

# Frontend Framework

*React*

# What is React?

- UI Library
- Component-based
  - Reusable  
function ដែលត្រូវបានប្រើប្រាស់ឡើងឡើង (API ការប្រើប្រាស់នេះមានចំណាំពីរបាន)
- Declarative API    vs Imperative
  - <https://codeburst.io/declarative-vs-imperative-programming-a8a7c93d9ad2>

# React components

ក្រុមការងារ 4 តាម

- JSX Syntax ក្នុង
- Element & component
- State & Props កែវាត់សំណើនាយករាជការនៃ component
- Virtual DOM ផលិតផលបញ្ហាផី

# Setting up Tools and Environment

- Node.JS
  - <https://nodejs.org/en/>
- Yarn
  - Package manager
    - Installing Node.js' dependencies
  - For Node.js version higher than 16.10
    - Enable [corepack] via [corepack enable](#) (ກະລຸນາຕົວຢ່າງ Admin)
    - Run the above command as an Admin
- Visual Studio Code
  - Install plugins:
    - ESLint
    - Prettier-Code formatter
      - Setting-> search format -> editor: Format On Save
    - Vscode-styled-components
- Chrome Extension [React Developer Tools](#) (ຫຼັງຈາກມີ Debug ມານີ້)
  - React Developer Tools:  
<https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienih?hl=en>

# Create React project

- Change to target folder
- Run command
  - `yarn create react-app project_name`
- After complete
  - VS menu open folder -> `project_name`
  - From menu Terminal -> new
    - `yarn start`

# Concept

- UI is the function of state

$$UI = f(state)$$

State component ດັນຈະສາມາ  
ກົດຕິວິທະຍາງ vendor ທີ່

# JavaScript XML (JSX)

- Using XML in JavaScript
- Use {} to evaluate the expression

# Element

- In react, html element can be defined using const as shown:

```
const h = <div><h1>SW Studio News03</h1></div>;
```

# Header

```
import './App.css';
import { Component } from 'react';
```

css

Component object

- `import React` Import React from 'react';

## Component creation (ES6 and earlier) *Type class*

- Import React from 'react';

```
var NewsHead04 = React.createClass({
  render: function() {
    return(
      <div>
        <h1>SW Studio News</h1>
      </div>
    );
  }
});
```

# Component

## Type Component

Given import

- import React from 'react';

```
class NewsHead01 extends React.Component {  
    render() {  
        return (  
            <div>  
                <h1>SW Studio News</h1>  
            </div>  
        );  
    }  
}
```

# Component creation (ES6)

- import { Component } from 'react';
- To create Component
  - Class's name must begin with capital letter

Example:

class component class

```
class NewsHead01 extends Component {  
  render() {  
    return (  
      <div>  
        <h1>SW Studio News</h1>  
      </div>  
    );  
  }  
}
```

Class component

function component

```
function NewsHead02() {  
  return (  
    <div>  
      <h1>SW Studio News</h1>  
    </div>  
  );  
}
```

function component

import { Component } from 'react';

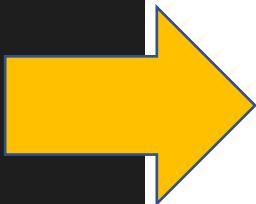
- type component
- create component which type

# Component

new static (constructor)

- To create element with class

```
class NewsHead01 extends React.Component {  
  render() {  
    return (  
      <div className='class01'>  
        <h1>SW Studio News</h1>  
      </div>  
    );  
  }  
}
```



```
<div class='class01'>  
  <h1>SW Studio News</h1>  
</div>
```

# Using Component

မျှေး component

```
class Link01 extends Component {  
    render() {  
        return(  
            <a href="https://sv1.picz.in.th/images/2022/02/06/nM1SyZ.jpg" >Click here</a>  
        );  
    }  
}  
  
class Block01 extends Component {  
    render() {  
        return(  
            <div>  
                <Link01 /> ←  
            </div>  
        );  
    }  
}
```

နိုင်လေ့လာမှုဆုံးမှတ်တမ်းမှုတွင် tag

- Ex click แล้ว alert ขึ้น

## Using JavaScript with element

```
class Link03 extends Component {  
  display(ev) {  
    alert(ev.target.href);  
  }  
  render() {  
    return(  
      <a href="https://sv1.picz.in.th/images/2022/02/06/nM1SyZ.jpg"  
        onMouseOver={this.display}>Click here</a>  
    );  
  }  
}
```

The diagram illustrates the flow of events. A blue arrow points from the `onMouseOver` attribute in the JSX code to the `display` method definition. Another blue arrow points from the `ev` parameter in the `display` method back to the `onMouseOver` attribute, indicating that the event object is passed from the event handler back to the component's state or props.

