**NOTE 3:**

Q: how to rename folder in terminal?

A: mv old\_name new\_name

Node server.js : to start the backend server

**PROPS:**

Look at this custom attribute

<User username=’Bryan’ />

* This is the same as giving a function an argument to call it: Function username(Bryan)

Now we can access that attribute by using this.props

Ex:

class User extends React.Component {

render(){

return(

<p> Username: {this.props.username} </p>

<p> Is Friend?: {this.props.friend}</p>

)

}

}

<User **username=’Bryan’ friend={ true}**/>

NOTE: {} denotes Javascript expression

Highlighted in red is the props of that **class**

**CREATE-REACT-APP:**

1)

\_Sudo npm –g install Create-React-App

\_Then create-react-app project\_name

This will give you an src file with App.cs and App.js

2) But if you clone it from the Facebook repo here <https://github.com/facebookincubator/create-react-app>

Those won’t have that, but it’s unnecessary to clone since it’s just Facebook backend files.

create-react-app my-app

**TERMINAL:**

\_mkdir creates a folder/ directory

\_To remove empty folder, use rm –r name

\_To create an empty file, use touch. Ex: touch filename

\_Install Shell Command on Atom: Use cmd + shift + P

\_To open a file, just use atom filename

\_You can create an empty file with atom using atom newfilename. After putting content into that file, you can save it, then it’ll show up

**REACT:**

\_We can store state in a component

Q: Why do I need to do **import React, { Component} from 'react'**?

A: First, we import the React module from the external react module

We use {Component} so that we will have access to use Component as opposed to writing React.Component

Ex: Without the {Component} imported:

class ListContacts extends React.Component{}

But with it:

class ListContacts extends Component{}

Q: Why do we need to extend React.Component class?

A: (not totally confirmed)

But we do it so that we can initialize the constructor and change by using this.state