

# React Review & Cookie

Software Studio

DataLab, CS, NTHU

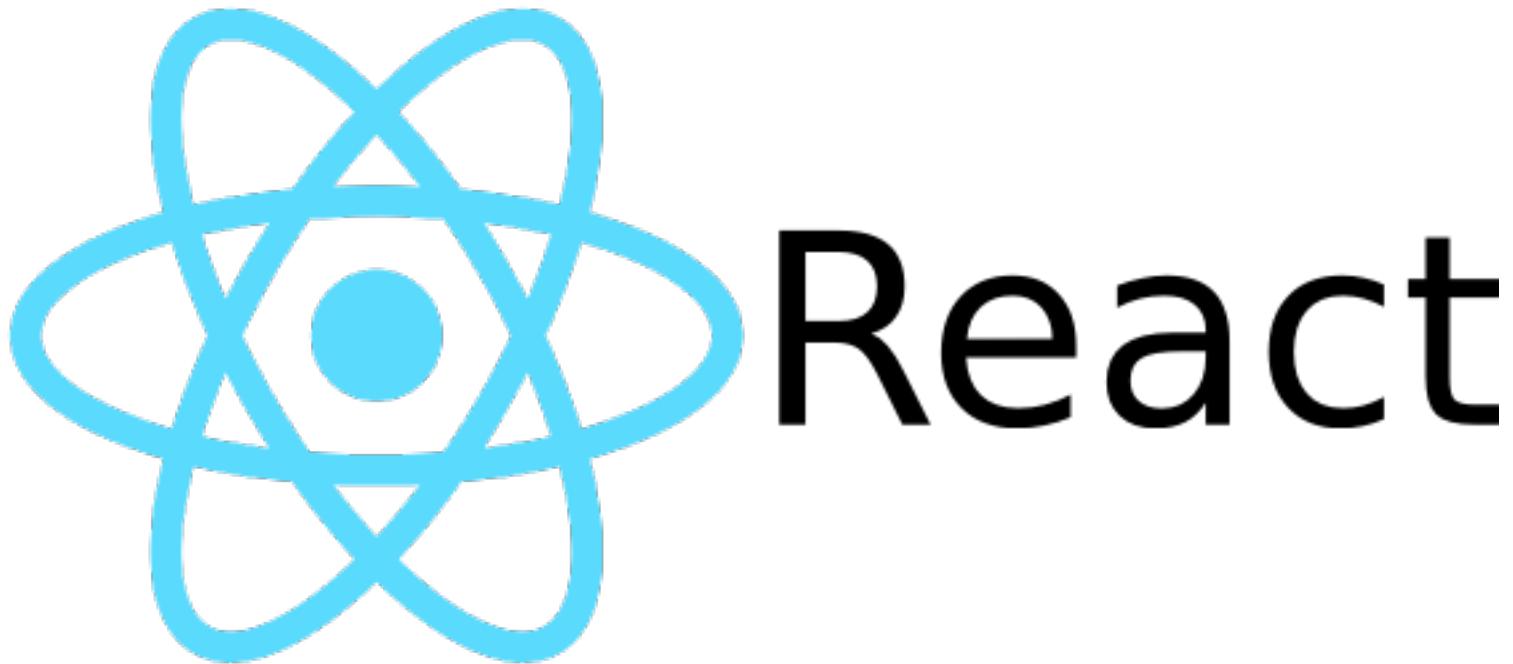
2017 spring

# Outline

- React
- Http request
- Cookie
- Today's Mission

# Outline

- React
- Http request
- Cookie
- Today's Mission



# React

- Write HTML in JS

```
render() {  
  return (  
    <div className={`today weather-bg ${this.state.group}`}>  
      <div className={`mask ${this.state.masking ? 'masking' : ''}`}>  
        <WeatherDisplay {...this.state}/>  
        <WeatherForm city={this.state.city} unit={this.props.unit} onQuery={this.handleFormQuery}/>  
      </div>  
    </div>  
  );  
}
```

