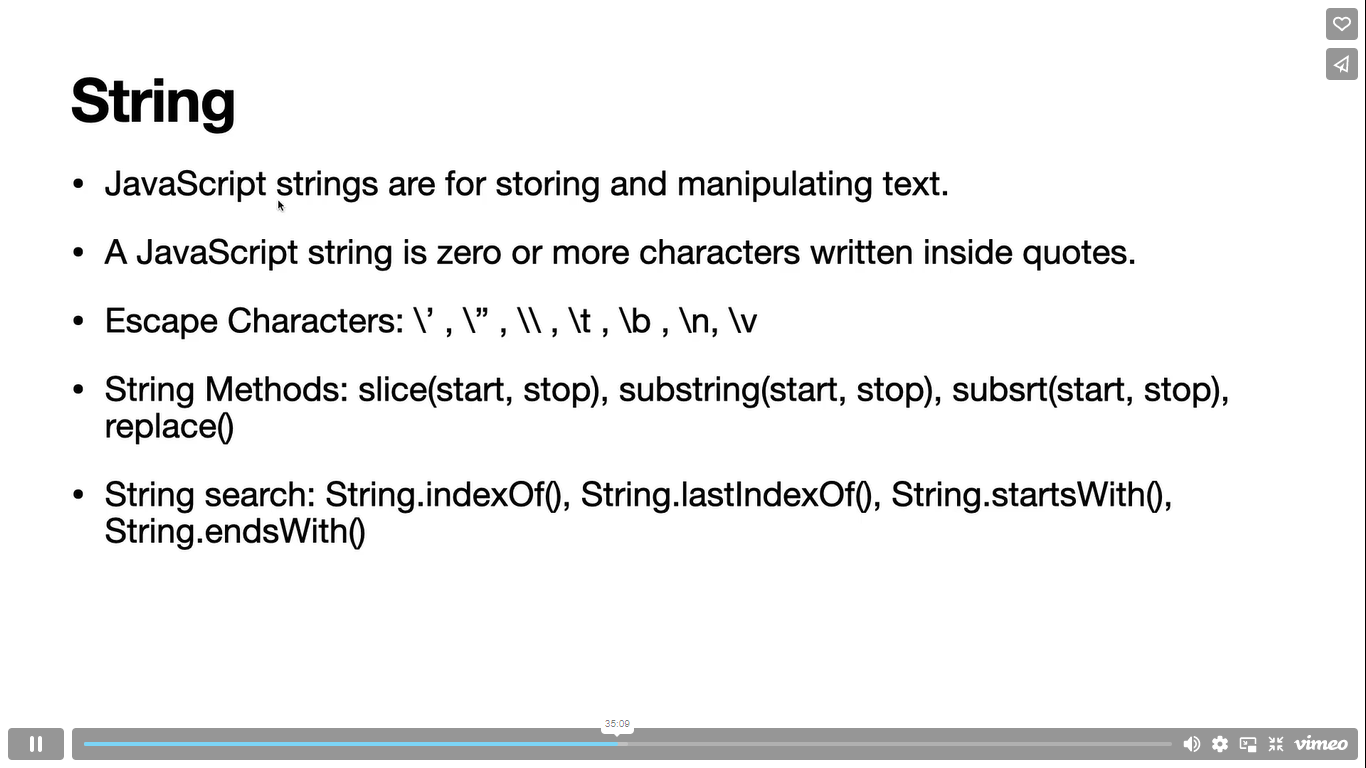
A data type is a classification of data which tells the compiler or interpreter how the programmer intends to use the data.

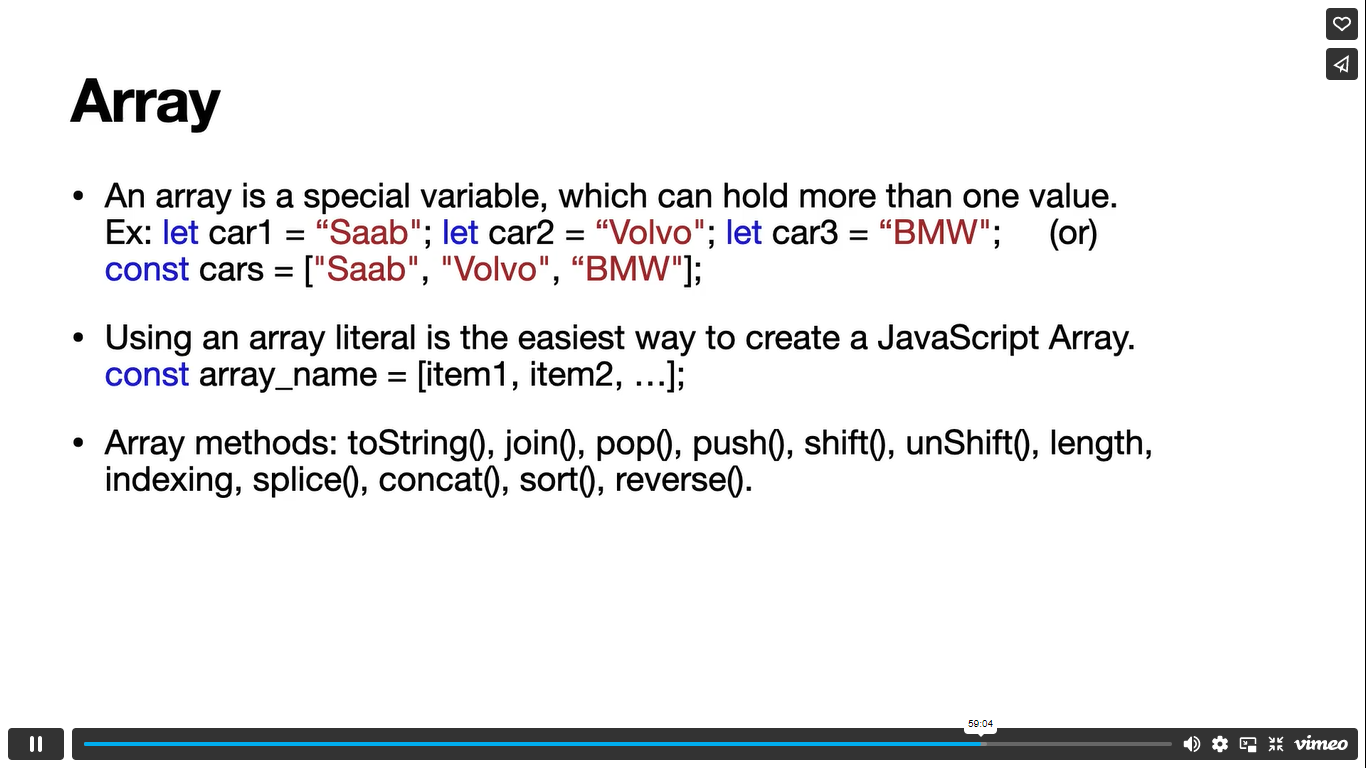
The data types in JS are:

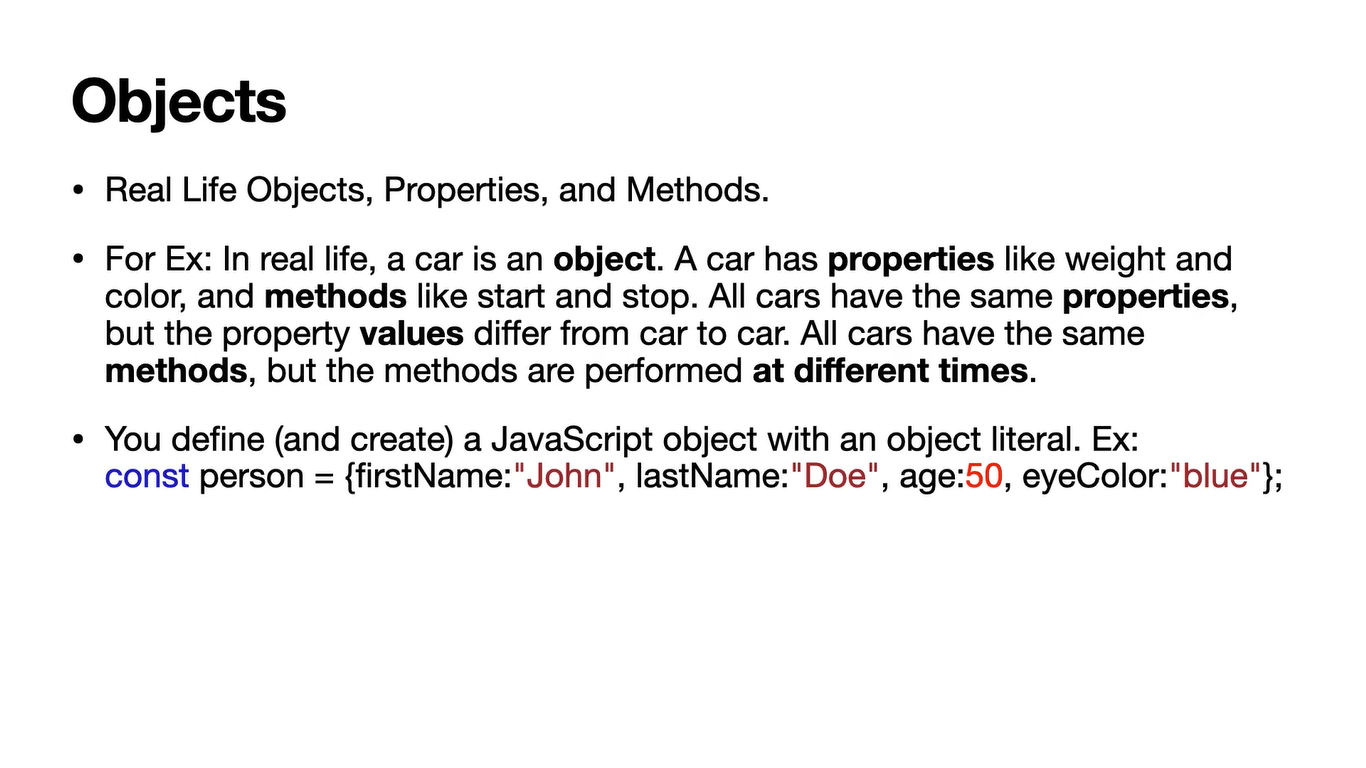
1. Number
2. String
3. Arrays
4. Objects

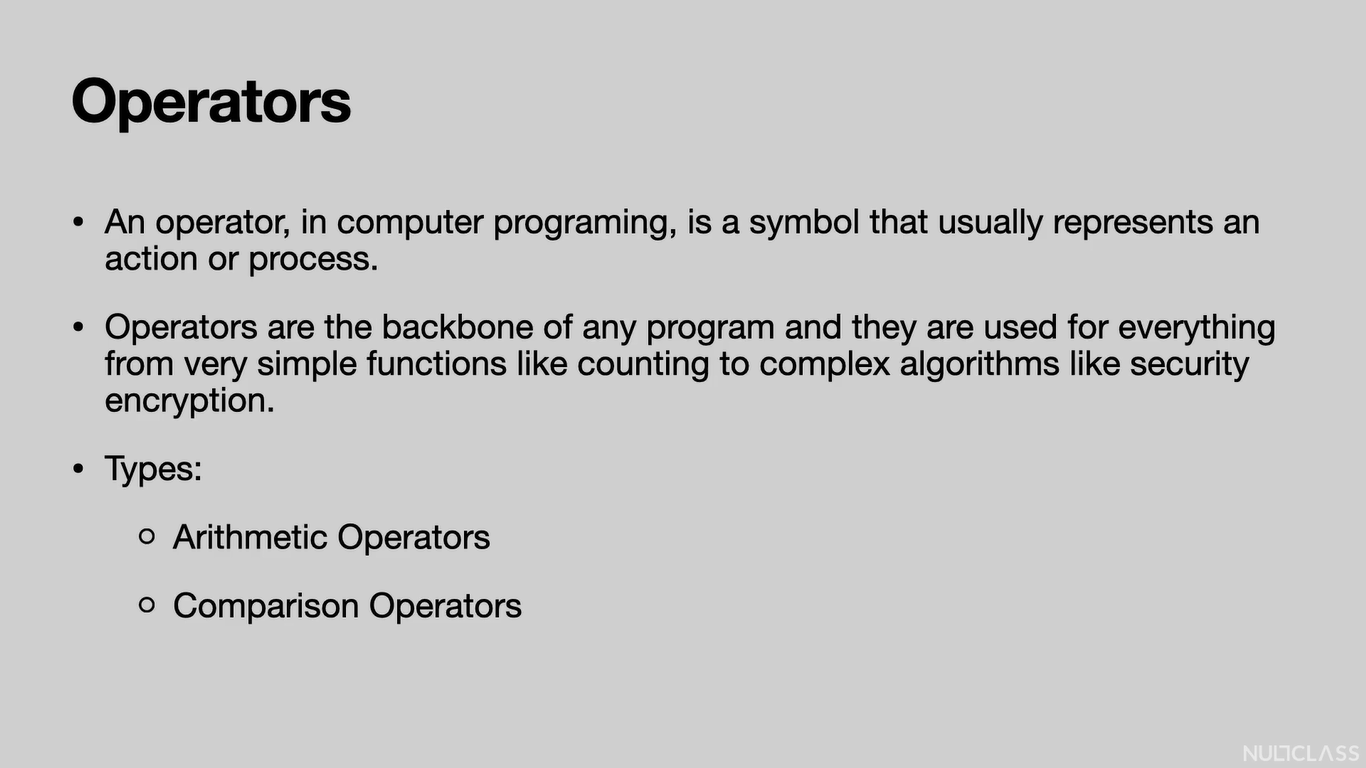


Double precision floating point numbers have the precision of 8.

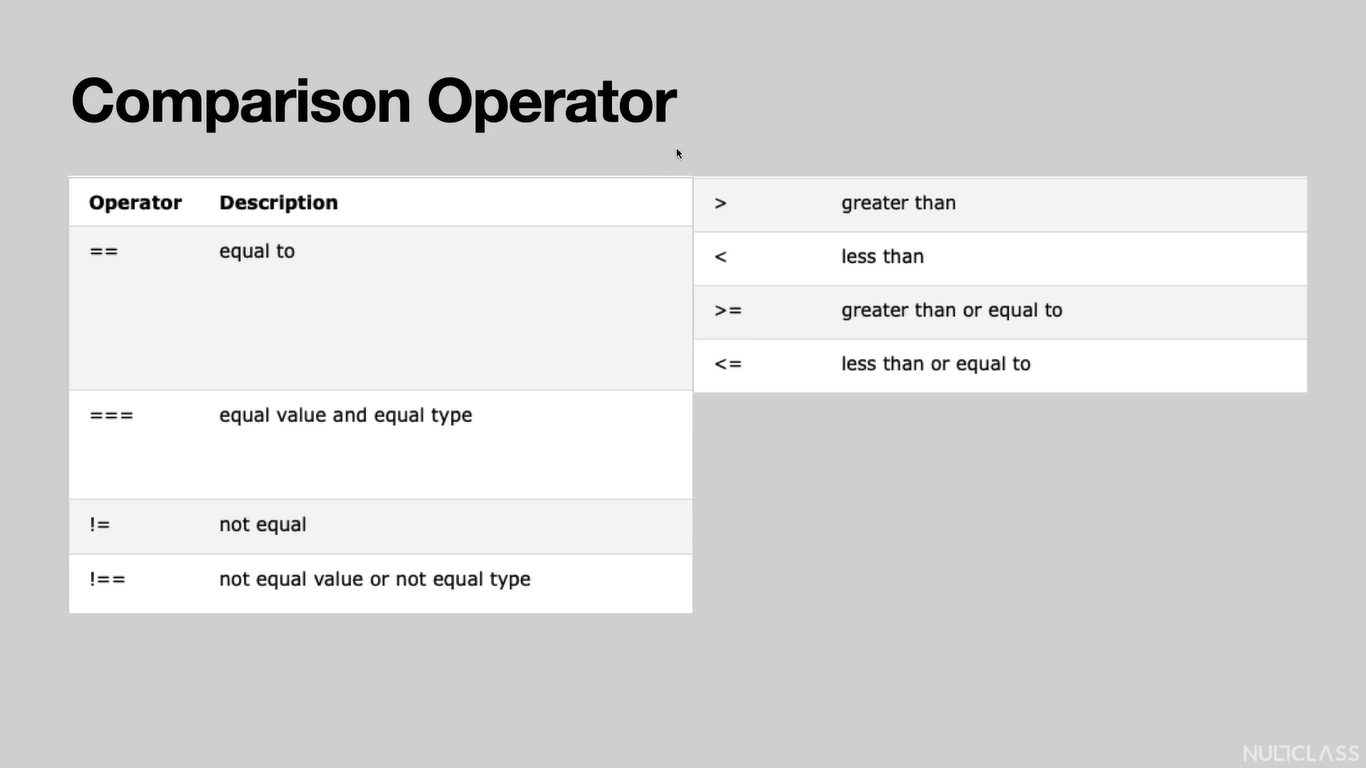












Logical Operators: &&(AND), ||(OR), !(NOT)

