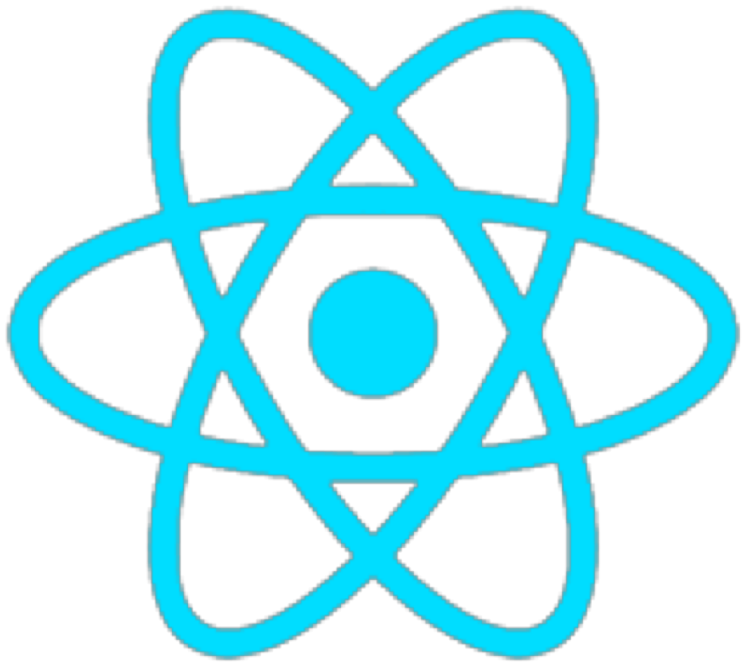
React-GoogleMap

Web Dev

DataLab, CS, NTHU

2019 Spring



React

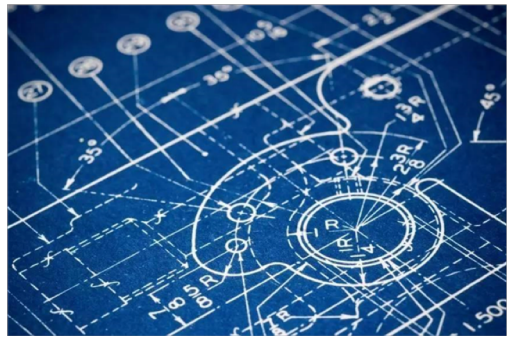
React

- Write HTML in JS

```
render() {
    return (
        <div className={`weather-display ${this.props.masking
            ? 'masking'
            : ''}`}>
            <img src={`images/w-${this.props.group}.png`}/>
            <p className='description'>{`${this.props.day}: ${this.props.description}`}</p>
            <h1 className='temp'>
                <span className='display-3'>{this.props.temp.toFixed(0)}&ordm;</span>
                 {(this.props.unit === 'metric')
                    ? 'C'
                    : 'F'}
            </h1>
        </div>
    );
}
```

- Component based

OOP(Object Oriented Programming)



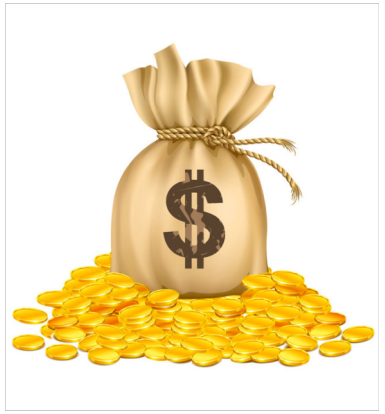
Class

new

Object

Object.method()

