**REACT**

* Java script library
* Group of reusable components
* Speed due to virtual DOM (changes only changed component). That is nav and header footer mostly does not change only content

**Command line commands:**

1. Pwd->full path name
2. Ls->list of directories
3. Mkdir->new directory
4. Cd->move into folder
5. cd..->backwards move outside folder
6. clear-> clear console
7. npm init->create package.json(manifest file), list dependencies
8. npm install <package name> --save -> install package locally and add to package.json. –save to add to package.json in npm version <5.0
9. npm install <package name> -g -> install globally
10. npm install <package name> --save-dev=> only for development
11. npm create-react-app <app name> =>to create react app with folder structure
12. npm start=> run react app. Start development server
13. npm run build -> bundles app into static files

**Babel => website which converts to new js**

**Folder structure:**

Node modules=> all packages installed and dependencies

Public=> index.html, css, logo

Index.html=> the root element <div id=”root”></div> is present

Build=>after npm run build .used for deployment

Src=> App.js(root component) and all component , css and index.js

.gitignore=> folder and files to not be added to git

Remove all needless files………logo…..

**RULES:**

1. Attributes of tags in **camelCase** => onClick
2. All components **first letter** capital => App
3. **All tags enclosed** inside single element and only one element to be returned
4. **className** instead of ~~class~~
5. instead of div for wrapping all element can also use <**React.Fragment**> which will wrap and return as single element but does not reflect in html as a div.
6. All element must be **closed** even those without close tag => <div></div> or <br/> or <App/>
7. {} inside component to start javascript inside html tags => <div> hello {user\_name}</div>
8. {{}} inside tags for styles=> <div style={{‘color’:’red’,’backgroundColor’:’yellow’}}>
   1. No – only camelCase
   2. No ; for end of line , for separating styles
9. If value undefined only empty (null) returned inside {}
10. While building component for each element of array must give attr key as unique value.
11. Destructing object

Eg: <App a=’one’ b=’two’/>

* 1. Can be done in parameter itself 🡪App=({a,b})=>{}
  2. Can be done inside function 🡪App=(props)=>{const {a,b}=props;}
  3. Can be done from tag itself 🡪 const p={a:’a’,b:’b’}

<App {…p}/>

* 1. The name must be same as object keys => const {l,b}=props 🡪 undefined since props has only a=’one’ and b=’two’

1. Destructing array 🡪 const a=[1,2,3,4]

* Const [a1,a2,a3,a4]=a;

1. **ALL EVENTS** supported by REACT => React events in docs
2. **Import and export**
   1. Export <data>

Eg:

Const a=’a’;

Var b=’b’;

Export default {a:a,b:b};

* 1. Only one default export per file……..can be imported by any name
  2. If not default export ……then must import with same name
  3. Import {data} from ‘file path’;

1. **Hooks**

Start with –use—

Must be inside component

The component must be capitalize

Cannot call conditionally …only the function returned can be called conditionally

* 1. UseState -----function of React package

Imported from ‘react’ package

Parameter is intial value

Returns the value and function that can be used to change that value

The function that is returned has parameter value which when called will change the value of variable (even for const variable)

* 1. useEffect----function of React package

by default runs after every re-render

like componentUpdate but for specific value

[] for only intial render in dependencies

useEffect can return a cleanup function which will be run first in the next call

1. **Conditional Rendering**

Return different contents based on the condition specified

1. **|| and &&**

A=”…..” b=”……”

C=a || b; 🡪 c=a if a is true(has characters) else c=b

D=a&&b 🡪d=b if a is true else d=””(none);