



# Web Programming Spring 2021

## #24

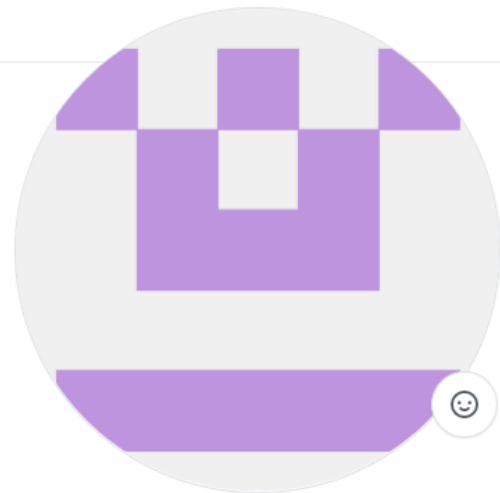
Chi-Jen Wu



# Homework #5

- npm run build
- git add ./build
- Without node\_modules
- Deadline 5/13 12:00 PM

- 整合一個 user profile 頁面
  - material-ui
  - 或是自己組也是可以
- 要有
  - id, name, 頭像 六個以上資訊
  - <https://api.github.com/users/cjwu>



**Chi-Jen Wu**

cjwu

Edit profile

🔍 3 followers · 0 following · ☆ 0

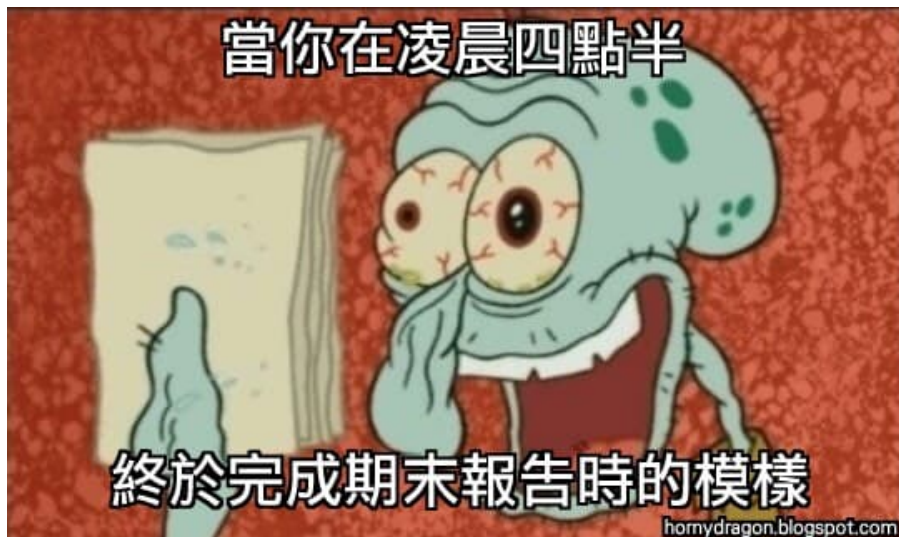
📍 Taiwan

✉ [cjwu@arbor.ee.ntu.edu.tw](mailto:cjwu@arbor.ee.ntu.edu.tw)

🔗 <https://cjwu.github.io>

# 期末報告分組 (40%)

- 2個人一組 或 乙個人一組 皆可
- 請用React 實作至少包含**五**個頁面的網站
  - 像是登入頁 個人頁 首頁 等等
  - 要有RWD
  - 要有Google Analytics
  - 要有 UI
- 後端必須要連動DB
  - API
- 部署到 heroku
  - 現在可以申請
  - 三個月免費使用 有VM, 有DB
- 下星期三前確定組員
- 最後兩堂課 請同學上來和報告 demo 成果
  - 每組10頁投影片介紹 每組大概十分鐘





# Inside a function

# 不要超過80行



# Function

“超過80行者”

分數 \* 80%

# 變數命名原則



```
let numberOfStudents
```

```
const numberOfLegs
```

```
function setBackgroundColor()
```

```
class Student{}
```

# 一張圖解釋何謂分組報告



# 分組報告

## 這件事



## 裝忙最會

想要耍廢但是還有一點良心，只好各種裝忙，假裝自己做了很多事，其實什麼事情也沒做

## 直接消失

只有分組時有看過他，其他時間完全看不到，群組也不回應，基本上可以不用算他一個



## 我是被逼的

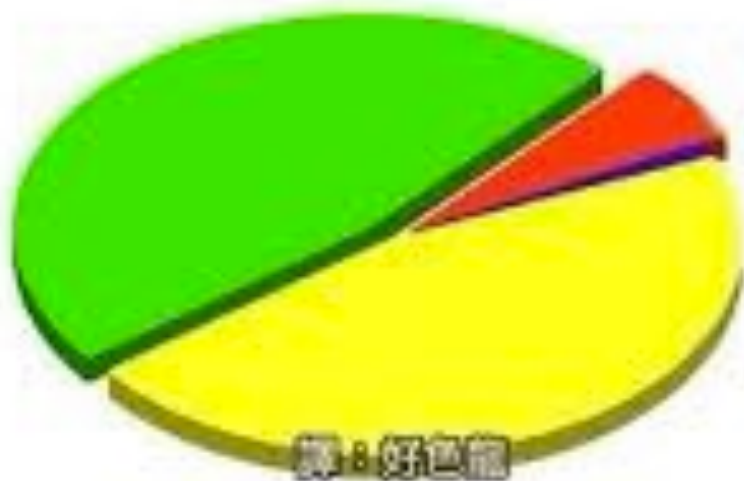
因為組員都是廢物，自己只能哭著硬幹，基本上80%的報告都會是他做的，為了過這門課只好委屈自己