Conditional Statements:

if – else if – else:

if(condition1) {

//statements

}

else if(condition2) {

//statements

}

else {

//statements

}

Switch:

switch(expression) {

case x:

//code

break;

case y:

//code

break;

default:

//code

}

Loops:

For

for (statement 1(variable): statement 2(condition): statement 3(increment/decrement)) {

//code

}

for (key in object) {

//code

}

for (variable of iterable) {

//code

}

While:

while (condition) {

//code

Increment/decrement

}

Ex:

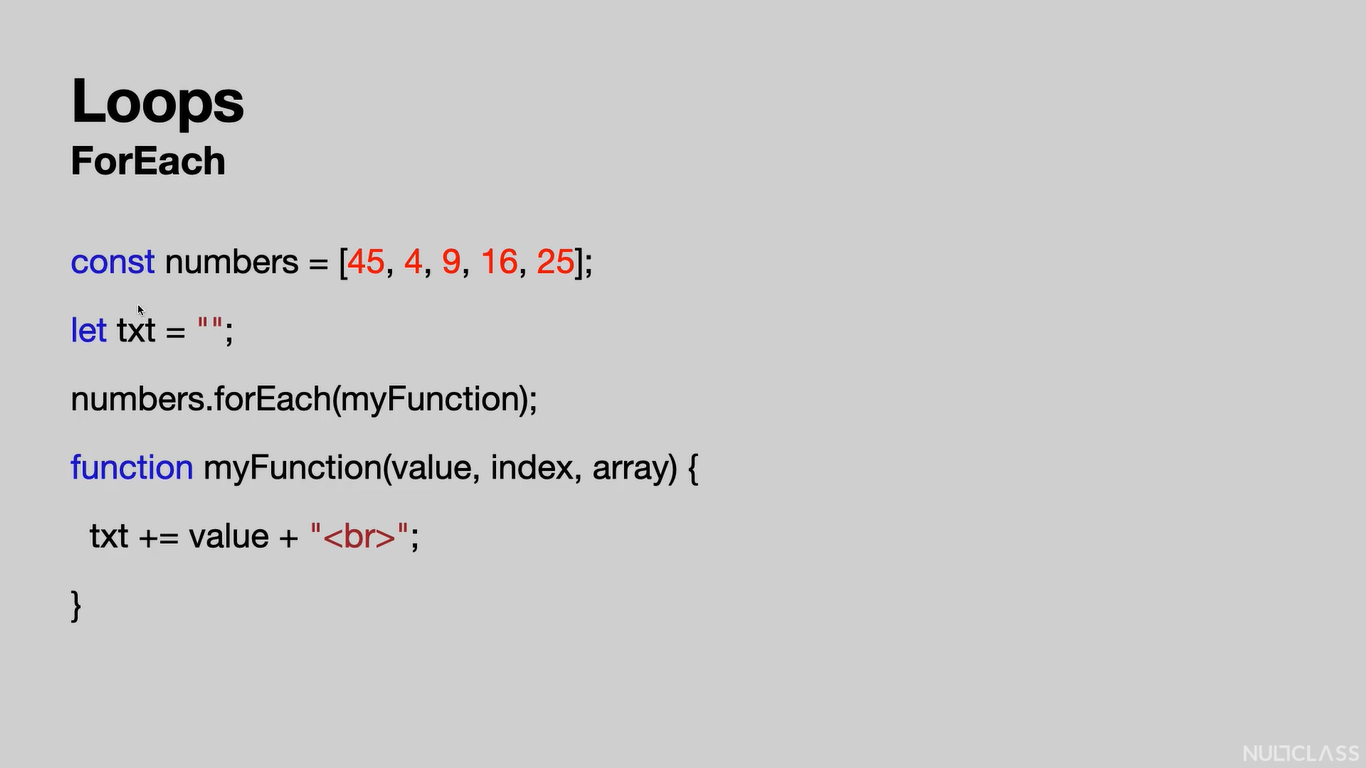
i = 0;

