React

Basic React Concepts

- Create react file with npx create-react-app reactiitdemo

- Replace app.js with this

- function Greeting({ name }) {

-   return <h1>Hello, {name}</h1>;

- }

-

- export default function App() {

-   return <Greeting name="world" />

- };

Components

```
function MyButton() {
  return (
    <button>
      I'm a button
    </button>
  );
}

export default function MyApp() {
  return (
    <div>
      <h1>Welcome to my app</h1>
      <MyButton />
    </div>
  );
}
```

React apps are made out of *components*. A component is a piece of the UI (user interface) that has its own logic and appearance. A component can be as small as a button, or as large as an entire page.

- JSX is stricter than HTML. You have to close tags like <br />. Your component also can't return multiple JSX tags. You have to wrap them into a shared parent, like a <div>...</div> or an empty <>...</> wrapper:

- HTML to JSxX - https://transform.tools/html-to-jsx

- Add style in css

```css
.avatar {
 border-radius: 50%;
}
```

Use style in jsx using {{}}

style={{}} is not a special syntax, but a regular {} object inside the style={ } JSX curly braces. You can use the style attribute when your styles depend on JavaScript variables.

```
const user = {
  name: 'Hedy Lamarr',
  imageUrl: 'https://i.imgur.com/yXOvdOSs.jpg',
  imageSize: 90,
};
```

```
export default function Profile() {
  return (
    <>
      <h1>{user.name}</h1>
      <img
        className="avatar"
        src={user.imageUrl}
        alt={'Photo of ' + user.name}
        style={{
          width: user.imageSize,
          height: user.imageSize
        }}
      />
    </>
  );
}
```

Concept 3 Rendering list and using logic in style

```
const products = [
  { title: 'Cabbage', isFruit: false, id: 1 },
  { title: 'Garlic', isFruit: false, id: 2 },
  { title: 'Apple', isFruit: true, id: 3 },
];
```

```
export default function ShoppingList() {
  const listItems = products.map(product =>
    <li
      key={product.id}
      style={{
        color: product.isFruit ? 'magenta' : 'darkgreen'
      }}
    >
      {product.title}
    </li>
  );

  return (
    <ul>{listItems}</ul>
  );
}
```

Concept 3

Event handling and useState

Functions starting with use are called *Hooks*. useState is a built-in Hook provided by React. You can find other built-in Hooks in the API reference. You can also write your own Hooks by combining the existing ones.

```
import { useState } from 'react';


function MyButton() {
 const [count, setCount] = useState(0);


 function handleClick() {
  setCount(count + 1);
 }


 return (
  <button onClick={handleClick}>
   Clicked {count} times
  </button>
 );
}
```

```
export default function MyApp() {

 return (

  <div>

   <h1>Counters that update separately</h1>

   <MyButton />

   <MyButton />

  </div>

 );

}
```

Concept 4 – Sharing the event using props

```
function MyButton({ count, onClick }) {
  return (
    <button onClick={onClick}>
      Clicked {count} times
    </button>
  );
}
```

```
export default function MyApp() {
  const [count, setCount] = useState(0);

  function handleClick() {
    setCount(count + 1);
  }

  return (
    <div>
      <h1>Counters that update together</h1>
      <MyButton count={count} onClick={handleClick} />
      <MyButton count={count} onClick={handleClick} />
    </div>
  );
}
```

Using Next Framework

- npx create-next-app

- npm run dev

- App Folder / Router
  - Router based on files system (Folders and Files are used for routing)
  - Layout

- Replate main element with

`<main><h1>Hello IIT</h1></main>`

. Remove line gradient is global css

- Install extension VS Code ES7+ React/Redux/React-Native/JS snippets
- Create users/page.tsx to create a new route
- Ctl +Shift+L
- Create racfe and Create the page
- Create another folder in users
- New
  - Page.tsx with NewUser page

- Modify main page
- Anchor vs Link

- `<Link href="/users">Users</Link>`

Client Side – Server-Side Rendering

- Bundles going to Client

- Search engine can not read

- Use server side to mask sensitive information

- Server generates the page and sends it

- Interactivity is lost

- State

- User server whenever you can and use client side when necessawry

Client-side Rendering

Large bundles

Resource intensive

No SEO

Less secure

Server-side Rendering

Smaller bundles

Resource efficient

SEO

More secure

Client

Server

Share   Info

AddToCart

NavBar

SideBar

ProductList

ProductCard

Pagination

Footer

```
import React from 'react'

const ProductCard = () => {
  return (
    <div>
      <button onClick={() => console.log("Added")}>Add to cart</button>
    </div>
  )
}

export default ProductCard
```

```jsx
import Image from 'next/image'
import Link from 'next/link'
import ProductCard from
'./components/ProductCard'

export default function Home() {
  return (
<main>
  <h1>Hello IIT</h1>
  <Link href="/users">Users</Link>
  <ProductCard />
</main>
  )
}
```

Error

```
'use client';
import React from 'react'

const ProductCard = () => {
  return (
    <div>
        <button onClick={() => console.log("Added")}>Add to cart</button>
      </div>
  )
}

export default ProductCard
```

```jsx
import React from 'react'

const ProductCard = () => {
  return (
    <div>
        <button onClick={() =>
console.log("Added")}>Add to cart</button>
        </div>
    )
}

export default ProductCard
```

```
import Image from 'next/image'
import Link from 'next/link'
import ProductCard from
'./components/ProductCard'

export default function Home() {
  return (
<main>
  <h1>Hello IIT</h1>
  <Link href="/users">Users</Link>
  <ProductCard />
</main>
  )
}
```

