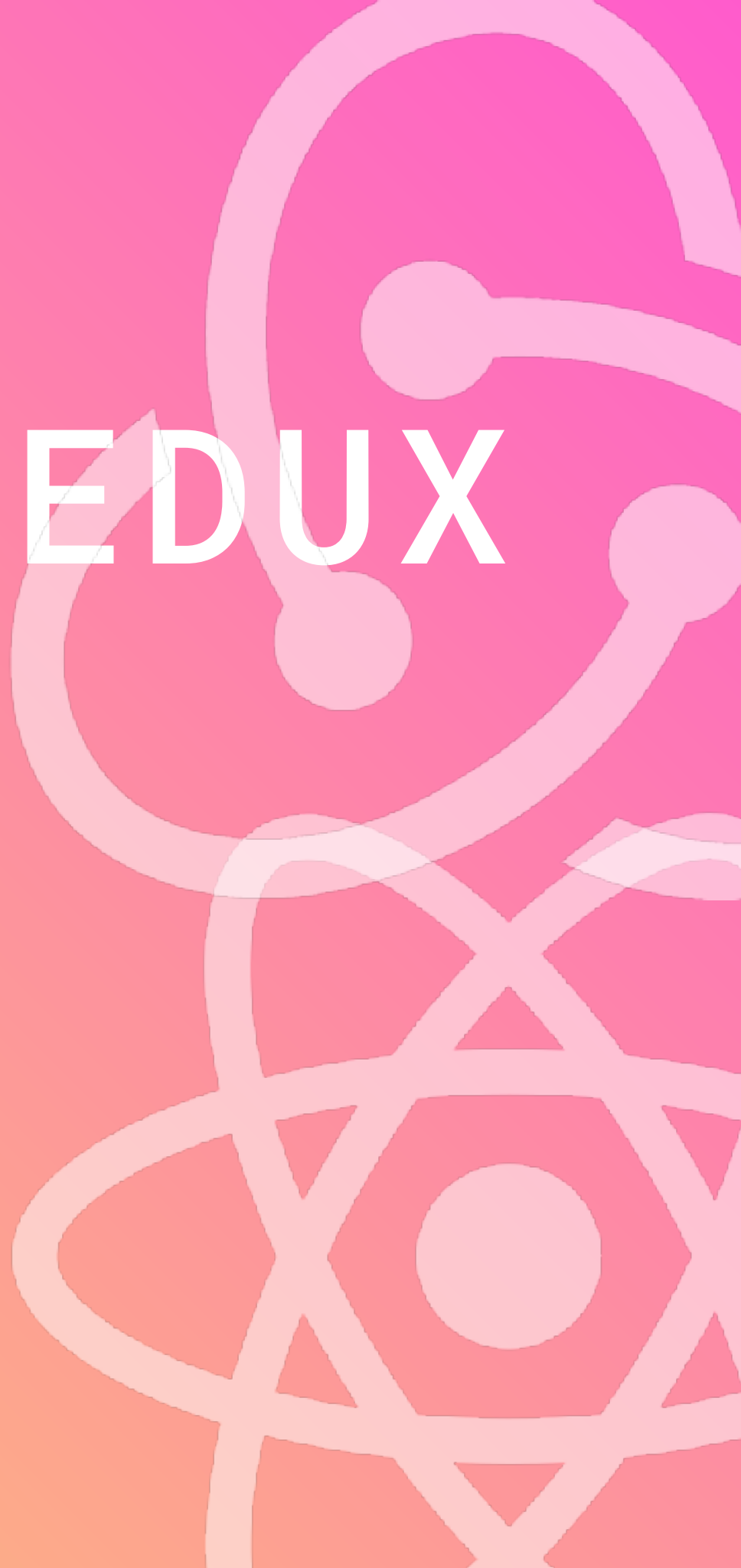


# INTRO TO REACT & REDUX

@OLARCLARA



Chrome

File Edit View History Bookmarks People Window Help

localhost:3000

To Do App

To Do List

Insert here your to do!

ADD

Elements

Console

Sources

React

Highlight Updates

Highlight Search

Search (text or /regex/)

<Provider store={dispatch: dispatch(), subscribe: subscribe()},>

<App>

<Container className="App"> == \$r

<div className="ui container App">

<Header>...</Header>

<Connect(AddTodo)>...</Connect(AddTodo)>

<Connect(TodoList)>...</Connect(TodoList)>

</div>

</Container>

</App>

</Provider>

Provider

App

Container

Props read-only

> children: Array(3)

className: "App"