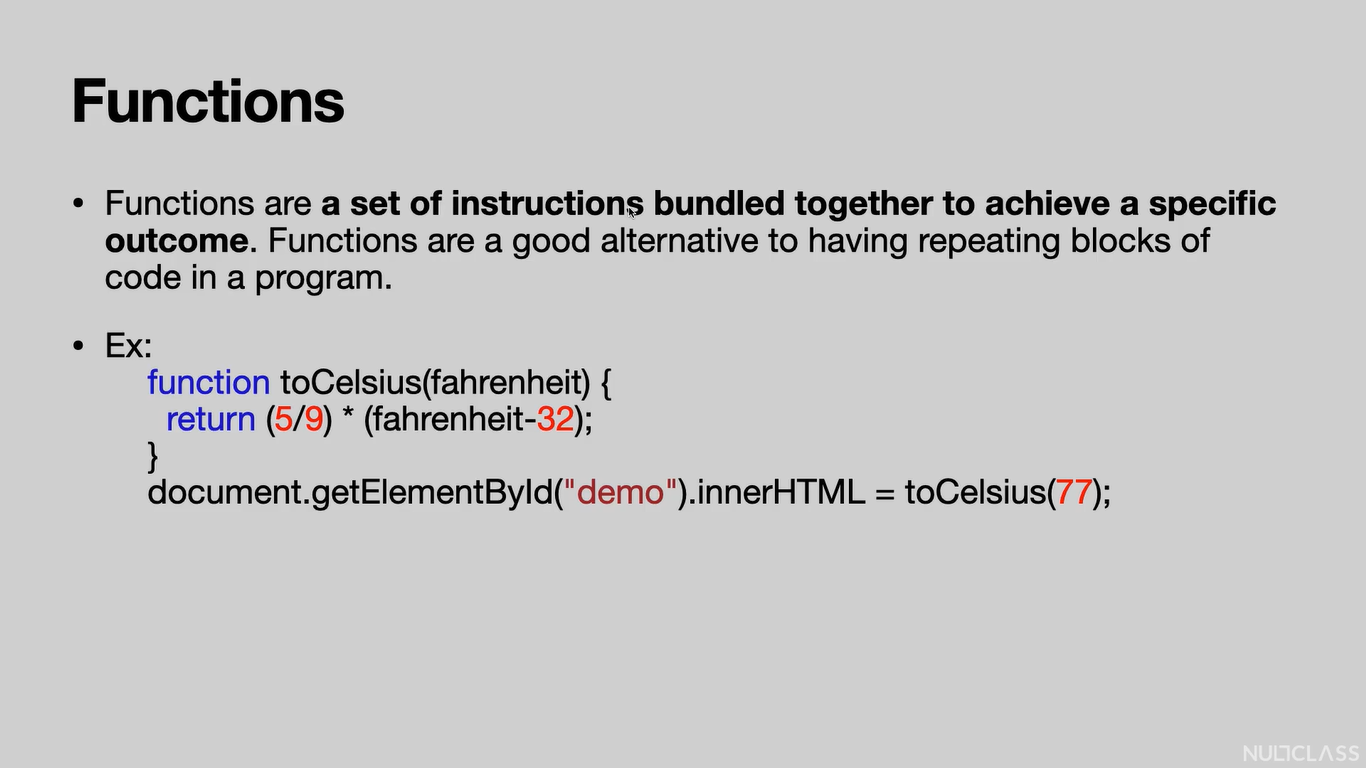
while (i < 10) {

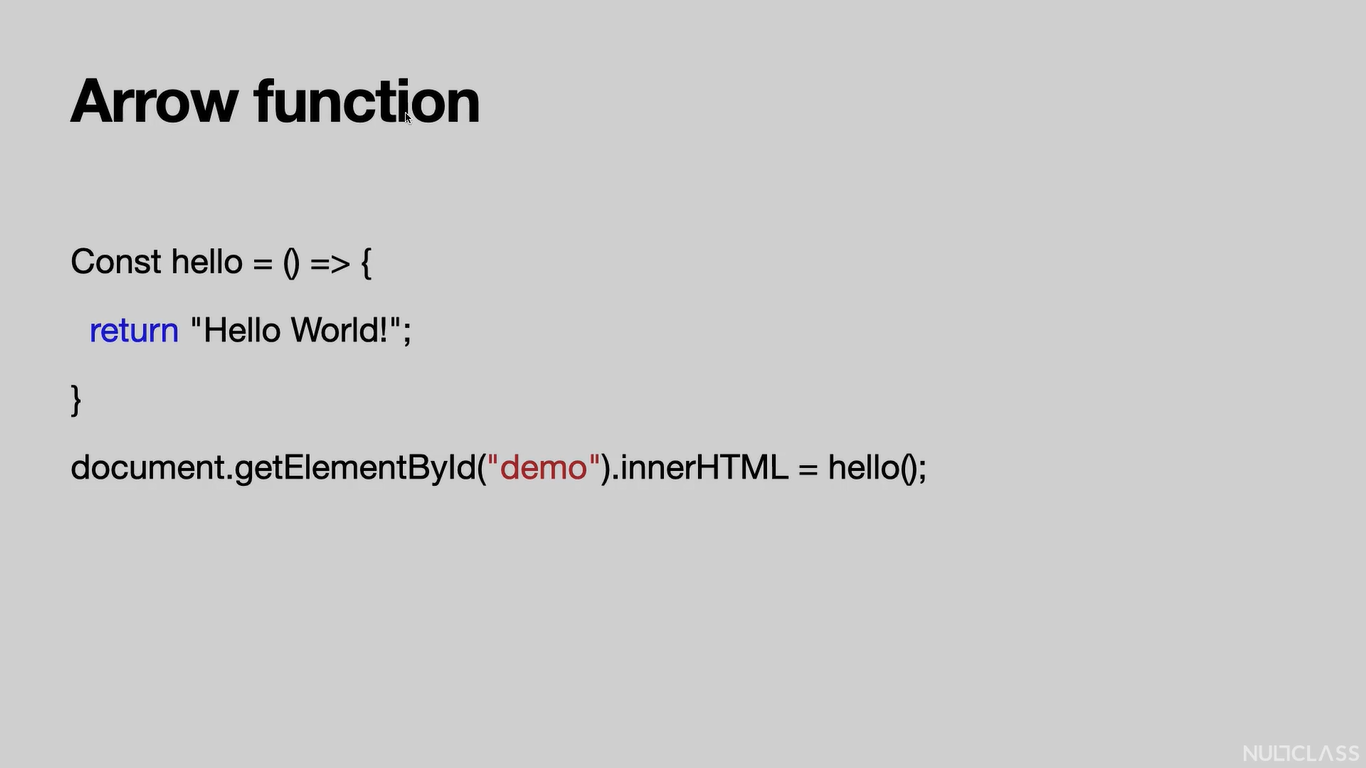
text + = “The number is “+ i:

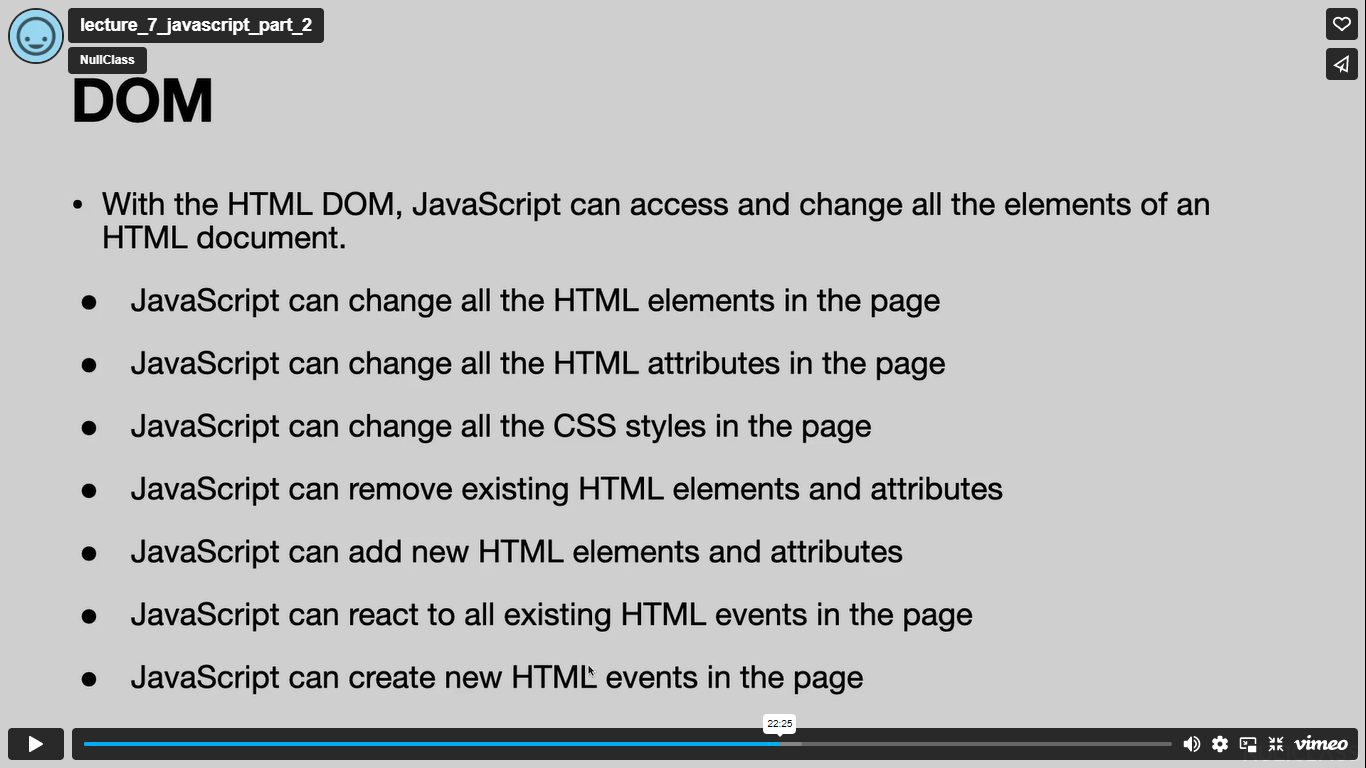
i++;