- Component based

# OOP

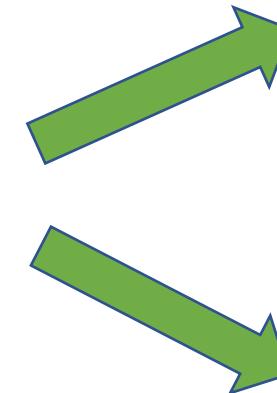


Class



Object

Object.method()



Object.property



# Component based in React

- Use Class

```
export default class WeatherDisplay extends React.Component {  
  constructor(props) {  
    super(props);  
  }  
}
```

- New Object

```
  render() {  
    return (  
      <div className={`today weather-bg ${this.state.group}`}>  
        <div className={`mask ${this.state.masking ? 'masking' : ''}`}>  
          <WeatherDisplay {...this.state}/>  
        </div>  
      <WeatherForm city={this.state.city} unit={this.props.unit} onQuery={this.handleFormQuery}/>  
    </div>  
  )  
}
```



Think about

How to communicate between components ?

by davejdoe



by David Blackwell

# Props

- Read only
- Parent to Child's property

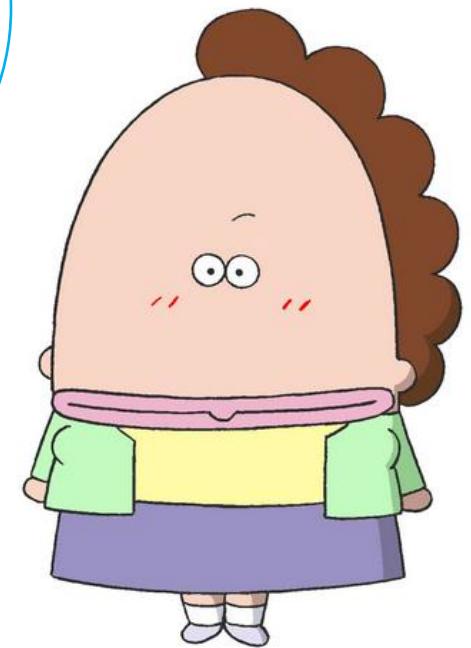
```
ReactDOM.render(  
  <div>  
    <Component name='Alice'>  
      <Component name='Bob'>  
    </div>,  
    document.getElementById('root')  
);
```

```
class Component extends React.Component {  
  constructor(props) {  
    super(props); ...  
  }  
  render() {  
    return <h2>This is {this.props.name}</h2>;  
  }  
}
```

Now, you can let child talk to parent!

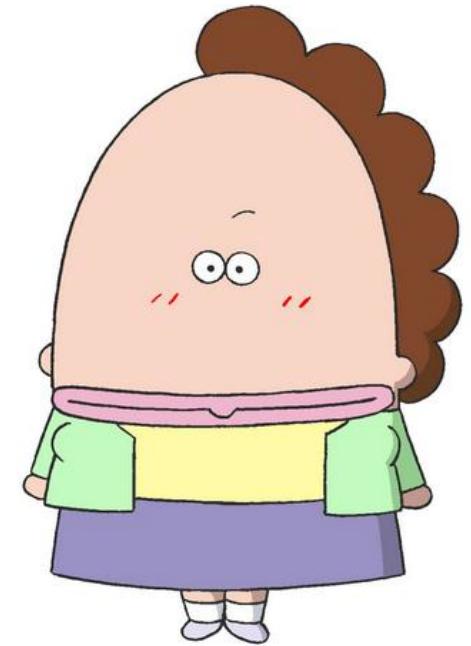
props

Parent.method





props.method





props.method

**Do something!**



How about parent talk to child?