/Users/mariacларasantana/Documents/react-intro/src/components/App.js:8

Console

top

Filter

Default levels

Hide network

Log XMLHttpRequests

Preserve log

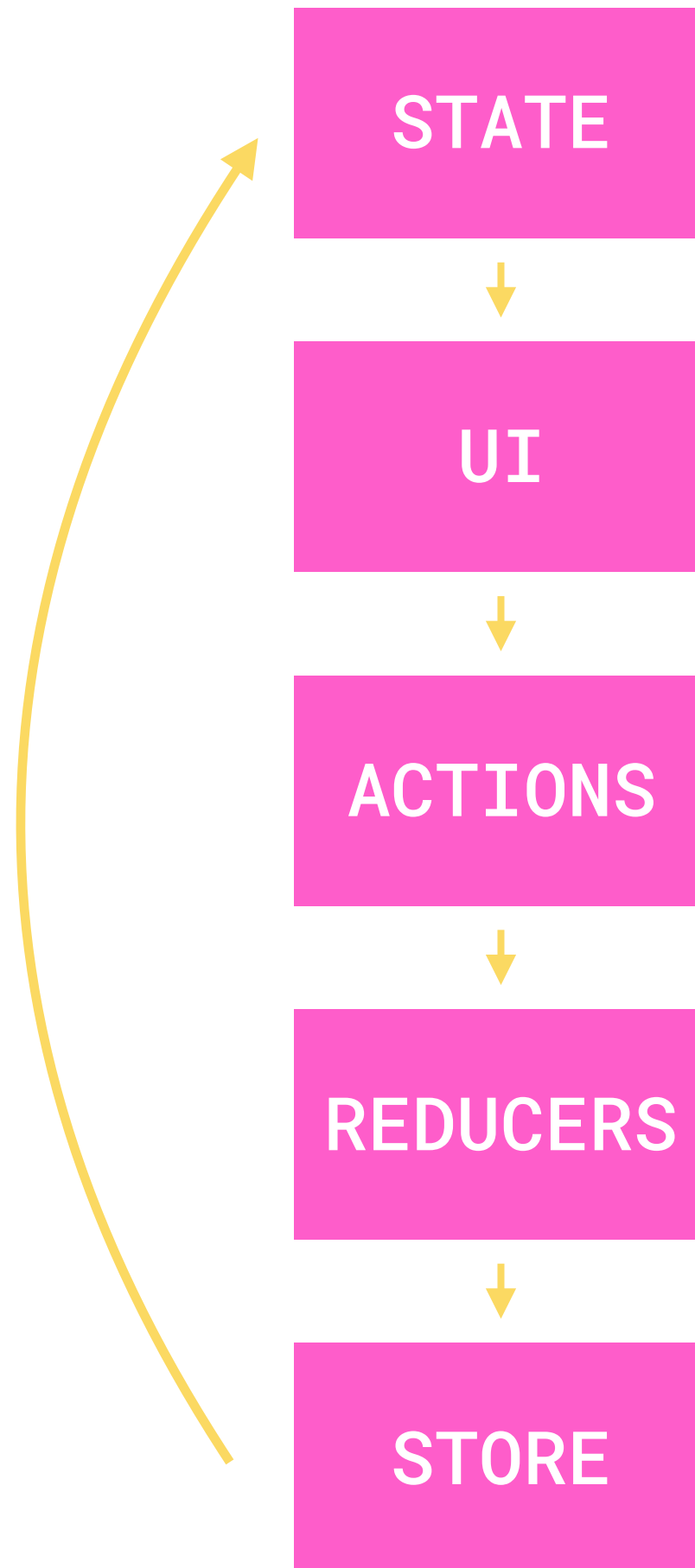
Show timestamps

Selected context only

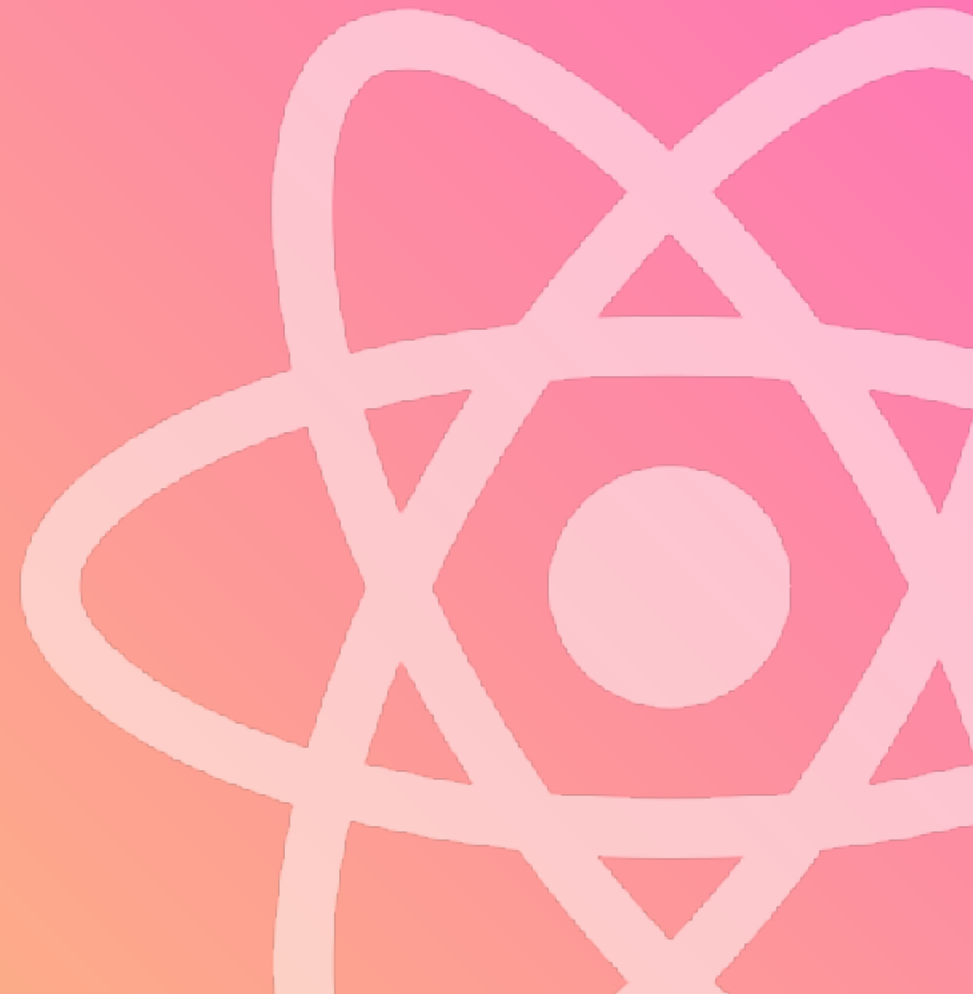
Autocomplete from history

User messages only

>



# REACT COMPONENTS



```
import React, { Component } from 'react'

class App extends Component {
  render() {
    return(
      <div className="App">
        <h1>Simple Stateless React Component</h1>
      </div>
    )
  }
}
```

# CAVEATS

- Components must return a single root element;
- All React components must act like pure functions with respect to their props;

# STYLING REACT COMPONENTS 🚧



```
$ npm i --save-dev semantic-ui  
$ cd semantic  
$ gulp build
```

```
// on index.js  
import './semantic/dist/semantic.min.css'
```

---

```
$ npm i --save semantic-ui-react
```

```
// example  
import React from 'react'  
import { Button } from 'semantic-ui-react'
```

```
const ButtonExampleButton = () => (  
  <Button>  
    Click Here  
  </Button>  
)
```

```
export default ButtonExampleButton
```



# INTEGRATING WITH REDUX



```
$ npm i --save react react-redux
```

---

```
const initialState = []

const todos = (state = initialState, action) => {
  switch (action.type) {
    case 'ADD_TODO':
      return [
        ...state,
        {
          id: action.id,
          content: action.content,
          date: action.date
        }
      ]
    default:
      return state
  }
}

export default todos
```

```
let todoId = 0;
```

```