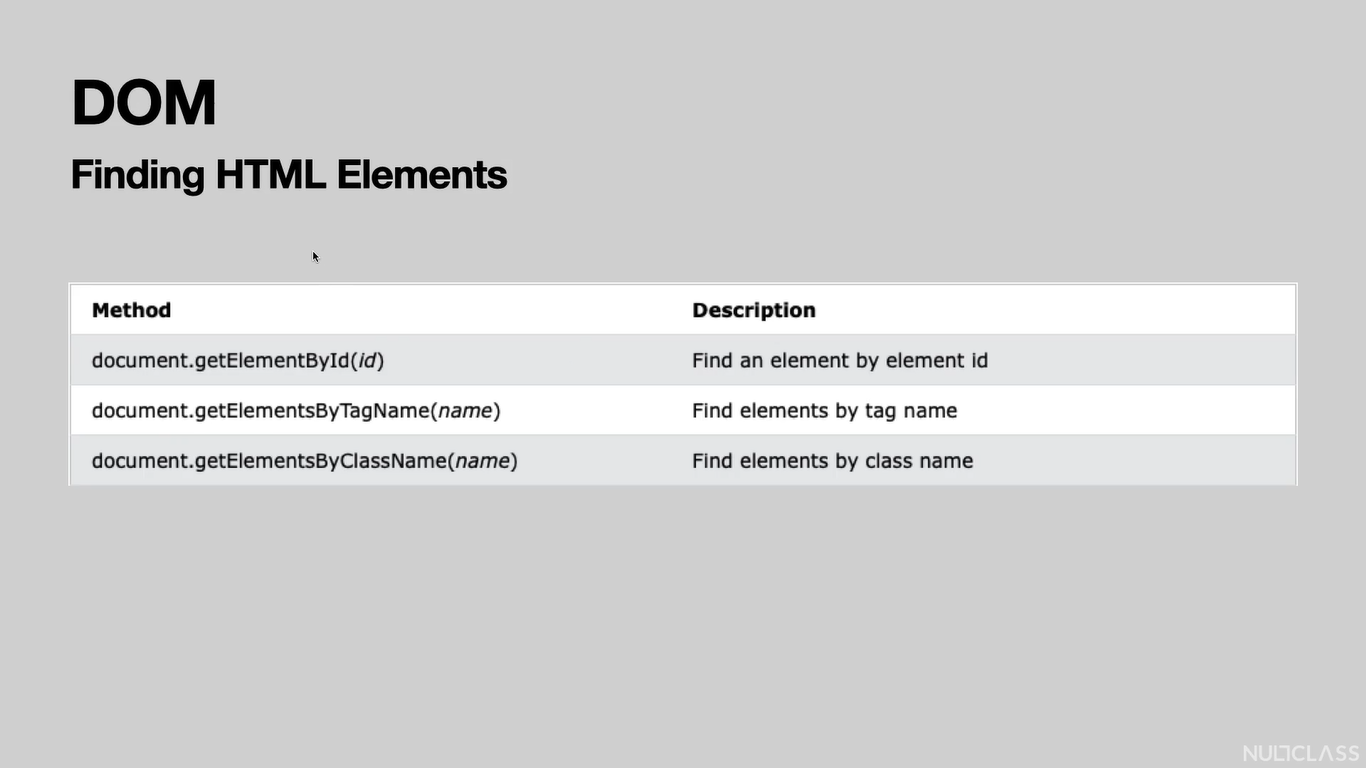
}

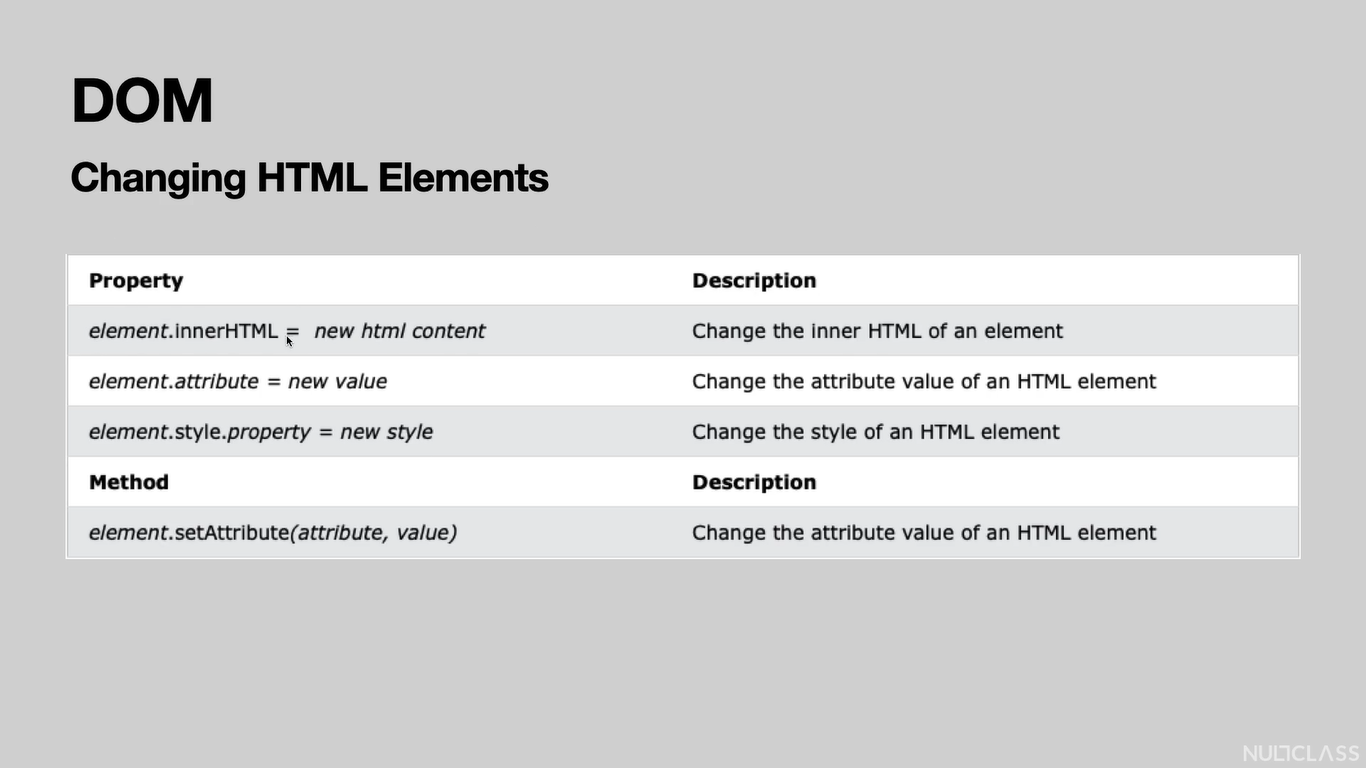


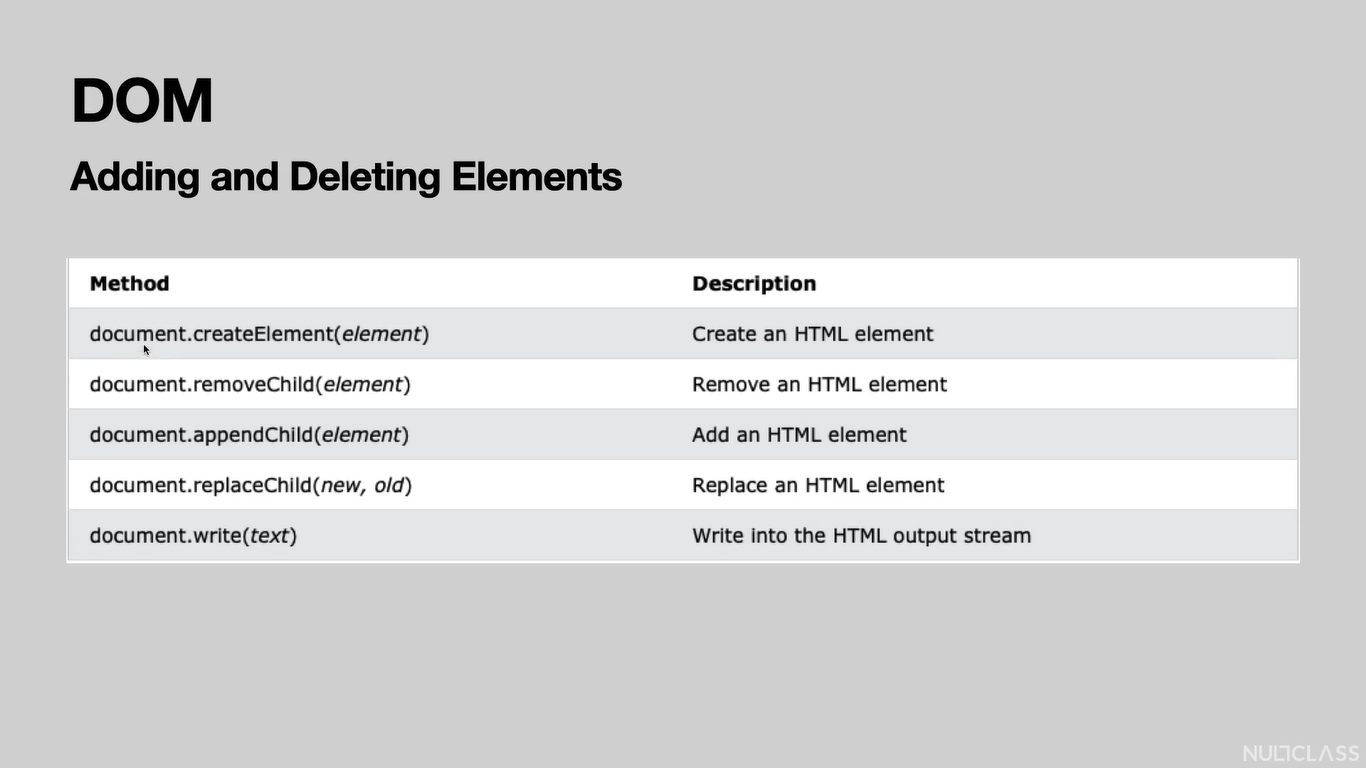


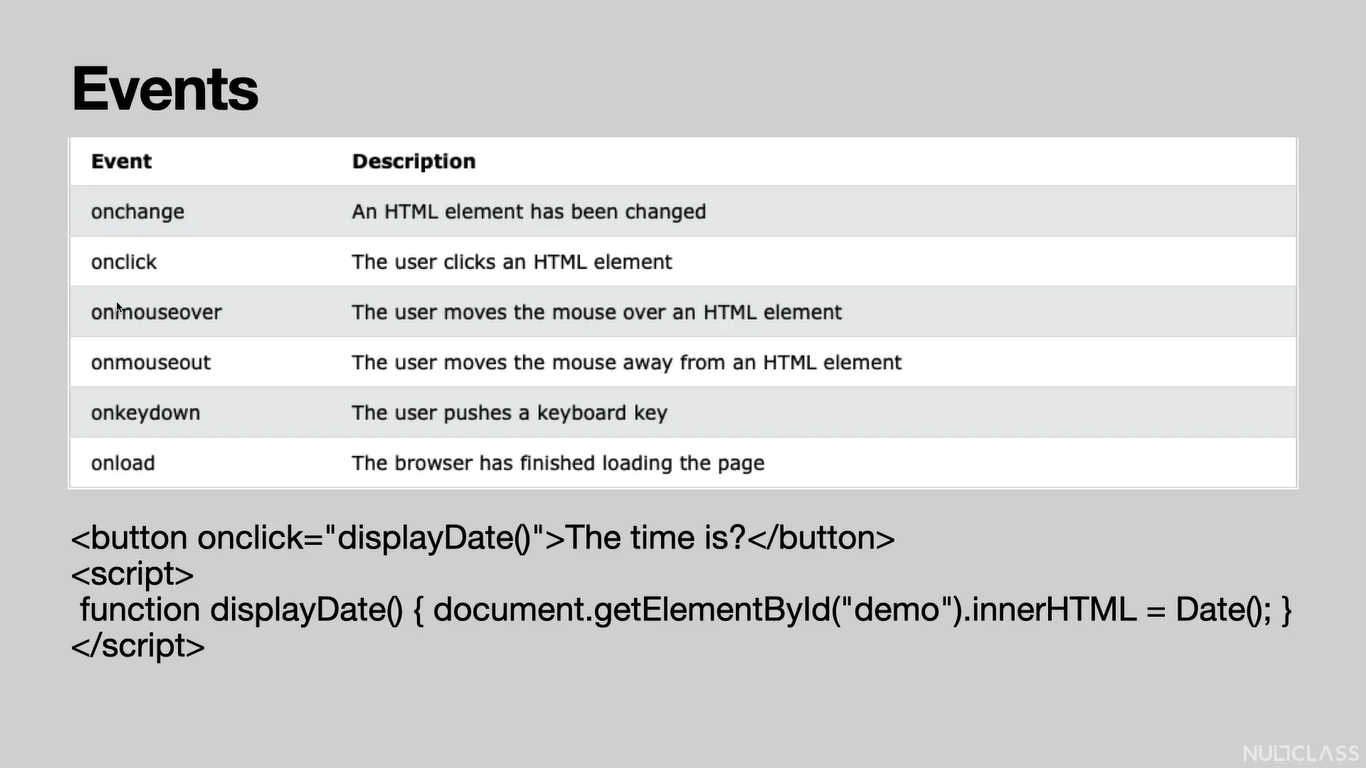












**ReactJS:**

In cmd,

C:\Users\saimanaswi1999\OneDrive\Desktop\NullClass Internship>npx create-react-app my-app

After my-app project is created in the folder, check the contents in the directory using dir command

Command:

C:\Users\saimanaswi1999\OneDrive\Desktop\NullClass Internship>dir

o/p:

Volume in drive C is OS

Volume Serial Number is 92C6-77A1

Directory of C:\Users\saimanaswi1999\OneDrive\Desktop\NullClass Internship

12/03/2022 11:54 AM <DIR> .

12/03/2022 11:54 AM <DIR> ..

12/02/2022 12:22 PM <DIR> HTML - CSS Tutorial

12/02/2022 09:47 PM 4,380 index.html

12/02/2022 09:48 PM 2,117,690 Learn To Build a Real Time Website Like StackOverflow.docx

12/03/2022 11:56 AM <DIR> my-app

2 File(s) 2,122,070 bytes

4 Dir(s) 26,356,084,736 bytes free

Open the project in VS Code, run the app in terminal

>npm start

In package.json, we have scripts tag, in that we have “start” property. When npm start command is run, it comes to this “start” and executes its value i.e., starts the react scripts.

The dependencies mentioned in package,json file will be installed inside node\_modules folder.

JSX: JavaScript Extension.

Props are the parameters of the components.

If you need to use javascript inside JSX, you need to provide it inside curly braces {}.

In App.js, if we click h1 / p and press enter in div tag as in HTML file, it will not automatically generate the h1 tag or p tag. To enable this facility, click on JavaScript which is available at the bottom, type react in the search box, click on JavaScript React.

Remove the unnecessary extensions and comments (from App.js, index.js, index.html) and files (from src and public folders).

