**Javascript**

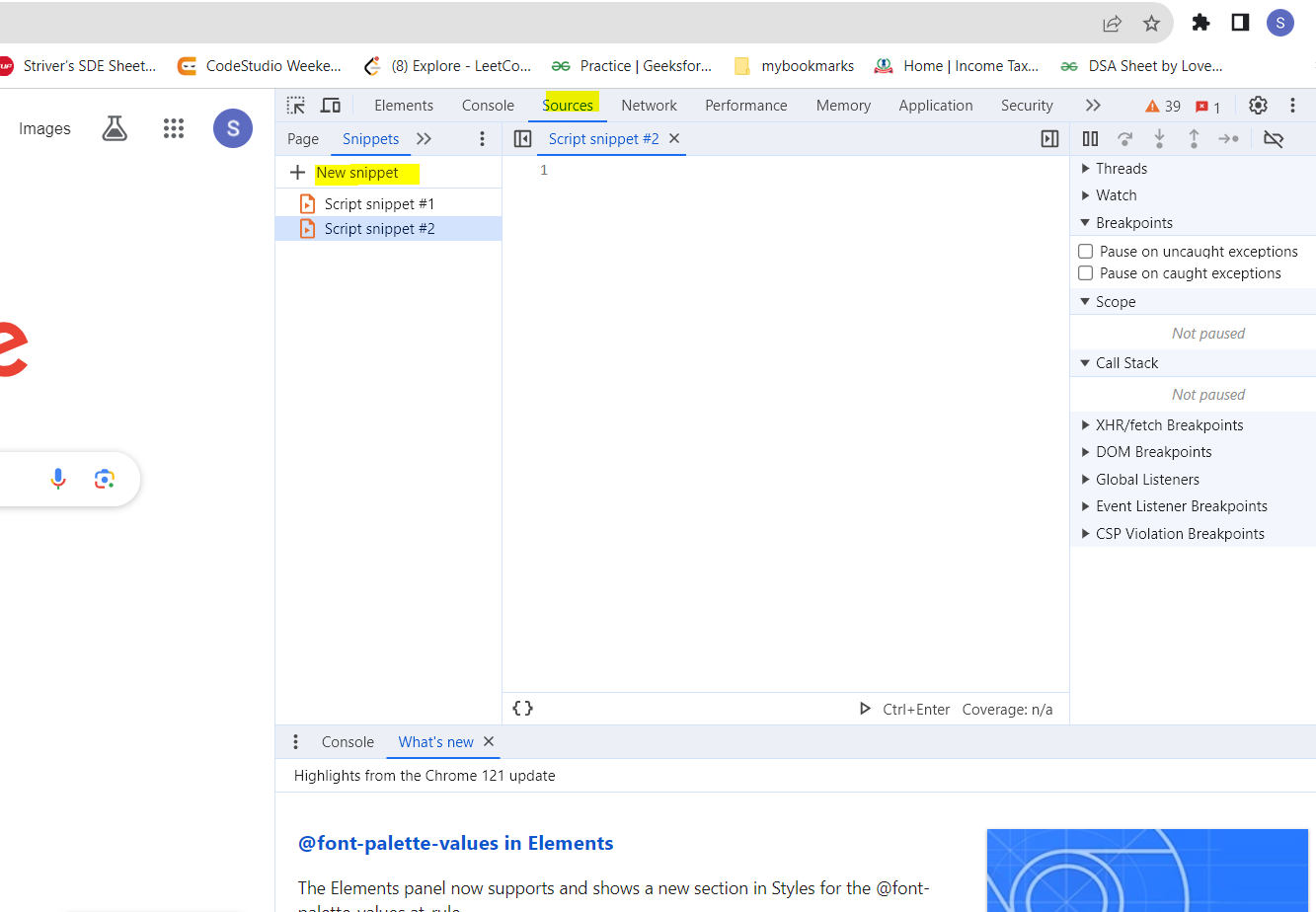
javascript is interpreted language(line by line execution)-python-, ruby