Object.property



Component based in React

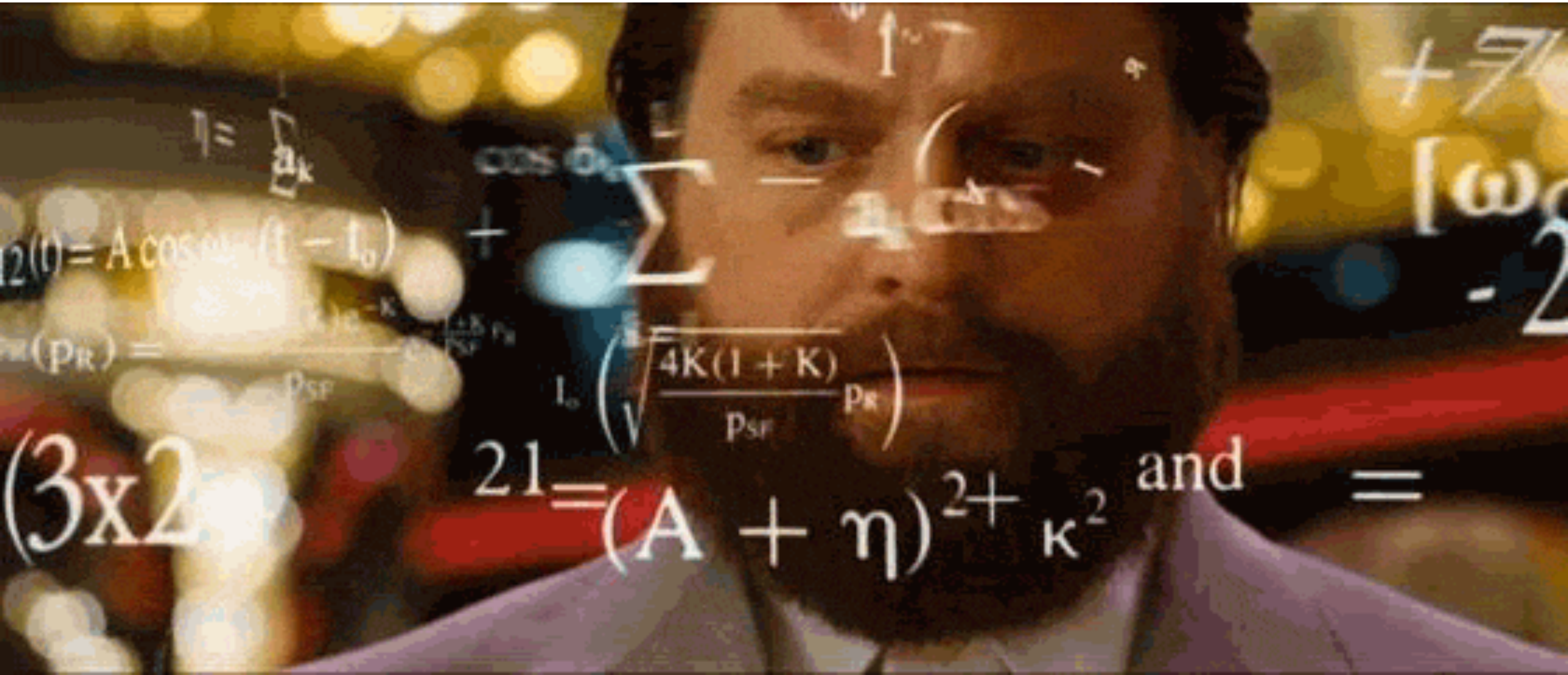
- Use class

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

    }
```

- New object

```
render() {

    document.body.className = `weather-bg ${tomorrow.group}`;
    document.querySelector('.weather-bg .mask').className = `mask ${masking ? 'masking' : ''}`;

    return (
        <div className='forecast'>
            <div className='tomorrow'>
                <WeatherDisplay {...tomorrow} day='tomorrow' unit={unit} masking={masking}/>
            </div>
```

How to communicate between components?



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

How about parent talk to child

State

- Can be modified

```
constructor(props) {
    super(props);

