React Chaitanya notes :-

Why React

disadvantages of J.S

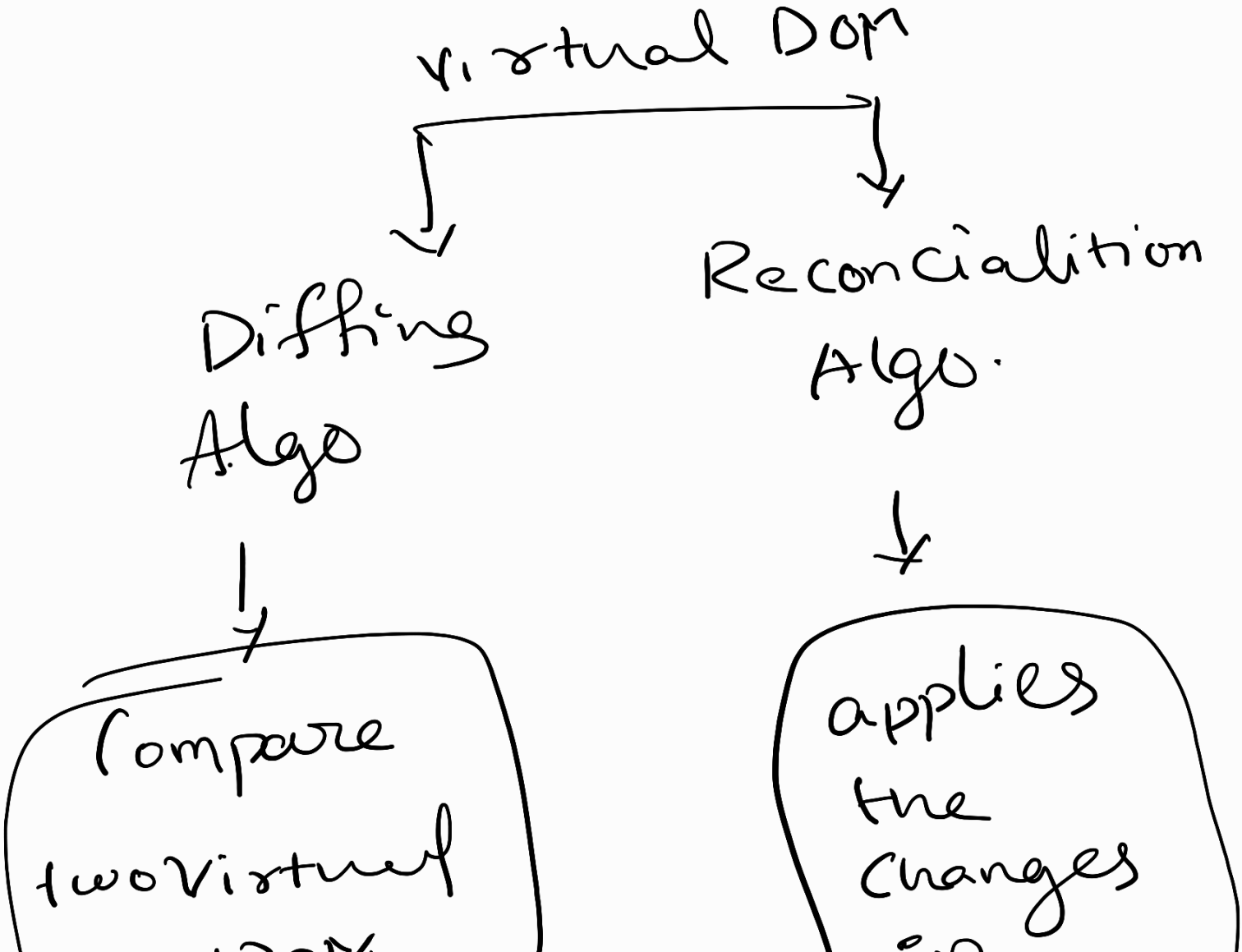
1. Complexity in state mangment (cross browsel compatabiliy)
2. Inconsistent Boowsel Support
3. Difficut in code organisation
4. Performance Concerns
5. Callback Hell.
6. Directly interact with DOM