# State

- Can be modified

```
constructor(props) {  
  super(props);  
  
  this.state = {  
    ...Today.getInitWeatherState(),  
    loading: true,  
    masking: true  
  };  
}
```

# State

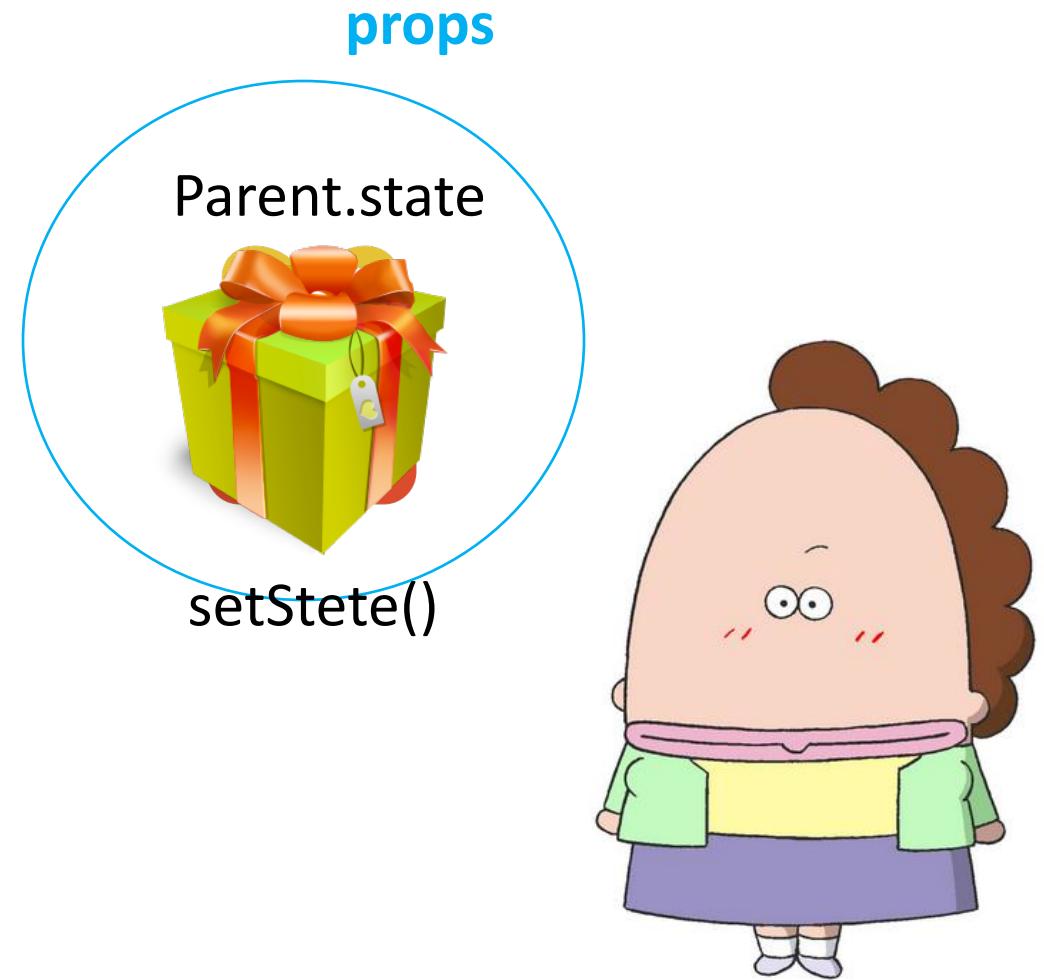
- Can be modified
- If state is used in render(), when you call setState(), it will trigger re-render automatically.

```
render() {  
  return (  
    <div className={`today weather-bg ${this.state.group}`}>  
      <div className={`${this.state.masking ? 'masking' : ''}`}>  
        <WeatherDisplay {...this.state}/>  
        <WeatherForm city={this.state.city} unit={this.props.unit} onQuery={this.handleFormQuery}/>  
      </div>  
    </div>  
  );  
}
```