In src folder, create a components folder and create HelloWorld.jsx file in it.

The .jsx extension will automatically enable JavaScript React in the file, instead of manually doing it like for the App.js folder.

From extension, install the ES7 React/Redux/GraphQL/React-Native Snippets.

In ReactJS, we don’t need to run the file with live server, it is inbuilt.

Write a function with name HelloWorld in HelloWorld.jsx file and export it.

function HelloWorld(){

}

export default HelloWorld

As we are exporting it by default, the file name and the function name have to be same.

Type the below code in the function,

return(

        <h3>Bye World</h3>

    )

Now import HelloWorld in App.js.

Add the following JSX tag in div tag of App.js,

<HelloWorld/>

We can have the above JSX tag multiple times for code reusability.

If it is not a default export, then

export function HelloWorld(){

. . .

}

We can write export beside the function keyword and,

import { HelloWorld, ByeWorld } from "./components/HelloWorld";

we should have curly braces like in the above code.

Also, the above code says that we can import multiple functions at the same time.

HelloWorld and ByeWorld are termed as components which are inturn functions.

We can have the parameters for a function using props,

export function Concern(props){

    return(

        <h3>{props.text}</h3>

        )

}

text comes from the component in App.js,

<Concern text="How are you?" />

Another way to give parameters without props keyword is,

export function Concern({text}){

    return(

        <h3>{text}</h3>

        )

}

React Hooks:

Hooks make React apps simpler.

In pre-hooks era, components were written in classes, instead of functions.

Hook is a special function that lets you “hook into” React features.

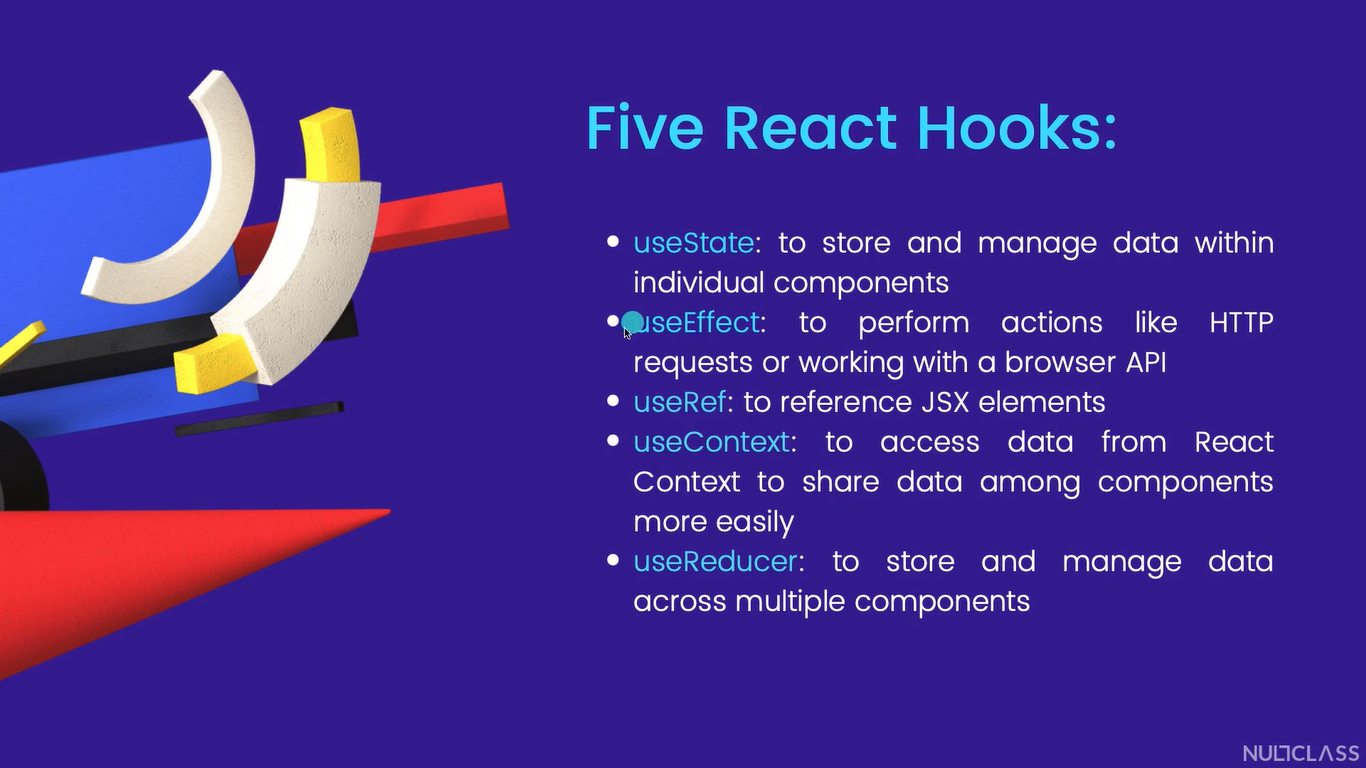
Ex: useState is a Hook that lets you add React state to function components.

