# React Course

## Chapter 1 : Getting started

React is a javascript library for building user interfaces.

> Components are essentially a piece of the UI.

The building process is about merging the multiple components to create a complexe Interface.

Every react application has a route component. It represent the entire application with other child component. So every application is a tree of component.

Every component is build like a javascript class.

```
class tweet {
  state = {};
  render() {
    // react element
  }
}
```

THe react element is mapping the dom element. React keep a lightweight representation of the dom in memory. We no longer need to work with the dom api.

React reacts to state of changes.

> Angular vs React: They have similar components. But Angular is a framework for complete solution and React is a library. And react only take care of rendering the vue and making the vue is in sink with the state. That's why it has very small api to learn. When buildidng an application with react, we need to use other libraries for routing, calling http services and so on.

### Setup

- install nodejs
- reactjs
- `npm i -g create-react-app` // @tospecifyaversion
- Vscode extension: simple React snippet and Prettier code formatter (editor format on save = true).

### First react app

Let's create our first react app.

```
# For installation
# $ create-react-app <project path>
$ create-react-app React/react-app

Creating a new React app in /Users/alexandrodisla/Desktop/repo/formation/Mosh/React/React/react-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts...


> fsevents@1.2.7 install /Users/alexandrodisla/Desktop/repo/formation/Mosh/React/React/react-app/node_mo
> node install

[fsevents] Success: "/Users/alexandrodisla/Desktop/repo/formation/Mosh/React/React/react-app/node_modul
```

```
> fsevents@1.2.4 install /Users/alexandrodisla/Desktop/repo/formation/Mosh/React/React/react-app/node_m
> node install

[fsevents] Success: "/Users/alexandrodisla/Desktop/repo/formation/Mosh/React/React/react-app/node_modul
Pass --update-binary to reinstall or --build-from-source to recompile
+ react-dom@16.8.4
+ react@16.8.4
+ react-scripts@2.1.8
added 1973 packages from 735 contributors and audited 36228 packages in 351.694s
found 63 low severity vulnerabilities
  run `npm audit fix` to fix them, or `npm audit` for details

Success! Created react-app at /Users/alexandrodisla/Desktop/repo/formation/Mosh/React/React/react-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can’t go back!

We suggest that you begin by typing:

  cd /Users/alexandrodisla/Desktop/repo/formation/Mosh/React/React/react-app
  npm start

Happy hacking!
```

we will get a lightweight developpment server, webpack for bundeling our files and babel for compiling our javscript code etc. However if you need to do some configuration on your code, ou can do `npm run eject`.

To start the app

```
# on the directory of the app
$ npm start
# to quit
$ control + c
```

> In our react-app folder, we have the node_modules which contains the third party libraries and react itself. The public folder is where we have the public asset of our application.

In the index.html, we will see the div tag `<div id='root'></div>`. It’s the container for running our application.