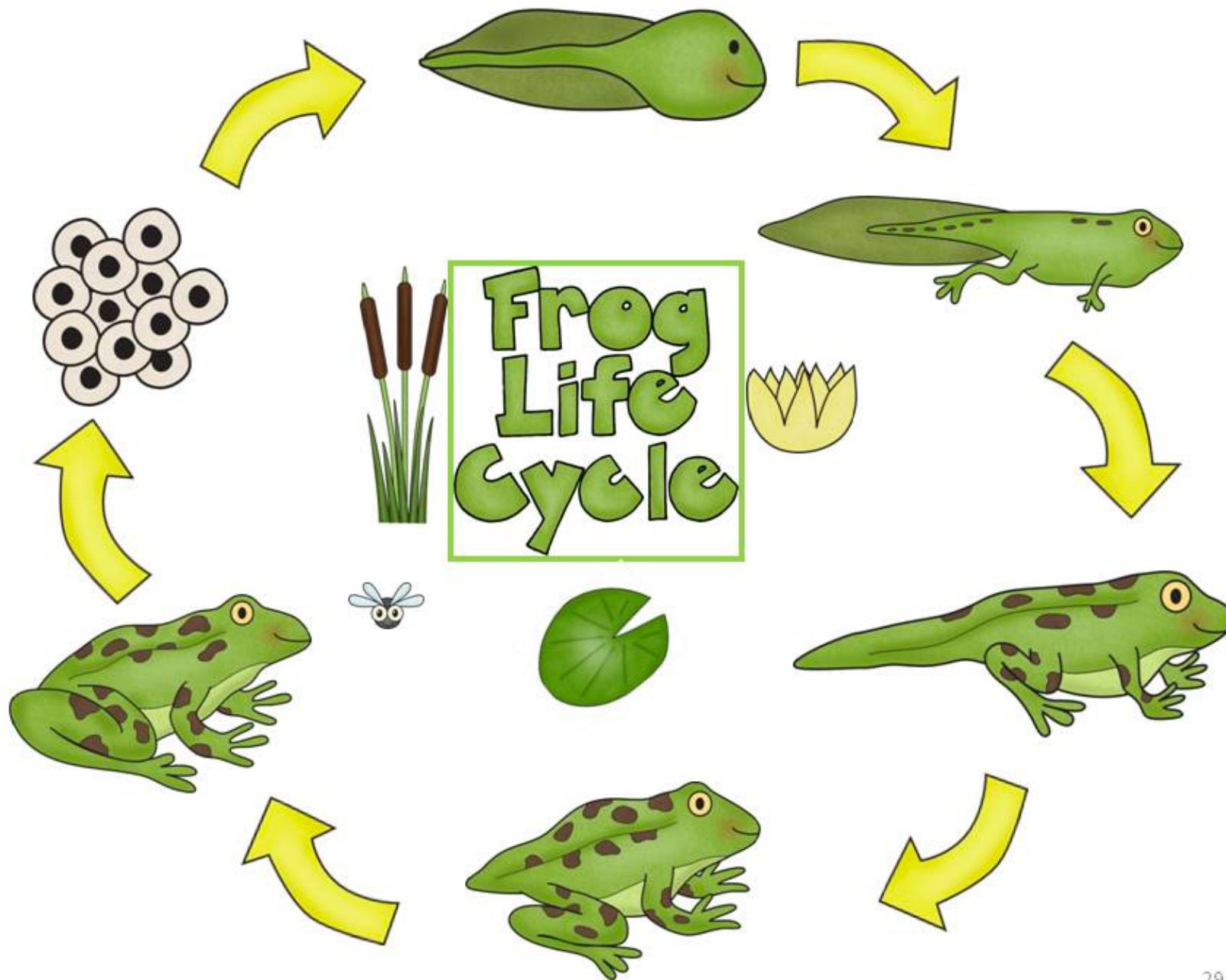
```
getWeather(city, unit) {  
  this.setState({  
    loading: true,  
    masking: true,  
    city: city // set city state immediately to prevent input  
  }, () => { // called back after setState completes  
    getWeather(city, unit).then(weather => {  
      this.setState({  
        ...weather,  
        loading: false  
      }, () => this.notifyUnitChange(unit));  
    }).catch(err => {  
      console.error('Error getting weather', err);  
  
      this.setState({  
        ...Today.getInitWeatherState(unit),  
        loading: false  
      }, () => this.notifyUnitChange(unit));  
    });  
});  
};
```



**Trigger re-render!!**



If I want to let child do some method when re-render?