Why moving to React :-

1. Component -Based

Architecture

- Virtual DOM
- State Management.
- Declarative Syntax
- Ecosystem & Community



DOM

real DOM

Why only ? React when Compare to othel Lib/Frame

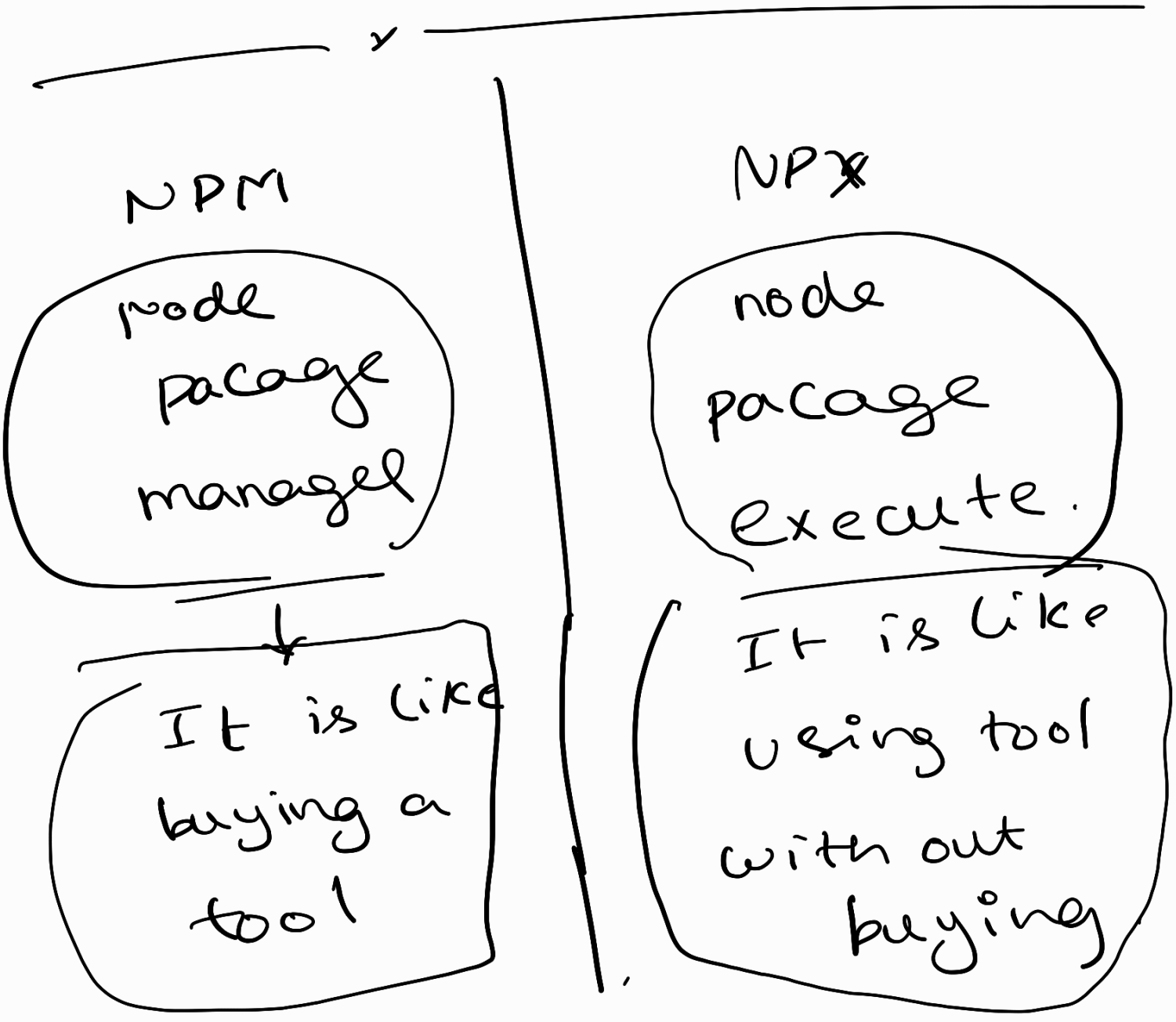
- Component based Architecture
- Flexibility & Ecosystem
- Virtual DOM for performance
- Declarative Syntax.
- Rich Ecosystem
- Jsx syntax

Javascript xml notion

- popular for Modern development

① Learning curve.