Remove 'use client'

```tsx
import React from 'react'

interface User {
  id: number
  name: string
}

const UsersPage = async() => {

 const res = await fetch('https://jsonplaceholder.typicode.com/users')
 const users: User[] = await res.json()
  return (
    <>
      <h1>Users</h1>
      <ul>
        {users.map((user) => (
          <li key={user.id}>{user.name}</li>
        ))}
      </ul>
    </>
  )
}

export default UsersPage
```
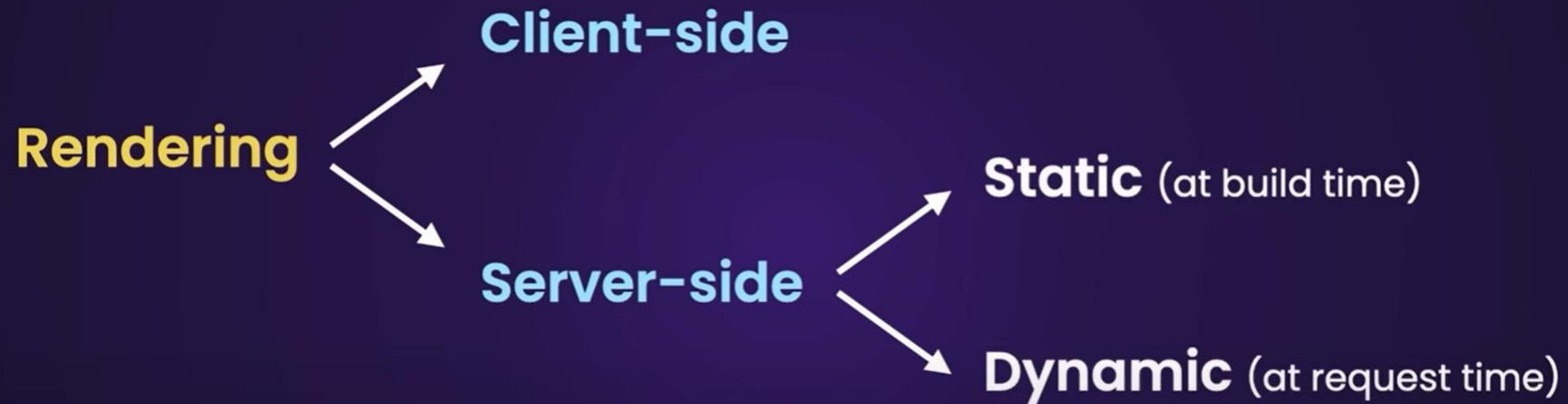
```tsx
import React from 'react'

interface User {
  id: number
  name: string
}

const UsersPage = async() => {

 const res = await fetch('https://jsonplaceholder.typicode.com/users',{next:
{revalidate: 10}})
 const users: User[] = await res.json()
  return (
    <>
      <h1>Users</h1>
      <ul>
        {users.map((user) => (
          <li key={user.id}>{user.name}</li>
        ))}
      </ul>
    </>
  )
}
```

```
https://jsonplaceholder.typicode.com/users',{c
ache: "no-cache"})
```

Global.css

```css
@tailwind base;
@tailwind components;
@tailwind utilities;

:root {
  --foreground-rgb: 0, 0, 0;
}


@media (prefers-color-scheme: dark) {
  :root {
    --foreground-rgb: 255, 255, 255;
  }
}


body {
  color: rgb(var(--foreground-rgb));
  padding: 2rem;
}
```

```jsx
import React from 'react'
import AddToCart from './AddToCart'
import styles from './ProductCard.module.css'

const ProductCard = () => {
  return (
    <div className= {styles.card}>
        <AddToCart />
        </div>
  )
}

export default ProductCard
```

Tailwindcss.com

Color pallet

Paddings

p-[number]

px-[number]

py-[number]

pt-[number]

pr-[number]

pb-[number]

pl-[number]

Margins

m-[number]

mx-[number]

my-[number]

mt-[number]

mr-[number]

mb-[number]

ml-[nu

RE VIDEOS

Size

text-xs

text-sm

text-base

text-lg

text-xl

text-2xl

text-3xl

Color

text-[color]

bg-[color]

Thickness

font-thin

font-light

font-normal

font-medium

font-bold

RE VIDEOS

Tailwind intellisense extension

```jsx
import React from 'react'
import AddToCart from './AddToCart'

const ProductCard = () => {
  return (
    <div className = 'p-2 my-2 bg-sky-600 text-white text-xl hover:bg-sky-800 '>
        <AddToCart />
        </div>
  )
}

export default ProductCard
```

npm i -D daisyui@latest

```
'use client';

import React from 'react'

const AddToCart = () => {
  return (
    <button className='btn btn-primary'
onClick={() => console.log("Added")}>Add to
cart</button>
  )
}

export default AddToCart
```

```tsx
import React from 'react'

interface User {
  id: number;
  name: string;
  email: string;
}
```

```jsx
  const res = await fetch('https://jsonplaceholder.typicode.com/users',{cache: "no-cache"})

  const users: User[] = await res.json()
   return (
     <>
       <h1>Users</h1>
       <p>{new Date().toLocaleTimeString()}</p>
       <table className='table table-bordered'>
         <tr>
           <th>email</th>
           <th>Name</th>
         </tr>
         <tbody>
           {users.map((user) => (
             <tr key={user.id}>
               <td>{user.email}</td>
               <td>{user.name}</td>
             </tr>
           ))}
         </tbody>
       </table>
     </>
   )
}


export default UsersPage
```