

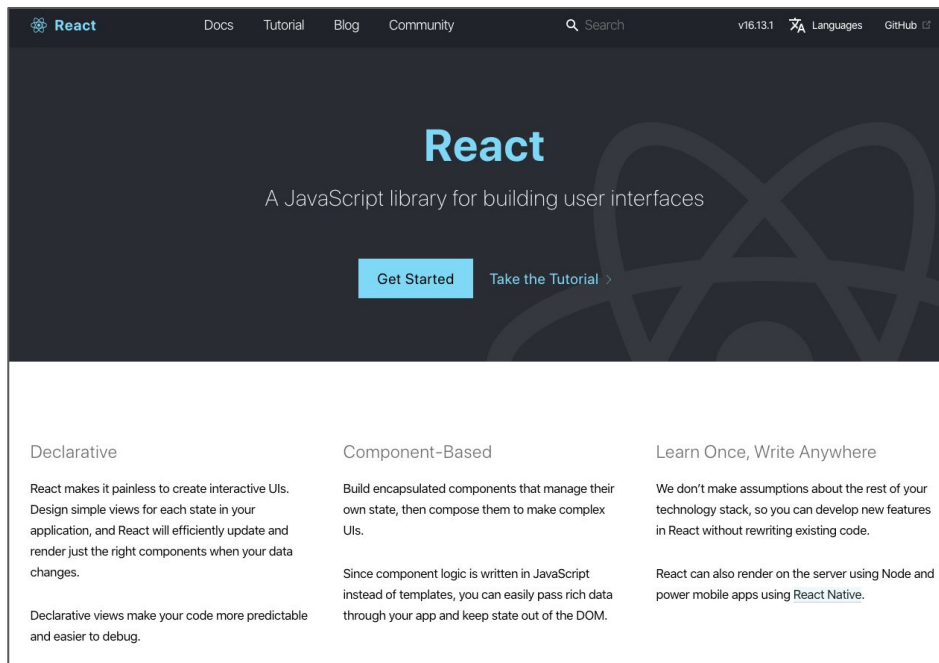
Tutorial #4

React

2020. 04. 14

[CS374] Introduction to HCI

What is React?



The image shows a screenshot of the React website homepage. The top navigation bar includes links for Docs, Tutorial, Blog, and Community, along with a search icon and the text 'v16.13.1 Languages GitHub'. The main content area features the React logo in blue, followed by the text 'A JavaScript library for building user interfaces'. Below this, there are two buttons: 'Get Started' and 'Take the Tutorial >'. The bottom section of the page is divided into three columns, each with a heading and a paragraph of text.

React
A JavaScript library for building user interfaces

[Get Started](#) [Take the Tutorial >](#)

Declarative
React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.
Declarative views make your code more predictable and easier to debug.

Component-Based
Build encapsulated components that manage their own state, then compose them to make complex UIs.
Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

Learn Once, Write Anywhere
We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.
React can also render on the server using Node and power mobile apps using React Native.