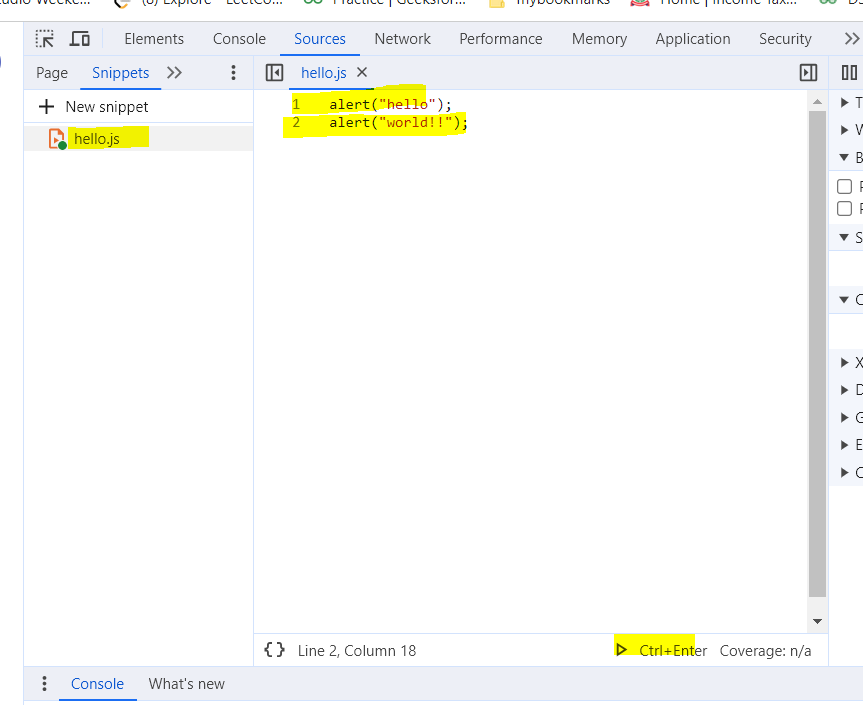
\*java is compiled language (compile whole program together)- c, c++, swift

js- used all places front end to backend used in such frameworks

js- supported by each browser, powers the web, used in website and web app

to write multiline js code in devlopers tool ->sources-> new snippets





\*\*Data types- primitive

1. String
2. Number (433)
3. Boolean true/false
4. Undefined
5. Null

\*\*data Types- Reference

1. Object
2. Array
3. Function

\*\*variables:

* Declared using let, var, const

\*\* Arithmatic and modulo operators

+, -, \*, /

% gives remainder

Precedence: \* >> +

\*\* increment / decrement

X++ or X—

X+=y

\*\*Functions

\*\*comparators

1. === is equal to
2. !== not equal to
3. >, < , >=, <=
4. && AND
5. || OR

Var a = 1, b=”1”

Then (a === b) => false🡪 here its checking data type also

And (a == b)=> true -> here its not checking data type

\*\*Collections

1. Arrays

\*\*Factory Function Vs Constructor function

1. Naming convention
2. Return and this keyword use
3. Directly creating function call object vs using new keyword to create obj.
4. We can use any one method from this. Constuctor one is more familier.

\*\*var Vs let

1. Var scope is not limited to particular block but its limited to whole function
2. Let, const=> block scoped
3. Var=> function scoped
4. Var variable value is assigned to window object..which should not be done

\*\* this keyword

1. The object that is executing the current function

\*\* Arrow function

1. When we need to pass a function as a callback function or as an argument we can pass it as an arrow function in easiest way
2. If no parameter then write empty parenthesis

Arr.filter(() =>

{console.log(“abc”);

});

\*\* destructuring assignments

<https://www.geeksforgeeks.org/destructuring-assignment-in-javascript/>

---------------------------------------------------------------------------------------------------------------------

**DOM**

Html boilerplate ->shortcut to create ->give ! mark and enter

**React-(NetNinja)**

<https://github.com/iamshaunjp/Complete-React-Tutorial/tree/lesson-6/dojo-blog>

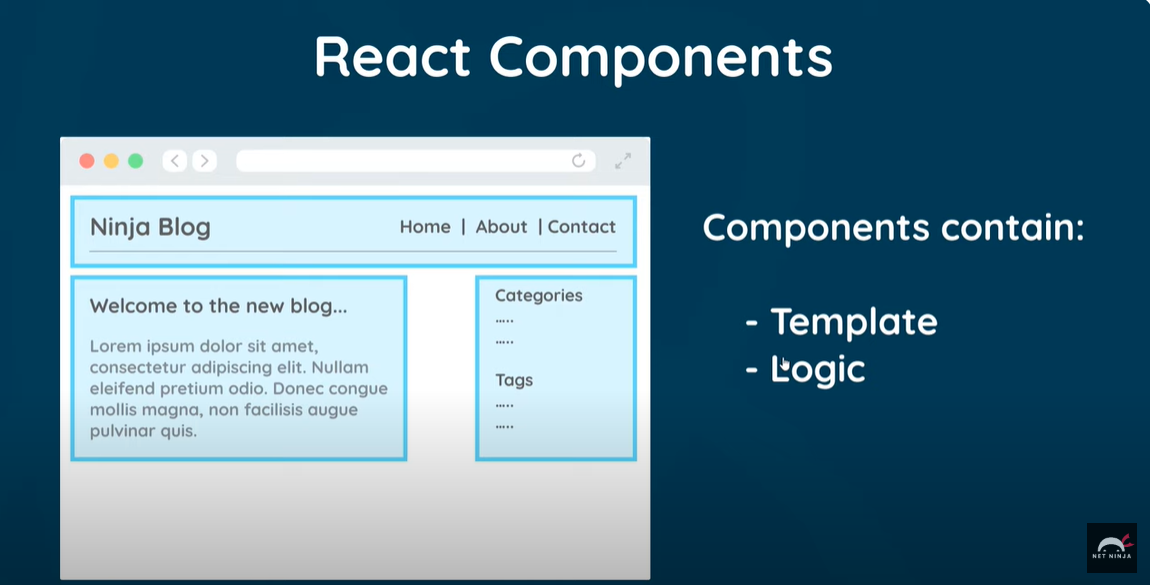
1. Javascript library used to create website
2. We can easily create single page apps ie SPA
3. In tradition application when user click on any link on page e.g. /contacts request will be sent to server everytime. But in react , everytime req is not sent to server , all req are handled by react only