export const addTodo = (content, date) => ({  
  type: 'ADD_TODO',  
  id: todoId++,  
  content,  
  date  
})
```

```
import React from 'react'
import { render } from 'react-dom'
import { createStore } from 'redux'
import { Provider } from 'react-redux'
import App from './components/App'
import reducer from './reducers'

const store = createStore(reducer)

render(
  <Provider store={store}>
    <App />
  </Provider>,
  document.getElementById('root')
)
```

```

import React, { Component } from 'react'
import moment from 'moment'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'
import { addTodo } from '../actions'
import { Form } from 'semantic-ui-react'

export class AddTodo extends Component {
  constructor(props) {
    super(props)
    this.state = { todo: '' }
    this.handleChange = this.handleChange.bind(this);
    this.handleAddTodo = this.handleAddTodo.bind(this);
  }

  handleChange(e) { this.setState({ [e.target.id]: e.target.value }) }

  handleAddTodo(e) {
    if (this.state.todo.length ≤ 0) return

    let date = moment().format('DD-MM-YYYY')
    this.props.addTodo(this.state.todo, date)
    this.setState({ todo: '' })
  }

  render() {
    return(
      <Form>
        <Form.TextArea id='todo' value={this.state.todo} onChange={this.handleChange}
          autoHeight placeholder='Insert here your to do!' />
        <Form.Button type='submit' onClick={this.handleAddTodo}>ADD</Form.Button>
      </Form>
    )
  }
}

const mapDispatchToProps = dispatch ⇒ bindActionCreators({
  addTodo: addTodo
}, dispatch)
export default connect(null, mapDispatchToProps)(AddTodo)

```