The most popular JavaScript Framework

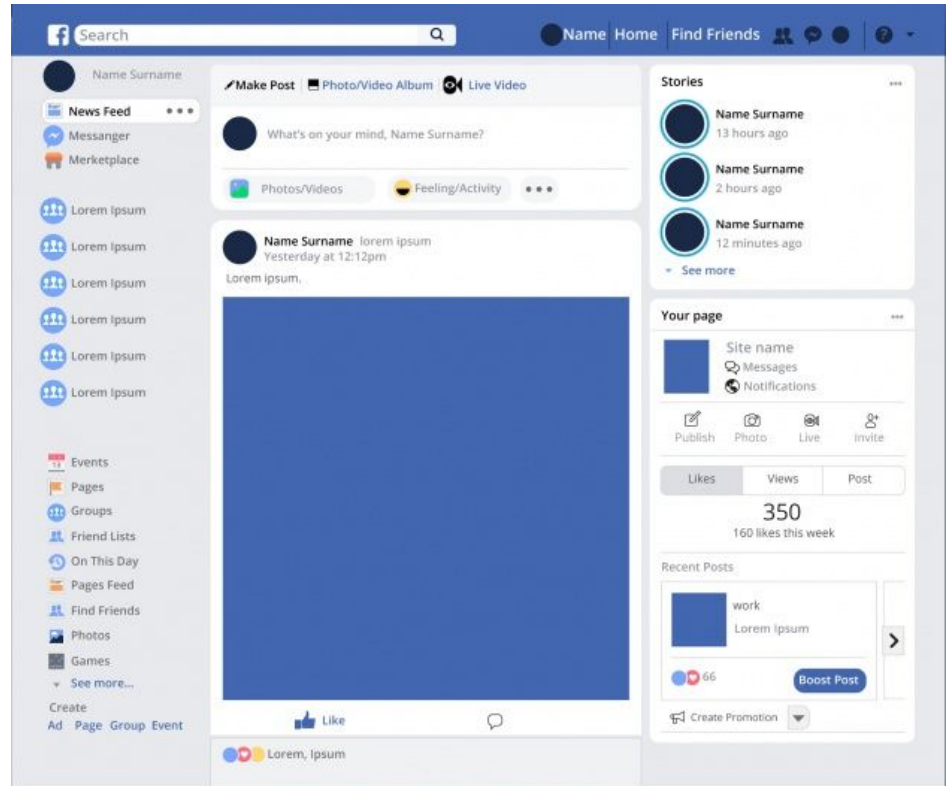
Single-Page-App (SPA)

Rendering only the minimal unit of the UI

e.g.

- only `Like` button
- only `Stories` tab

Offers better speed to users



React

A JavaScript library for building user interfaces

[Get Started](#)[Take the Tutorial >](#)

Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs.

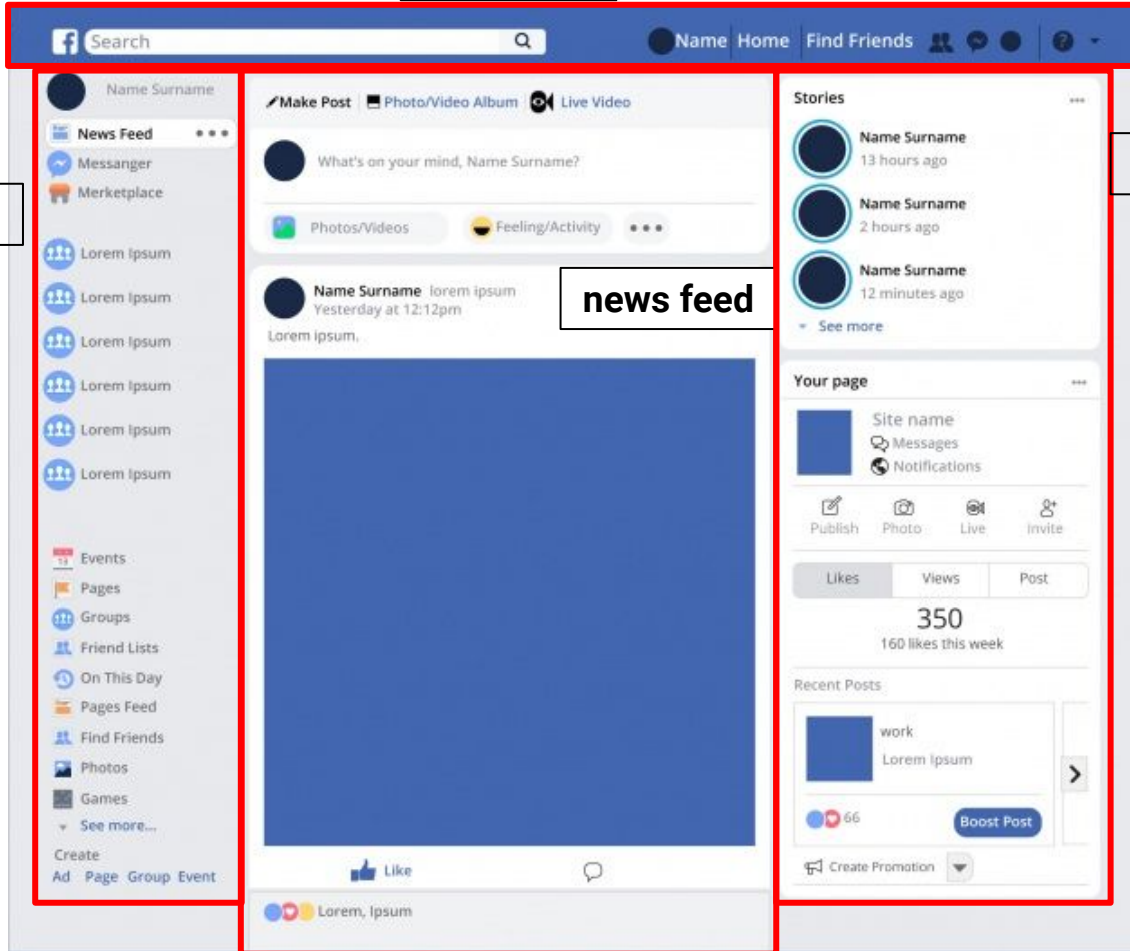
Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using [React Native](#).