Cd to location

npx create-react-app myapp1

npm install -> to install node module folder

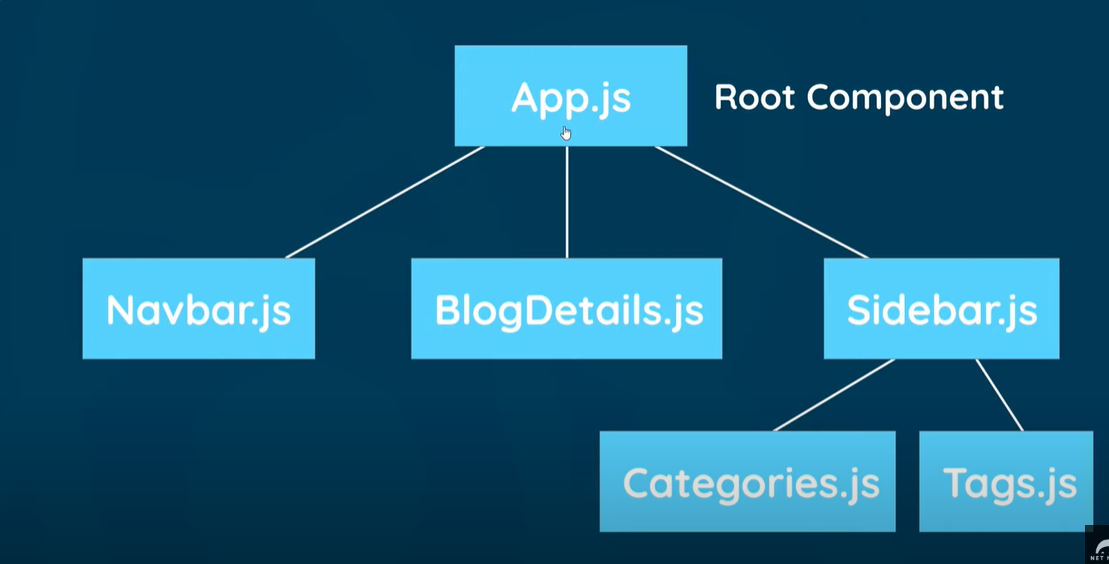
npm run start



App.js-> jsx code=> at runtime it will get converted to html code

**Component** is just a function that returns jsx code.

1. Dynamic values in templates : {}
2. Multiple components



1. Sfc ->to create component boiler plate code
2. UseState Hook

import {useState} from 'react';

const Home = () => {

//let name= 'mario';

   const [name, setName]= useState('mario');

   const [age, setAge]= useState(25);

    const handleClick =() =>{

        setName('Sneha');

        setAge(24);

    }

    const handleClickAgain =(name, e) =>{

        console.log("Hello " + name, e);

    }

    return (

        <div className="home">

            <h2>Homepage</h2>

            <p>{name} is {age} years old.</p>

            <button onClick ={handleClick}>Click me</button>

            <button onClick ={(e) => handleClickAgain('Sneha', e)}>Click me again</button>

        </div>

     );

}

export default Home;

1. Props- reusable code. Used to pass data from parent component to child component
2. useEffect hook – used to run code on every render
3. dependency in useEffect- we don’t want to run useeffect after every render in that case use dependency array. Give state names in array so that after changing those states useeffect will run
4. json server