In class component, we had state variable which used to contain the variable declared by us.

We can store any data type like number, array, etc in useState.

Hooks are essential because with hooks,

1. You don’t have to refactor a functional component into a class component(it is easier to write a functional component than a class component).
2. You don’t have to worry about “this” anymore.
3. No more method bindings.
4. Easier to decouple logic from UI, making both more reusable.



useContext is used to share data from parent component to its child components. It is not used more in industry. Instead of it, redux is used.

If a variable is declared normally(without useState hook) using var and it was incremented using a function and printed both in console and on screen, it will show the incremented value only in console, but not on the screen.

const [ counter, setCounter ] = useState(0);

the above code is a way to declare a useState variable with the help of array destructuring.

Here useState variable is an array whose first value is a variable and the second value is a function.

Here useState variable is of number data type.

We can have it of array data type or object data type also.

useEffect creates a side effect.

useEffect is a function which allows you to pass a function inside it.

document.title = `clicked ${counter} times`;

this is a way of string formatting and to have any value like counter inside the string.

useEffect(() => {

    document.title = `clicked ${counter} times`;

  });

With this, useEffect() does the side effect of changing the title of the document.

document.title = `clicked ${counter} times`;

--- this is the request

The main use of useEffect() is to fetch data through API from the backend.

useEffect(() => {

document.title = `clicked ${counter} times`;

  },[]);

The [] is used to mention dependency list.

If [] is not mentioned, then the request will be sent to the server a million times.

In the above example, we can provide counter value in the dependency list, because the title of the document will change only if the counter value changes.

useEffect(() => {

    document.title = `clicked ${counter} times`;

  },[counter]);

React Router can be used to route to different components staying in the home page itself.

To use react router, install the react-router-dom in the terminal using the following command

>npm install react-router-dom

Now we can see react-router-dom in dependencies in package.json.

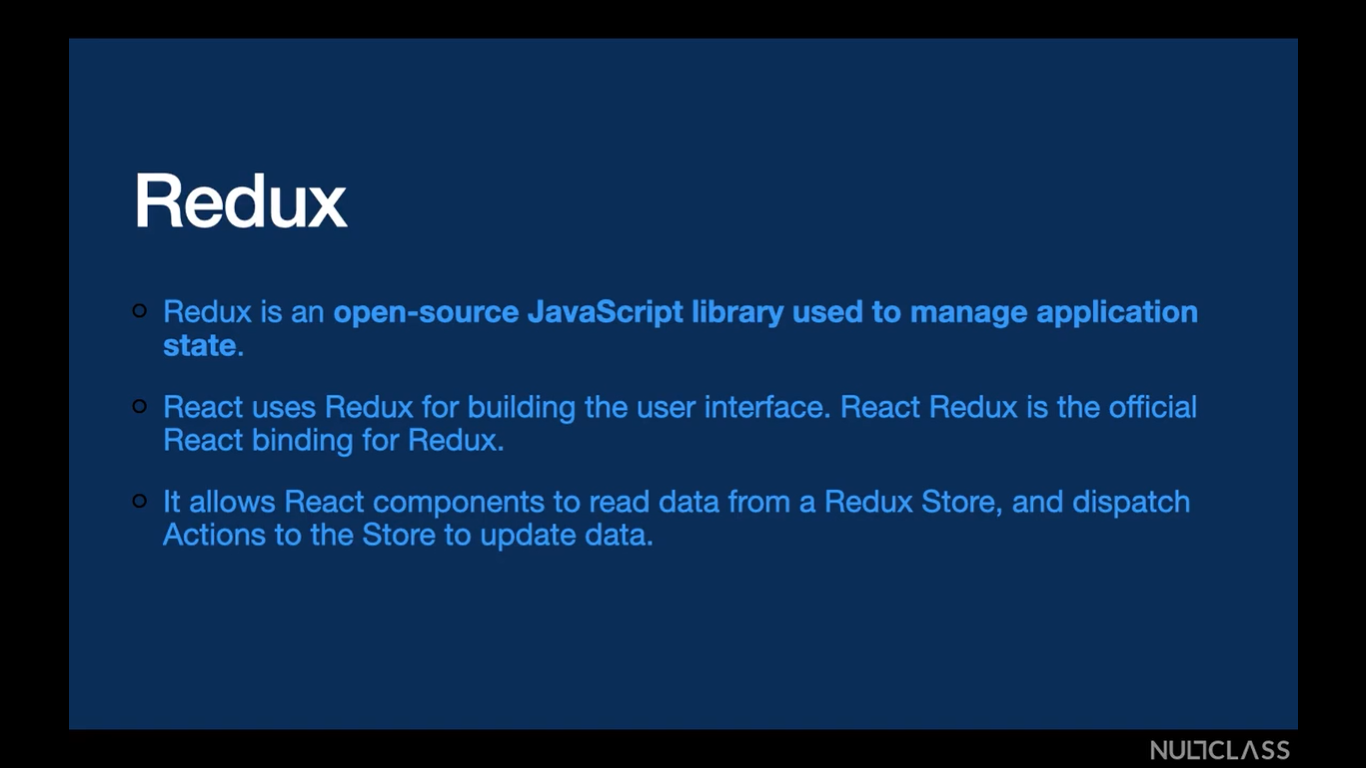
Import react-router-dom in App.js, add nav tag in div.

Make the necessary changes in App.css.

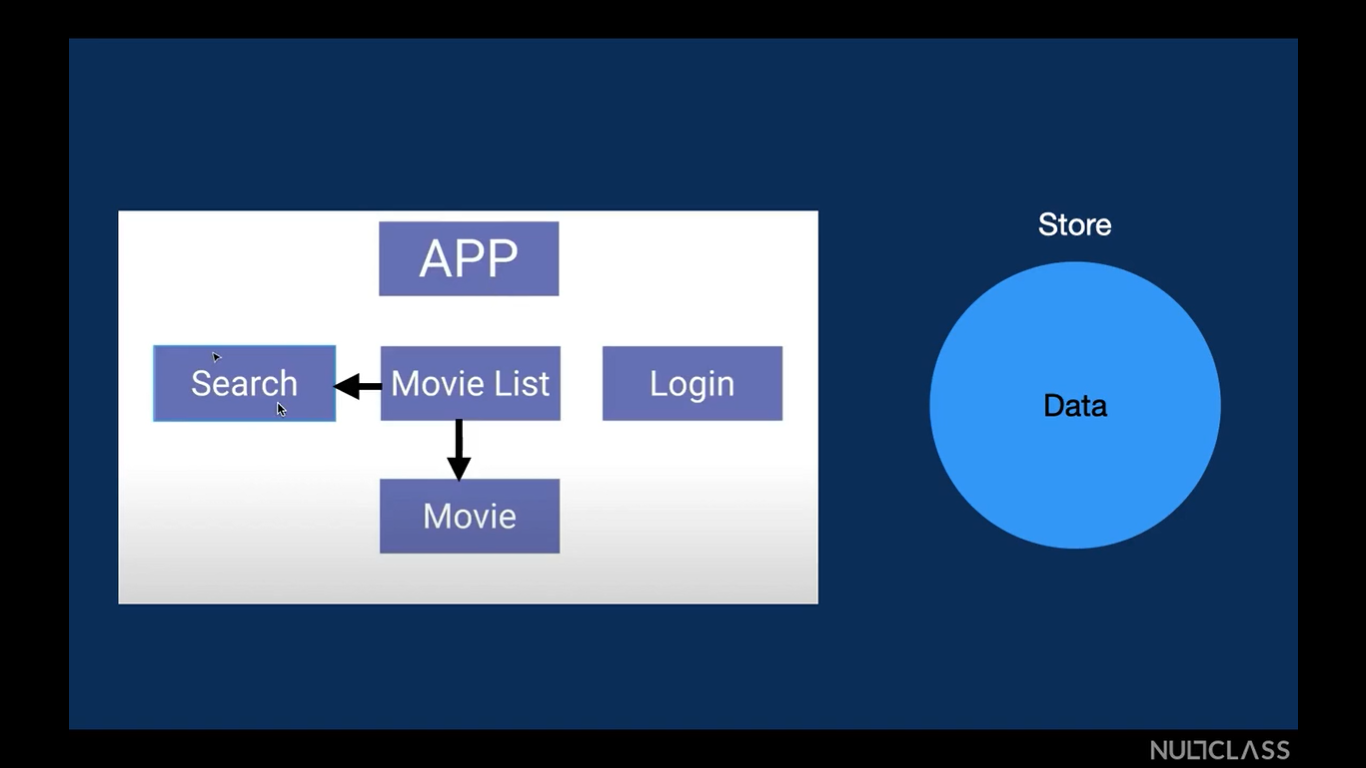
Now create Contact.jsx file and type rafce and press enter, this will automatically generate required react code.