header



sidebar

news feed

stories

Goal

InputBox

NewsFeed

PostBox

LikeButton

The image shows a web form and a news feed. The form at the top has a purple border. It contains a 'Username:' label, an input field with 'username123', a 'Post:' label, an input field with 'This is a sample post #4', and a 'Submit' button. Below the form is a news feed with a red border. It contains four post items, each with a blue border. Each post item includes a title (Post #0 to Post #3), the username 'username123', the post text, and a 'LIKE' button. The 'LIKE' button in the first post is highlighted with a green border.

Username:
username123

Post:
This is a sample post #4

Submit

Post #0
Username: username123
This is a sample post #1
LIKE

Post #1
Username: username123
This is a sample post #2
LIKE

Post #2
Username: username123
This is a sample post #3
LIKE

Post #3
Username: username123
This is a sample post #4
LIKE

State

State

- Private and controlled by the component
- Can be overwritten

e.g.

- Username
- Count of likes

Username:
username123

Post:
This is a sample post #4

Submit

Post #0
Username: username123
This is a sample post #1
LIKE

Post #1
Username: username123
This is a sample post #2
LIKE

Post #2
Username: username123
This is a sample post #3
LIKE

Post #3
Username: username123
This is a sample post #4
LIKE

Props

Props

- is simply “properties”
- Passing data from one component to another
- Uni-directional (parent → child)
- Read-only

Prerequisite

1. Basic knowledge of HTML, CSS, Javascript
 - a. If you went over tutorial #1 & 2, it should be okay
2. Node version ≥ 8.10 and npm version ≥ 5.6
 - a. <https://reactjs.org/docs/create-a-new-react-app.html>

class-based vs. functional component

```
import React from 'react';
import './index.css';

class InputBox extends React.Component {
  constructor(props) {
    super(props);
    this.onSubmit = this.onSubmit.bind(this);
  }

  onSubmit() {
    const { onSubmit } = this.props;
    const username = document.getElementById('username').value;
    const post = document.getElementById('post').value;
    console.log(username, post);

    onSubmit(username, post);
  }

  render() {
    return (
      <div id="input-box-container">
        <div id="input-box">
          <p>Username: <input id="username" /></p>
          <p>Post: <textarea id="post" /></p>
          <button onClick={this.onSubmit}>Submit</button>
        </div>
      </div>
    );
  }
}

export default InputBox;
```

```
function InputBox(props) {
  function onSubmit() {
    const { onSubmit } = props;
    const username = document.getElementById('username').value;
    const post = document.getElementById('post').value;
    console.log(username, post);

    onSubmit(username, post);
  }
  return (
    <div id="input-box-container">
      <div id="input-box">
        <p>Username: <input id="username" /></p>
        <p>Post: <textarea id="post" /></p>
        <button onClick={onSubmit}>Submit</button>
      </div>
    </div>
  );
}

export default InputBox;
```

React Plugin

<https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi?hl=en>