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

State

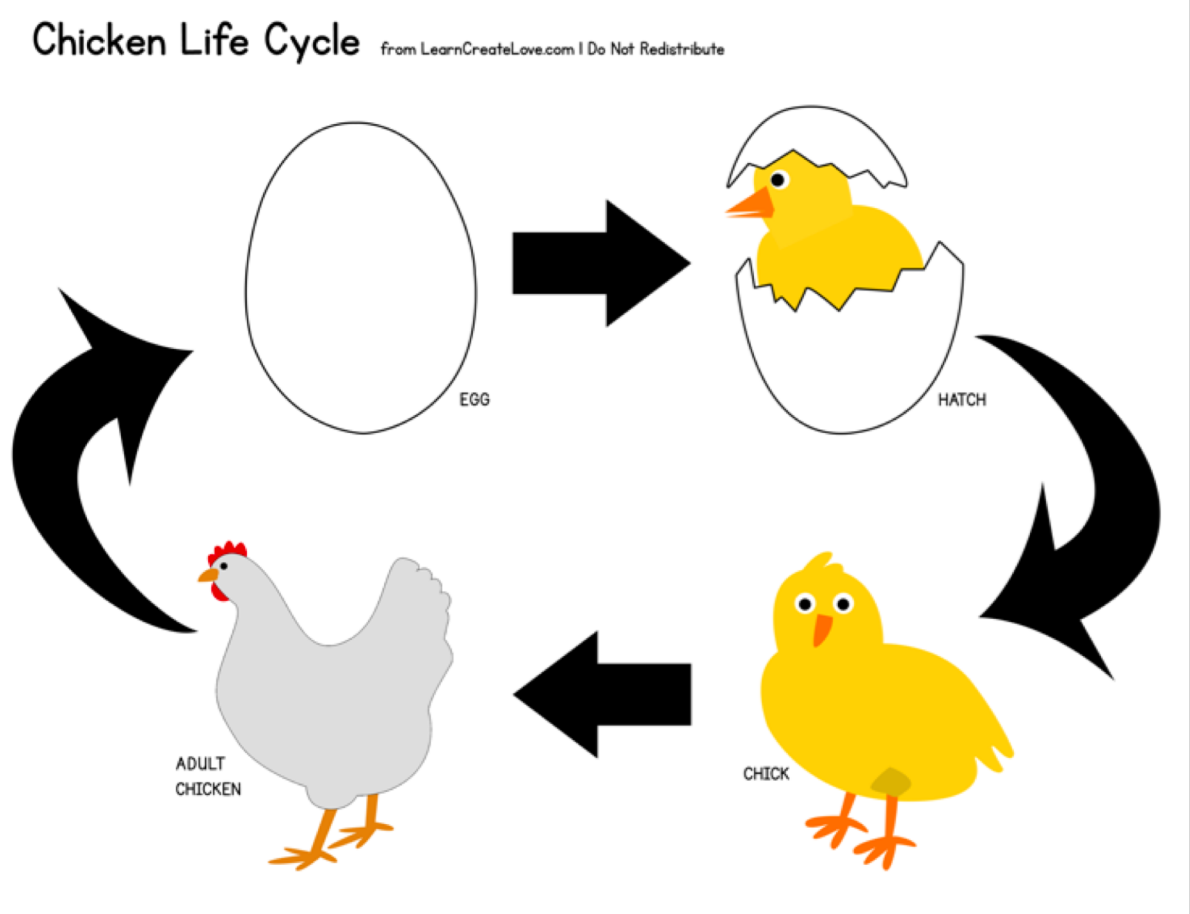
- Can be modified
- If state is used in render(), when you call setSate(), it will trigger re-render automatically.

```
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>
    );
}
```

```
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));
        });
    });
```

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

Component Life Cycle

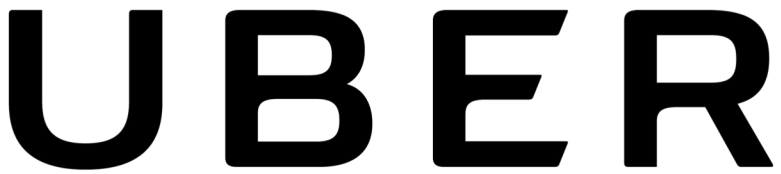


Component Life Cycle

- Life Cycle

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

Case Study-GoogleMap



GoogleMap API

React-Googlemap

- google-maps-react
- npm install --save google-maps-react

React-GoogleMap

- Add API KEY

```
import {GoogleApiWrapper} from 'google-maps-react';

// ...

export class MapContainer extends React.Component {}

export default GoogleApiWrapper({
  apiKey: (YOUR_GOOGLE_API_KEY_GOES_HERE)
})(MapContainer)
```

Add Google Map

- In index.html

```
<body>
  <div id="map"></div>
  <script src="vendor.bundle.js"></script>
  <script src="index.bundle.js"></script>
</body>
```

- In index.jsx

```
var map = document.querySelector("#map")

ReactDOM.render(
  <div>
    <MapContainer/>
  </div>,
  map
);
```

- In map.jsx

```
render() {
  const style = {
    width: '100%',
    height: '100%',
    'marginLeft': 'auto',
    'marginRight': 'auto'
  }
  return (
    <Map
      item
      style = { style }
      google = { this.props.google }
      onClick = { this.onMapClick }
      zoom = { 13 }
      initialCenter = {this.state.position}
    >
```

Define state to interact with map

- State:
  - showingInfoWindow:boolean
  - activeMarker:{}
  - Position:{lat, lng}
  - Name:{city:,address:}

```
this.state = {
  showingInfoWindow: false,
  activeMarker: {},
  position:{lat: 24.795894, lng: 120.993528},
  name:{city:'city', address:'address'}
}
```

Add Marker and InfoWindow

```
<Marker
  onClick = { this.onMarkerClick }
  position = { this.state.position}
/>
<InfoWindow
  marker = { this.state.activeMarker }
  visible = { this.state.showingInfoWindow }
>
<div>
  <h2>{this.state.name.city}</h2>
  <h3>{this.state.name.address}</h3>
</div>
</InfoWindow>
```