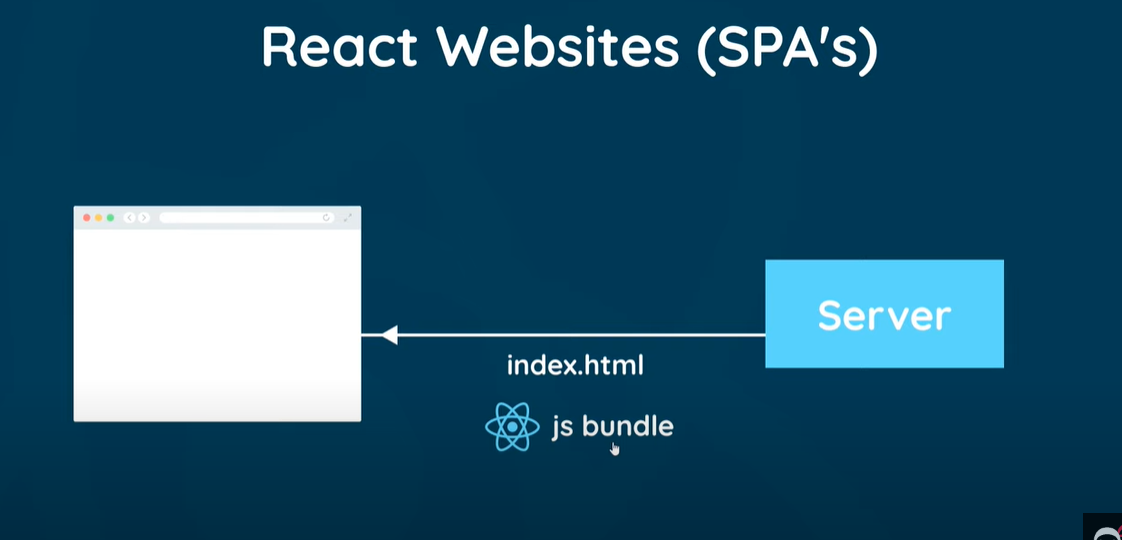
npx json-server - -watch data/db.json - -port 8000

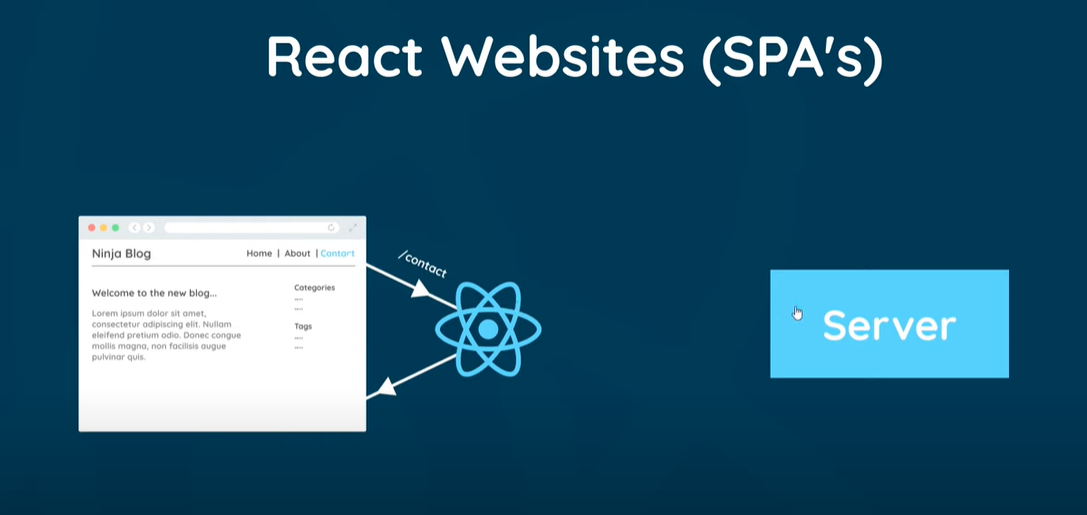
  //https://jsonplaceholder.typicode.com/posts

1. Custom Hook- e.g. useFetch => Crete new .js file. store all code here to fetch data. we can use this code/hook everywhere. always name start with 'use'.
2. React Router- **npm install react-router-dom**

Generally non-react websites sends request to server for each page req and the server send those pages e.g index.html or contacts.html.

But in react we no need to send req to server again nd again .react router will do the work for u.







1. Router Links
2. useEffect cleanup-
3. Route parameters - / blogs/5 (/blogs/:id) -**useParams()**
4. Controlled inputs (forms)
5. Submit events
6. Making a POST request
7. Programmetic redirects –**useHistory()**

* Going back to its previous page/activity

1. Delete blogs
2. **404** pages