② support for mobile development.



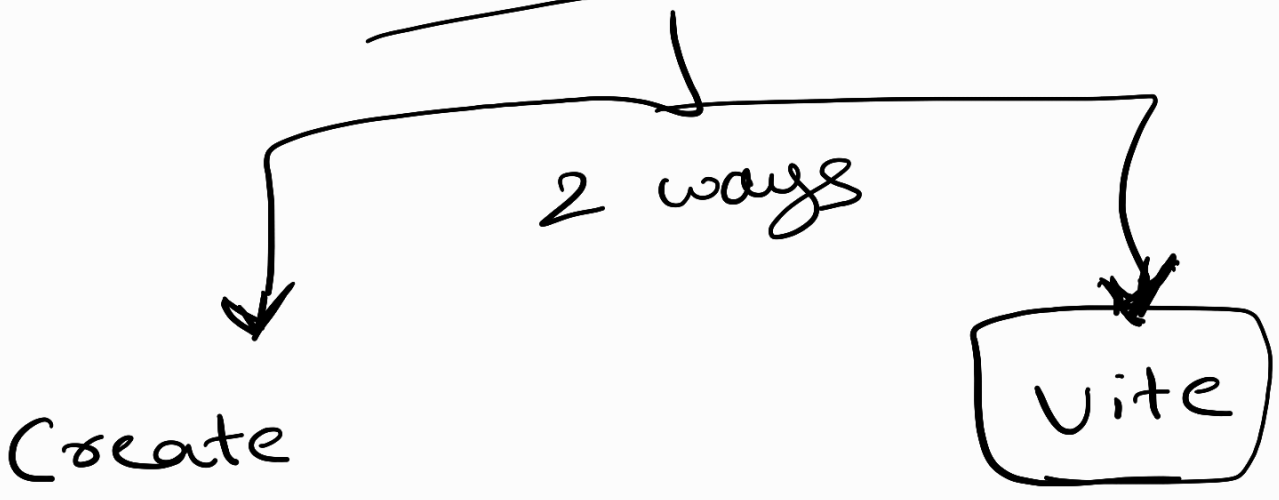
→ Requirements →

→ Node JS ✓

→ NPM | NPX ✓

→ Code editors ✓

Creating React APP



→ npx Create-React-APP my-app

→ cd my-app

→ npm start

folder structure :-

Node modules ✓

public

src

.gitignore. ✓

package-lock.json ✓

package.json . ✓

Readme.md. → ✓

Readme → we can write

project info

like setup instructions

usages,

other instructions

gitignore → ignores the files & folders

package.json → contain metadata about project

node_modules → all packages dependencies

By vite.

npm create vite@latest

then press Y

then enter project name

Select framework → React

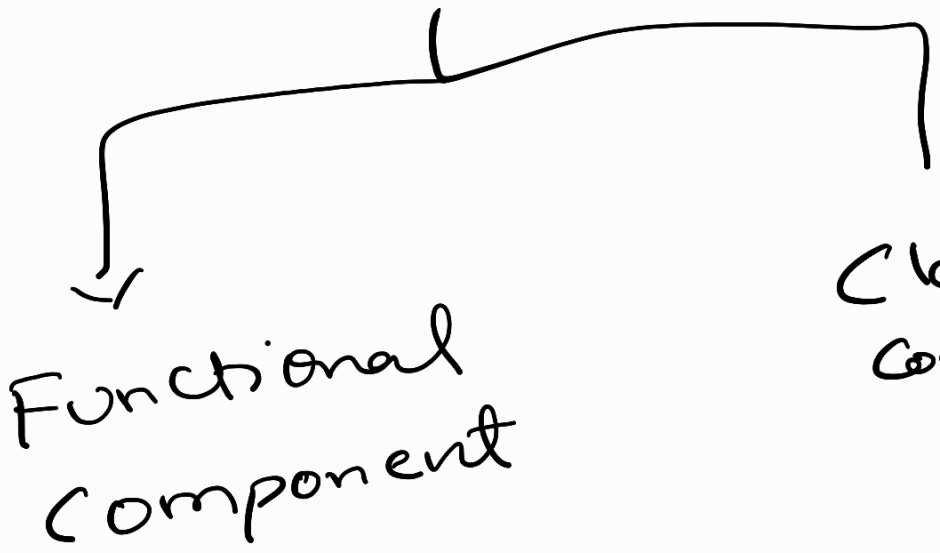
select variant → Javascript.

```
cd Projectname
npm install
npm run dev.
```

