



# React Intro

CUNY Tech Prep (2019)

Edgardo Molina, PhD | Head Instructor

---

React is a  
*declarative* and *component-based*  
front-end library for building  
interactive applications



# React is

**Interactive Applications:** are dynamic and/or animated applications that immediately respond to a user's input without requiring the page to reload, instead the DOM is manipulated with JavaScript.

**Declarative:** in React we describe our applications as views that correspond to our applications state changes. And user input events manipulate the applications state. The React library does the actual work of changing our DOM.

**Component-based:** in React we develop our front-end as small, simple, testable, and reusable components that can be composed into larger complicated applications.

---

React was developed by Facebook and later open sourced. It is now developed and used by a larger community

Learn more about the History of React.js:

<https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/>



# Why React?

## What about other libraries/frameworks?

jQuery, Angular, Vue, etc...

Good too, but React offers the following in a stable library:

- Makes rendering highly dynamic and interactive apps **fast**
- Makes developing interactive apps better/easier ... (*Subjective*)
- Has a *large community* developing and supporting it
- React can be used in Desktop and Mobile apps via *React Native*

# Thinking in React



Thinking in React:

<https://reactjs.org/docs/thinking-in-react.html>

---

Instead of thinking of and developing  
HTML/CSS/JS for your entire app at once,  
we want to break it down into simpler (and  
possibly *reusable*) components.



Given the following Mockup:

☐ Only show products in stock

Name	Price
<b>Sporting Goods</b>	
Football	\$49.99
Baseball	\$9.99
<b>Basketball</b>	\$29.99
<b>Electronics</b>	
iPod Touch	\$99.99
<b>iPhone 5</b>	\$399.99
Nexus 7	\$199.99

The working app:

<https://reactjs.org/docs/thinking-in-react.html#step-5-add-inverse-data-flow>





We want to break it into simpler components:

☐ Only show products in stock

Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

We want to break it into simpler components:

Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

1. **FilterableProductTable** (orange):  
contains the entirety of the example
2. **SearchBar** (blue):  
receives all user input
3. **ProductTable** (green):  
displays and filters the data collection  
based on user input
4. **ProductCategoryRow** (turquoise):  
displays a heading for each category
5. **ProductRow** (red):  
displays a row for each product

---

Read the *Thinking in React* article:

<https://reactjs.org/docs/thinking-in-react.html>

# Let's learn React hands on:

<https://gist.github.com/medgardo/9d328a3b0029965158186cbdf5e3a42>





# React Main Concepts review

1. [Hello World](#)
2. [Introducing JSX](#)
3. [Rendering Elements](#)
4. [Components and Props](#)
5. [State and Lifecycle](#)
6. [Handling Events](#)
7. [Conditional Rendering](#)
8. [Lists and Keys](#)
9. [Forms](#)
10. [Lifting State Up](#)
11. [Composition vs Inheritance](#)
12. [Thinking in React](#)

Can also learn interactively at freecodecamp:

<https://learn.freecodecamp.org/front-end-libraries/react/>



---

Do the official *Tic Tac Toe tutorial*:

<https://reactjs.org/tutorial/tutorial.html>

Try it out locally with `create-react-app`

The final result:

<https://codepen.io/gaearon/pen/gWWZgR?editors=0010>



## Additional Resources

More about Immutability:

<https://medium.com/pro-react/a-brief-talk-about-immutability-and-react-s-helpers-70919ab8ae7c#.fa28ulxbm>

JSX In Depth: <https://reactjs.org/docs/jsx-in-depth.html>