# Component Life Cycle

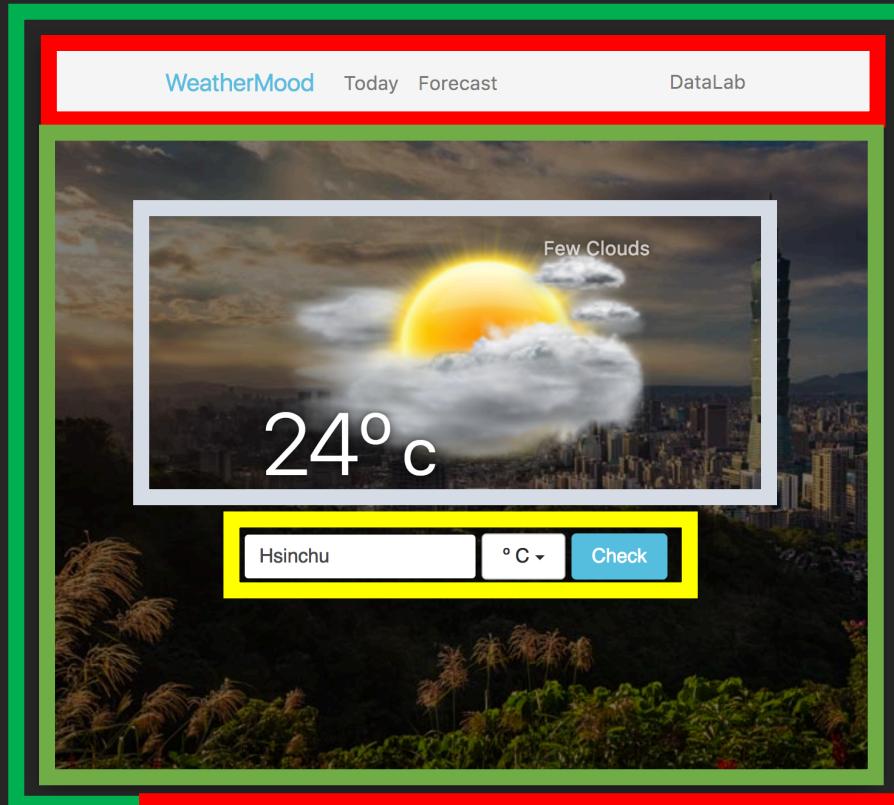
- Life Cycle

```
componentWillReceiveProps (nextProps) {  
    // do some method when re-render  
    // you can use use nextProps  
}
```

# Case study

# Step 5: Upward Data Flow

Main {  
  unit  
}



Navbar { link1, link2 }

WeatherDisplay {  
  temp, unit  
  weather, desc  
}

WeatherForm {  
  city, unit  
}

Today { weather, temp, desc, city }  
Forecast { weather, temp, desc, city }

# Case study

- If I want to set city as the same in Today & Forecast ?
- Any Idea?