Components :-

Component is resuable bits

of code which creates user interfaces (UI)



Rules

- every component will return Html type of code called as JSX

- component should start with Capital letter

- component can be as small as button & as large as screen

⊙ nested components can be formed to create entire application ✓

JSX ↴

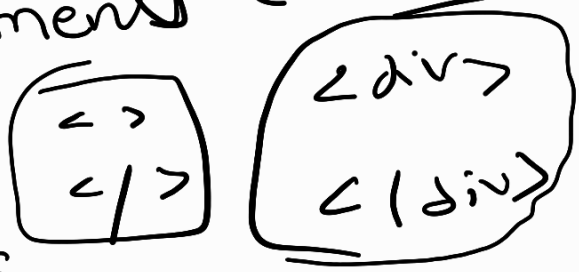
Javascript xml Notation

Because of JSX we can write JS and html content in one file.

Rules :-

every component should return the single parent,

that parent Container can be div, Fragments etc...



React Fragments

↳ Empty shorable Container.

it don't create extra node in the dom

① Every Attribute must be Camelcase e:- onClick etc.

② Every J.S code must be in { } curly bases inside return

③ class → Classnam.

- use expressions in JSx but not statements.

Like (ternary operator can be used)

- Comments →

{/* */}

expression | statements

val a = 10 → statement

→ expression.

Props → properties

State lifting

unidirectional flow

Pass data from one compo → another component

typically parent → child.