Add onClick event to map

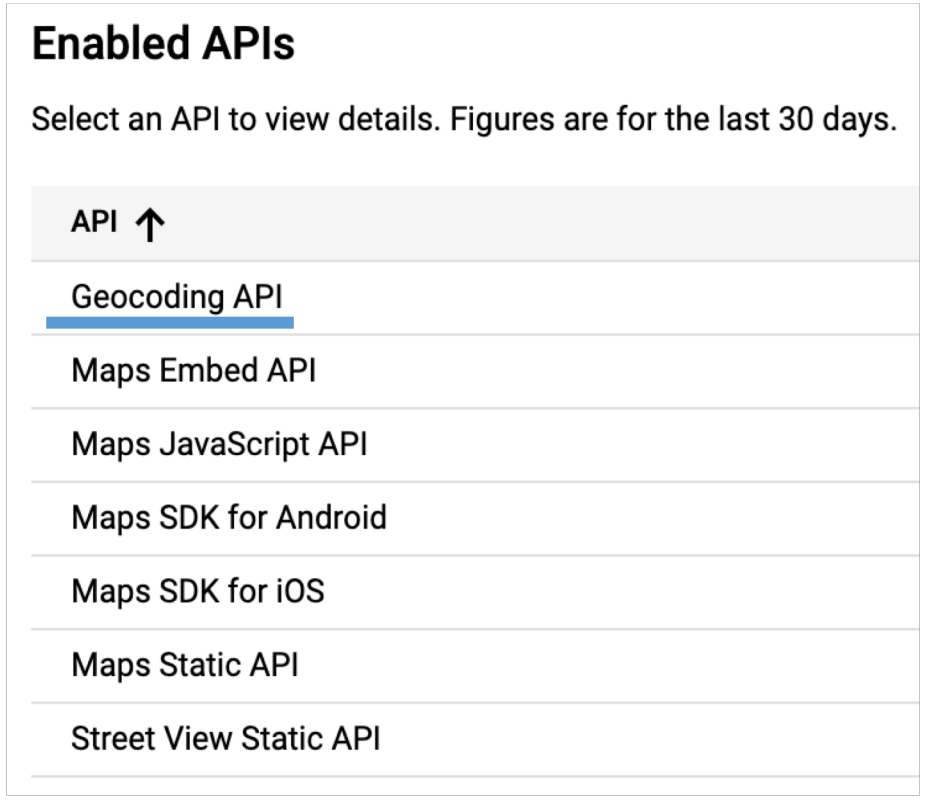
```
<Map
  item
  style = { style }
  google = { this.props.google }
  onClick = { this.onMapClick }
  zoom = { 13 }
  initialCenter = {this.state.position}
>
```

```
constructor(props) {
  super(props);
  this.state = {
    showingInfoWindow: false,
    activeMarker: {},
    position:{lat: 24.795894, lng: 120.993528},
    name:{city:'city', address:'address'}
  }
  // binding this to event-handler functions
  this.onMarkerClick = this.onMarkerClick.bind(this);
  this.onMapClick = this.onMapClick.bind(this);
}
```

```
onMapClick = (props, map, coord) => {
  const { latLng } = coord;
  const lat = latLng.lat();
  const lng = latLng.lng();

  Geocode.enableDebug();
  // Get address from latidude & longitude.
  Geocode.fromLatLng(lat, lng).then(
    response => {
      const address = response.results[0].formatted_address;
      const city = response.results[6].formatted_address;
      this.setState({
        position:{lat, lng},
        showingInfoWindow: false,
        activeMarker: null,
        name: {city, address}
      });
    },
    error => {
      console.error(error);
    }
  );

}
```