# Outline

- React
- HTTP Method
- Cookie
- Today's Mission

# HTTP Method

- OPTIONS
- GET
- HEAD
- **POST**
- **PUT**
- DELETE
- TRACE
- CONNECT

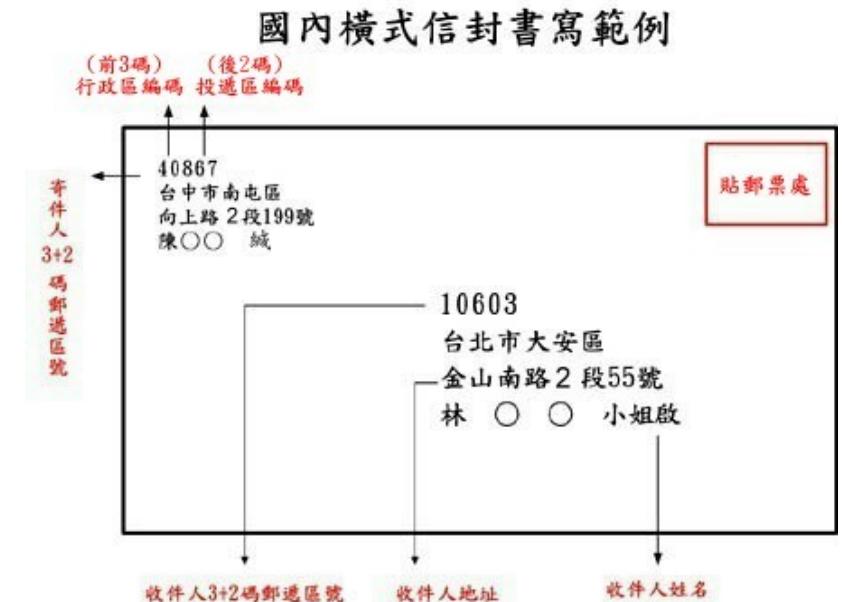
# GET

- Append Query String (key/value pair) behind URL.
- <http://api.openweathermap.org/data/2.5/weather?appid=Myid&q=London>



# POST

- Put key/value pair in message body.



# Outline

- React
- Http request
- **Cookie**
- Today's Mission

# What is cookie?



# How do we keep user's info?

```
public function signin() {
    if($_POST && $POST['username'] && $POST['password']) {
        $username = $_POST['username'];
        $password = $_POST['password'];

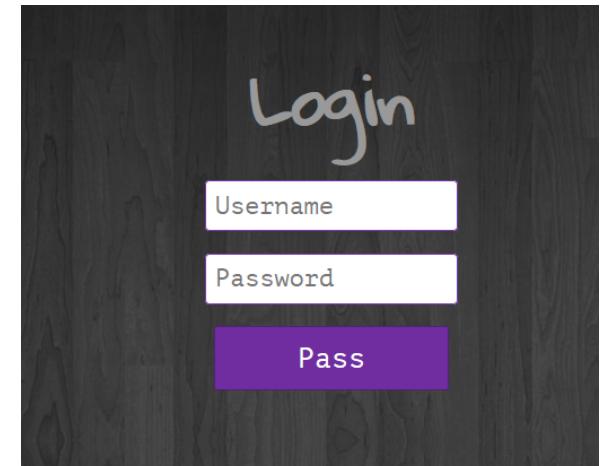
        $query = "SELECT * FROM users WHERE username = '$username' AND password = '$password'";
        $res = $this->db->query($query);
        if($res->num_rows == 1) {
            $user = $res->fetch_assoc();
            session_start();
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['username'] = $user['username'];
            $_SESSION['password'] = $user['password'];
            $_SESSION['email'] = $user['email'];
            header('Location: index.php');
        } else {
            echo "User not found or password is incorrect";
        }
    }
}
```



You can use Cookie!

# Cookie

- **Small** piece of data stored locally.
- String only.
- Expire when turn off the browser.
- name=value;expires=date;path=path;domain=domain;secure



# Some constraint

- Too small (only 4KB)
- Pass as http request header
- Not secure

# Using cookie via js

```
<script>
  if(navigator.cookieEnabled){
    console.log("網頁的 cookie :" + document.cookie);
  }
  else {
    console.log("不支援Cookie");
  }
</script>
```

```
document.cookie = "123=QQQ";
document.cookie = "123=0u0";
document.cookie = "abc=yaya";

document.write("Cookie 字串: " + document.cookie);
```

123=0u0 ; abc=yaya

# Using cookie via ReactJs

- [react-cookie](#)
- Step 1: npm install xxx
- Step 2: import
- Step 3: use their api call
  - cookie.save(name, val)
  - cookie.load(name)
  - cookie.remove(name)

# Outline

- React
- Http request
- Cookie
- Today's Mission

# Today's Mission

- Add a button to store favorite city in cookie.
- Show weather info when the navbar has clicked.
- Deadline 4/12 23:59

# Some reminder

- Assignment