→ Immutable ✓

→ Unidirectional dataflow

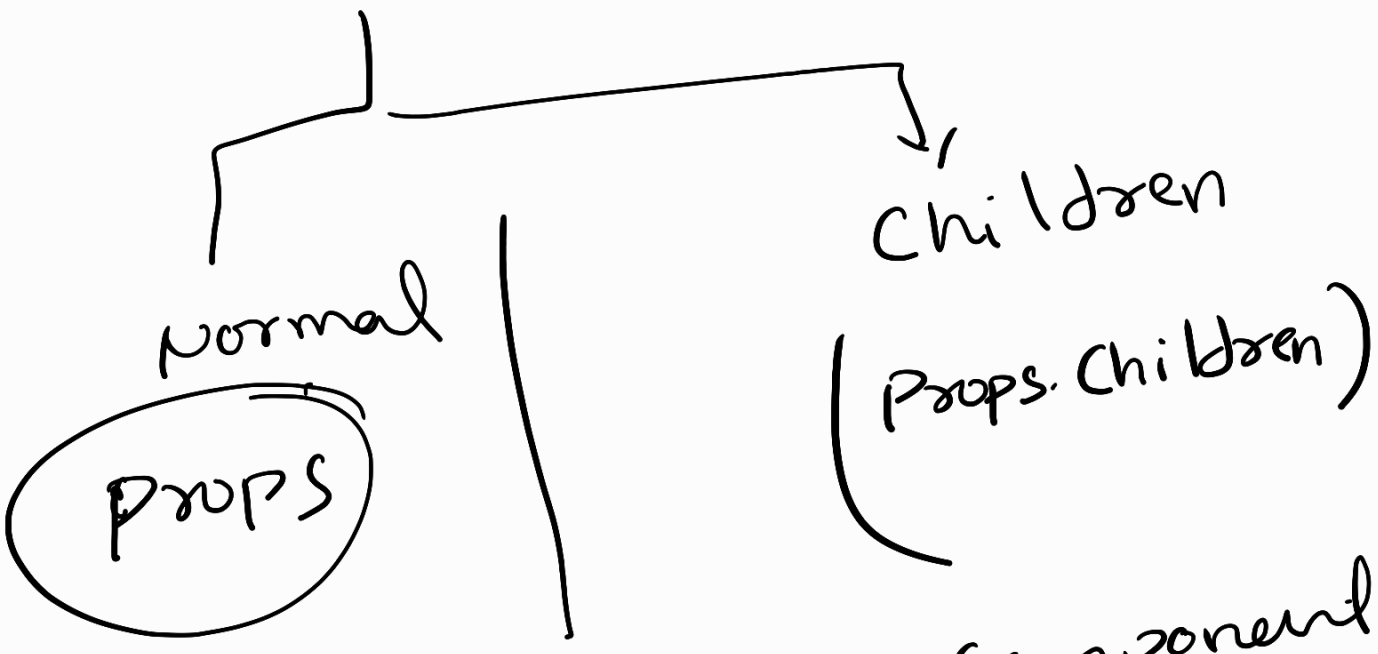
→ props can be any data

number, object, strings & even Component.

lifting

→ State

Two types of props →



→ Component.

→ Data.

→ HTML (elements)

ES6 Concepts

Map → using map how we render the data:

Types of Styles in React

- Inline | javascript object styles
- external css styling.
- external modules styling
- External Library styling.

→ Inline Styling :-

```
const style = {
    color : "blue",
    fontSize : "20px",
```

}

① <div style={{color: 'red'}}>

↓ replace with J.S object.

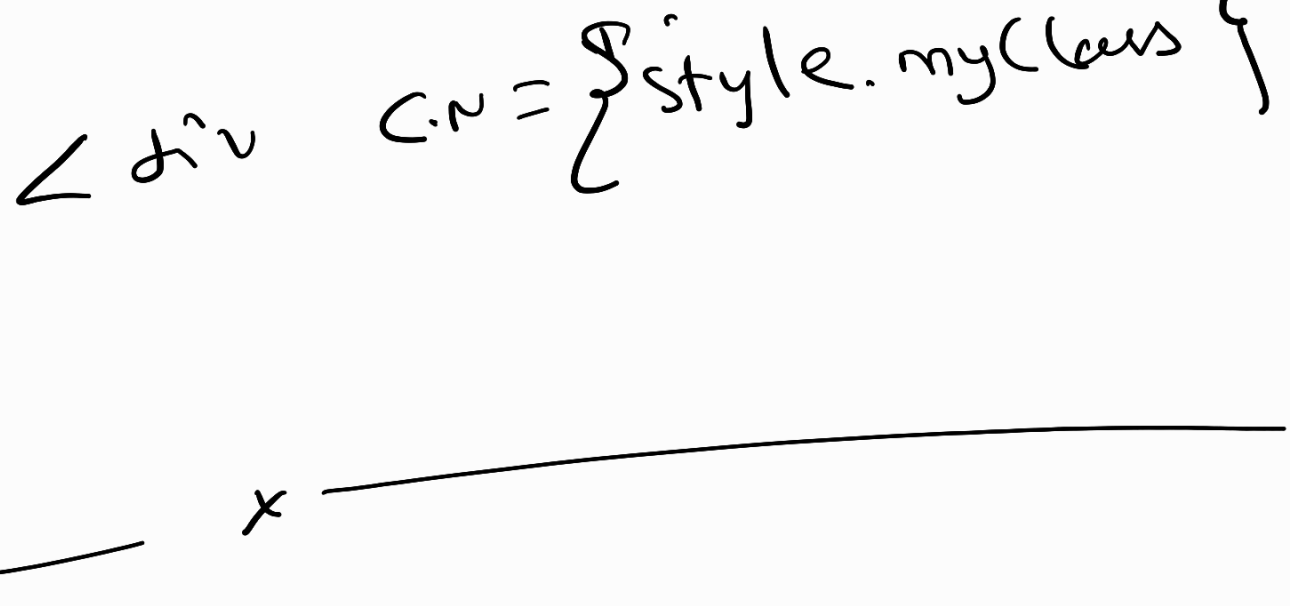
② Stylesheets

import style sheets into Component

import './style.css'

③ Module name as style.module.css

import style from filename


< div c.n = {`style.myClass}