## 根本廢物

只會嘴砲，不會做事，遇到事情就跑第一，只想靠別人罩，又愛催進度，是最討人厭的類型



# 分組報告讓我學到的事



- 知識
- 團隊合作
- 如何靠自己完成整份報告
- 人是一種機掰的生物

# 當你一個人就完成分組報告時



譯:高粱梗圖

# Topics

- The concepts of Web Services
- Web data protocols
  - HTTP, WebSocket, WebRTC
  - HTML, CSS
- Web JavaScript programming
- Cookies and sessions
- **Web Frontend frameworks**
- Web Backend frameworks
- RESTful API design





# React / ReactJS

- React Ecosystem
- 開發環境
- 核心概念 — 元件(Components)
- 單向資料流 (**Unidirectional data flow**)
- Virtual DOM
- Declarative UI / JSX
- React Router

# React / ReactJS 單向資料流

- 由上而下的「單向資料流」
  - Parent Component 透過 props 傳遞方法到 Child Component 去改變父元件的 state
- 在開發複雜的**大型**應用程式
  - 這會非常的難以管理
  - 所以有了**Flux** or **Redux** 等中央控管
    - 進階留給有興趣的同學
    - 建議等要寫大型應用程式時再了解



# React 最後一個不完整的範例 ToDoList

## CGU ToDoList

add a new todo...

Add

✓ React 最後一個範例 ToDoList



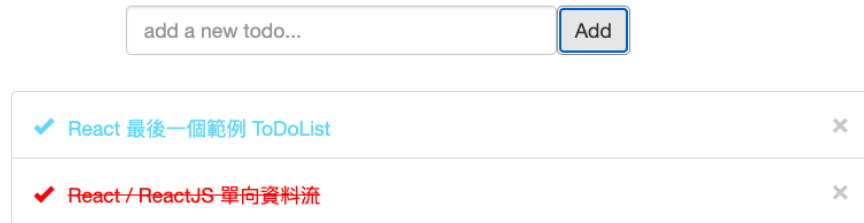
✓ React / ReactJS 單向資料流



# ToDoList components

- ToDoApp
  - ToDoHeader
  - ToDoAddForm
  - ToDoList
    - ToDoListItem #1
    - ToDoListItem #2
    - ToDoListItem #3

## CGU ToDoList

A screenshot of the CGU ToDoList application interface. It features a text input field with the placeholder text "add a new todo..." and an "Add" button. Below the input field is a list of two items, each with a checkmark icon and a close button (X). The first item is "React 最後一個範例 ToDoList" with a blue checkmark. The second item is "React / ReactJS 單向資料流" with a red checkmark.

✓ React 最後一個範例 ToDoList	×
✓ React / ReactJS 單向資料流	×

# ToDoApp components

- ToDoTitle
- ToDoAddForm
- ToDoList

## CGU TodoList

Add

✓ React 最後一個範例 ToDoList

✕

✓ React / ReactJS 單向資料流

✕







# ToDoTitle components

```
class ToDoTitle extends React.Component {  
  render () {  
    return <h1>CGU Todo list</h1>;  
  }  
}
```





# ToDoAddForm components

```
class ToDoAddForm extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {value: ''};  
    this.handleChange = this.handleChange.bind(this);  
    this.handleSubmit = this.handleSubmit.bind(this);  
  }  
  handleChange(event) {  
    this.setState({value: event.target.value});  
  }  
  handleSubmit(event) {  
    this.setState({value: event.target.value});  
    event.preventDefault();  
  }  
  render() {  
    return (  
      <form onSubmit={this.handleSubmit}>  
        <input type="text" value={this.state.value} onChange={this.handleChange} />  
        <input type="submit" value="Submit" />  
      </form>  
    );  
  }  
}
```





# Text Input review



```
import React from 'react';  
import ReactDOM from 'react-dom';
```

```
class MyForm extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = { username: '' };  
  }  
  myChangeHandler = (event) => {  
    this.setState({username: event.target.value});  
  }  
  render() {  
    return (  
      <form>  
        <h1>Hello {this.state.username}</h1>  
        <p>Enter your name:</p>  
        <input type='text' onChange={this.myChangeHandler} />  
      </form>  
    );  
  }  
}
```

```
ReactDOM.render(<MyForm />, document.getElementById('root'));
```

```
import React from 'react';
import ReactDOM from 'react-dom';

class MyForm extends React.Component {
  constructor(props) {
    super(props);
    this.state = { username: '' };
  }
  myChangeHandler = (event) => {
    this.setState({username: event.target.value});
  }
  render() {
    return (
      <form>
      <h1>Hello {this.state.username}</h1>
      <p>Enter your name:</p>
      <input
        type='text'
        onChange={this.myChangeHandler}
      />
      </form>
    );
  }
}

ReactDOM.render(<MyForm />, document.getElementById('root'));
```

# Hello CGUer

Enter your name:







# Homework #6

- 把ToDoList 範例完成
  - 新增
  - 刪除
  - 做完
- npm run build
  - git add ./build
  - Without node\_modules
- 然後 push github

## CGU Todo list

Add

✓ React / ReactJS 單向資料流	×
✓ React 最後一個範例 ToDoList	×
✓ sxsqwq	×
✓ cd	×



# Thanks!

## Open for any questions

**CJ Wu**

[cjwu@mail.cgu.edu.tw](mailto:cjwu@mail.cgu.edu.tw)