**Redux:**

Redux can be used with any JavaScript framework.



The data in the Redux Store will be outside our react app but is accessible by the react app.



From the above, Search, Movie List, Login are three components and Movie is the child component of Movie List.

In React, the data in Movie List can only pass downwards, ie., to Movie, but not to Search.

With useContext API in React, we will move the data from Movie List to APP, from APP we send the data to Search.

In Redux, we just store the data in Movie List in another file called store. Now this data can be used anywhere and in any component like Search/Login/Movie in our APP.

**Creating real project “Stack Overflow”:**

Create a new folder named stack-overflow-clone, open it in vscode.

Create two files named client and server in it.

Open the terminal,

>cd client

>npx create-react-app ./ -🡪 ./ implies that no new folder will be created and the react project will be created in the client folder.

Split the terminal and in other terminal,

>cd server

>npm init

In the client side terminal,

>npm i axios jwt-decode moment react-copy-to-clipboard react-router-dom redux react-redux redux-thunk

In the server side terminal,

>npm i bcryptjs jsonwebtoken cors dotenv express mongoose nodemon

After installing all the necessary packages, delete the server side terminal and have only client side one.

In the client side terminal,

>npm start

Make necessary changes in App.js, index.js and index.html files.

Create a folder named assets in src folder.

Clear all the content in App.css.

fontawesome.com ---- the website to search for logos.

<label htmlFor="email">

                    <h4>Email</h4>

                    <input type="email" name='email' id='email'/>

htmlFor and id should have the same value, because if id is not there or have any other value then in the browser when we select on Email then the input will not be selected. Only if the id has the value of email, the input will be selected.

Link and NavLink are same, the Link used in Navbar is called NavLink.

In this way, create necessary files and make changes for front-end in client folder.

In server folder,

In package.json,

In “scripts”, change to “start”: “nodemon index.js”

After “main”: “index.js”,

Add “type”: “module”

Then in index.js file, we can import using the command

import express from ‘express’

If “type”: “commonjs” is present(default one),

Then in index.js we have to import using the command,

const express = require(‘express)

const app = express();

with this, our express server will be created of name app.

In MongoDB Atlas, create a new project named stack-overflow-clone

Then Build a Database which is Shared

Give the cluster name as stack-overflow-clone, click Create.

Check for its status in Database.

In Database Access, Add New Database user, give username and password as admin each, then click on Add User.

In Network Access, Add IP Address, click on Allow Access from anywhere and Confirm.

After the database-cluster is created, click on Connect, Connect your application.

Driver be Node.js and Version is 4.0 or later.

Copy the url code.

app.use(express.json({ limit: "30mb", extended: true }));

app.use(express.urlencoded({ limit: "30mb", extended: true }));

app.use(cors());

whatever written in use() is called a middleware.

To run the code in terminal,

>cd server

>npm start

When a sever is sending request to another server, then another server will throw an error, to eliminate that we use cors.

Create necessary folders, files and make necessary changes in those files.

Now in the client folder,

In src, create reducers folder to store the data and actions folder to dispatch any update or modifying data which is stored by the reducers.

To use redux, we need to wrap our entire App with the redux.

Redux thunk is used to do asynchronous actions inside redux.

export const signup = (authData, navigate) => async (dispatch) => {

}

Here async(dispatch) => is the syntax to be followed when using redux thunk.

There are three ways to store the data inside the browser,

1. Storing in cookie
2. Storing in session
3. Storing in localStorage

If useEffect is used inside the App.js file, whenever our application is live the useEffect is gonna run.

post() is used to post a new record.

get() is used to retrieve the record.

patch() is used to update a record.

router.delete("/delete/:id", deleteQuestion);

--- this is because we have to delete the entire record in the database if we delete a question.

router.patch("/delete/:id", deleteAnswer);

--- this is because we just have to delete an object from a record of questions in the database if we delete an answer.

{ $pull: { 'answer': { \_id: answerId } } }

--- this pull property will match a specific id in the answer array and pulls it out.

{question.askedOn}

--- this will output in the format: asked 2022-12-11T10:49:01.938Z

{moment(question.askedOn).fromNow()}

--- this will output in the format: asked 5 minutes ago

Token-based authentication:

If a token exceeds its validity time, then it needs to be expired.

It can be implemented in useEffect().

In css,

px: 40px;

--- this means padding on left and right is 40px.

py: 30px;

--- this means padding on top and right is 30px.

To use fontawesome directly in React install three dependencies in terminal,

>cd client

>npm i @fortawesome/react-fontawesome @fortawesome/free-solid-svg-icons @fortawesome/fontawesome-svg-core

Status Codes in Restful API : https://restfulapi.net/http-status-codes/

map() function returns an array.

Deploying the backend in Heroku.

Deploying the frontend in Netlify.

Using dotenv, we can make few variables confidential, that is those variables will not be moved to github or get deployed.

The variables in .env file should not have values in single or double quotes.

.gitignore file consists of those files which should not be pushed to git.

Create a .gitingore file in sever with the following text,

node\_modules

.env

In index.js (in server) import dotenv and write the following line,

dotenv.config();

The Procfile is especially for Heroku.

Create a README.md file.

In file explorer of our stack-overflow-clone project, in client we will find a .git file, delete it.

Now create a repository named “Stack-overflow” in github.

In terminal in vs code,

Quit the client and server processes.

In a new terminal,

In our project folder,

>git init

>git add .

>git commit -m "first commit"

>git branch -M main

>git remote add origin <https://github.com/Pranavi2002/Stack-overflow.git>

>git push -u origin main

Now in the .gitignore file of server, comment the .env file. This is because .env file is needed when deploying in Heroku.

Also delete the .git file created in the file explorer.

Signup and login to Heroku.

New -> Create a new app -> Give a name and click create.

Download and install the Heroku CLI.

In vscode terminal,

>heroku login

Then you will be redirected to browser where you can login to Heroku cli.

Then in terminal,

>cd server --- as we deploy only backend in heroku

>git init

>heroku git:remote -a stack-overflow-production

>git add .

>git commit -am “make it better”

>git push heroku master

Now we can open app to see our client working.

In client, src, api, index.js

const API = axios.create({ baseURL: "http://localhost:5000" });

--- in the above code in baseURL, add the deployed heroku url by removing / at last. (ex. https://stack-overflow-production.herokuapp.com)

Also in client, src, pages, Questions, QuestionsDetails.jsx,

const url = 'http://localhost:3000'

--- in the above code also give the heroku url.

From now you can see the project backend being deployed in heroku.

>cd ..

>cd client

In client, src, pages, Questions, QuestionsDetails.jsx,

const url = 'http://localhost:3000'

--- in the above code also give the deployed netlify url.

>npm run build or >yarn build (check in README.md file of client)

--- with this a build folder will be created in our client.

Signup and login to the Netlify.

Now drag and drop the build folder in the netlify home page.

Open the site url to view the project, you can change the url if needed.