Geocode

- [React-geocode]
- npm install --save react-geocode



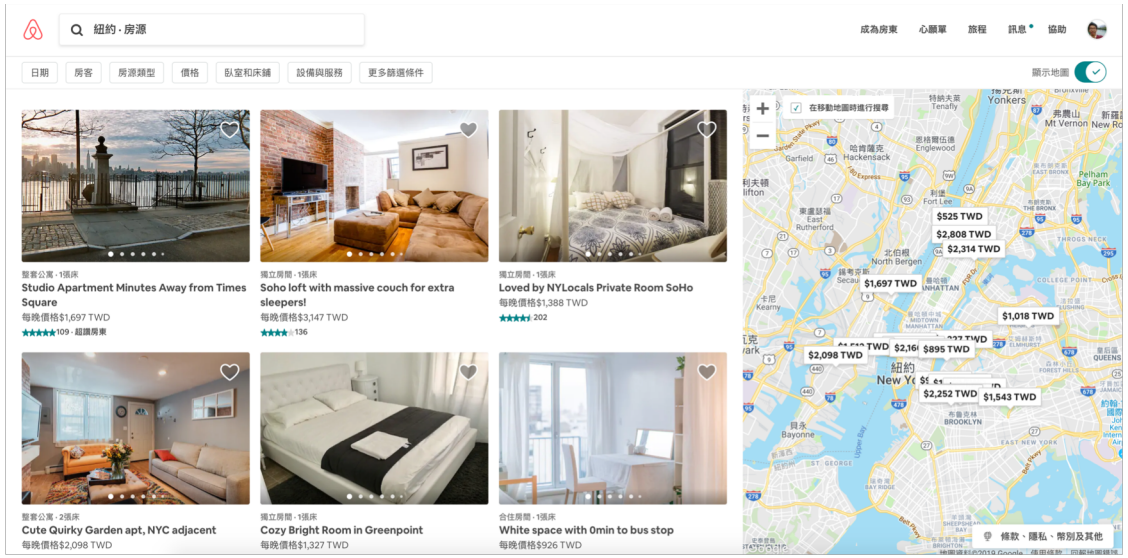

```
Geocode.fromLatLng(lat, lng).then(
  response => {
    const address = response.results[0].formatted_address;
    const city = response.results[6].formatted_address;
    this.setState({
      position:{lat, lng},
      showingInfoWindow: false,
      activeMarker: null,
      name: {city, address}
    });
  },
  error => {
    console.error(error);
  }
);
```

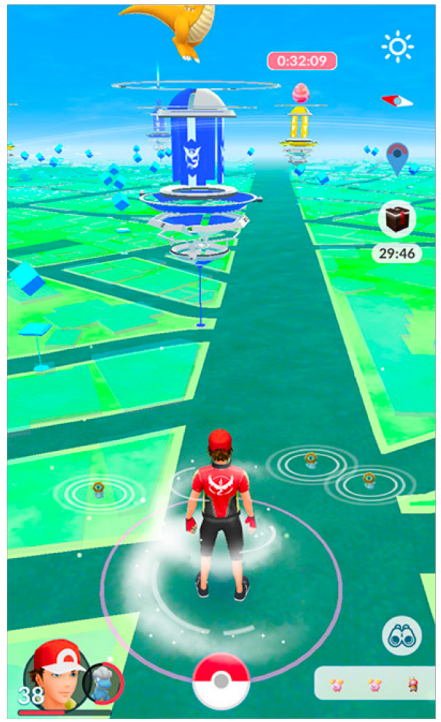
Add onClick event to marker

```
<Marker
  onClick = { this.onMarkerClick }
  position = { this.state.position}
/>
```

```
this.onMarkerClick = this.onMarkerClick.bind(this);
```

```
onMarkerClick = (props, marker, e) => {
  this.setState({
    activeMarker: marker,
    showingInfoWindow: true
  });
}
```


紐約 · 房源
成為房東
心願單
旅程
訊息
協助
日期
房客
房源類型
價格
臥室和床舖
設備與服務
更多篩選條件
顯示地圖
整套公寓 · 1張床
Studio Apartment Minutes Away from Times Square
每晚價格$1,697 TWD
109 · 超讚房東
獨立房間 · 1張床
Soho loft with massive couch for extra sleepers!
每晚價格$3,147 TWD
136
獨立房間 · 1張床
Loved by NYLocals Private Room SoHo
每晚價格$1,388 TWD
202
整套公寓 · 2張床
Cute Quirky Garden apt, NYC adjacent
每晚價格$2,098 TWD
獨立房間 · 1張床
Cozy Bright Room in Greenpoint
每晚價格$1,327 TWD
合住房間 · 1張床
White space with 0min to bus stop
每晚價格$926 TWD
$525 TWD
$2,808 TWD
$2,314 TWD
$1,697 TWD
$1,018 TWD
$2,098 TWD
$895 TWD
$2,252 TWD
$1,543 TWD
條款、隱私、幣別及其他


0:32:09
29:46
38


Uber
駕駛
搭乘
企業版
更多項目
登入
註冊
新竹縣, Taiwan
清華大學郵局ATM, Hsinchu City, Ea...
很抱歉，我們無法預估該趟行程的費用。
新竹市
竹北車站
新竹火車站
千甲火車站
國立清華大學
https://www.google.com/intl/zh-TW_US/help/terms_maps.html