Method

parameter ev = link

Ex. នឹងដំឡើង bg លើក + color អ៉ាវិក

## Event handling

ជីវិតុក នៃ component

បញ្ជាក់ function នៃការការពារ

```
const changecolor = event => {
    const e = event.target;
    e.style.color = 'blue';
    e.style.background = 'yellow';
}

class Link01 extends Component {
    render() {
        return(
            <a href="https://sv1.picz.in.th/images/2022/02/06/nM1SyZ.jpg"  

                onMouseOver={(event) => changecolor(event)}>Click here</a>
            );
    }
}
```

## Event handling (cont.)

ผู้ใช้ component ที่มีผู้ใช้ function

```
class Link02 extends Component {  
    display(ev) {  
        ev.target.style.color='blue';  
        ev.target.style.background = 'yellow';  
    }  
    render() {  
        return(  
            <a href="https://sv1.picz.in.th/images/2022/02/06/nM1SyZ.jpg"  
                onMouseOver={this.display}>Click here</a>  
        );  
    }  
}
```

# Props

ព័ត៌មាន

- Props = Properties  
*នឹង object ដែលត្រូវ value*
- Props is an object that stores the value  
*ក្នុងការសងការ នឹង parent component ទៅ child component*
- Props can be passed from parent component to child component
  - Uni-directional (parent to child only)
  - Props' data cannot be modified by the child component (read only)  
*child អនុវត្តន៍យកគោរព props មិនអាច  
ការកំណត់ត្រាវិញ្ញាបន្ត*

# Props example

Prop គឺជាអ្នកផ្តល់ព័ត៌មាន អូតិស៊ីនុយោង ឬតាមឯកសារ

```
① class ShowTitle extends Component {  
    render() {  
        return (  
            <h1>Headline {this.props.title}</h1>  
        );  
    }  
}  
class Block01 extends Component {  
    render() {  
        return(  
            <div>  
                ①.1 <ShowTitle title="Headline 01" />  
                ①.2 <ShowTitle title="Headline 02" />  
                ①.3 <ShowTitle title="Headline 03" />  
            </div>  
        );  
    }  
}
```

The diagram illustrates the flow of props from the parent component to the child component. A yellow arrow points upwards from the 'title' prop in the 'Block01' component's render method to the 'ShowTitle' component in the 'Block01' component's render method. Another yellow arrow points downwards from the 'ShowTitle' component in the 'Block01' component's render method to the 'title' prop in the 'ShowTitle' component's render method. Handwritten annotations include '1.1.3' in a circle at the top right, '1.1.2' in a circle next to the first 'ShowTitle' component, and '1.2.3' and '1.3.3' in circles next to the second and third 'ShowTitle' components respectively.

**State** state ແນວຍກາ = state ສິ່ງໃນ re-render ອີເມວດກຳໄປຂອງ

Data ດູກເກີບການໃຊ້ component

- Data storage within component  
ສໍາງ ໂດຍ ຄັດມາ ໂດຍ component
- Creates and manages by component  
ກຸມສະໜອງຈະເຮັດໃນ component ໃນ
- State cannot be passed to other components
- \*\*used inside component only\*\*  
ເຊື່ອ state ເລີຍາ ດົນ DOM ມີ re-render ຖ້າ ແຕກໍາໄປໂຄງ
- When state changes, DOM re-rendering will occur
  - Just a part of DOM will be updated
  - Or more specific, only updated component will be re-rendered

# State example

```
class ShowTitle extends Component {  
    state = { class name is state  
        | datetime: new Date()  
    }  
    | myupdate1207  
    updateDatetime() {  
        | get on state in field  
        | this.setState({datetime: new Date()});  
    }  
  
    render() {  
        return (  
            <div>  
                <p>{this.state.datetime.toString()}</p>  
                <a href="https://sv1.picz.in.th/images/2022/02/06/nM1SyZ.jpg"  
                    onMouseOver={this.updateDatetime.bind(this)}>Headline {this.props.title} </a>  
            </div>  
        );  
    }  
}
```

↑

မျှမှန်၍ Header အတွက် func updateDatetime ဖြစ်သော်လည်း  
မျှမှန်၍ ပြုလုပ်နိုင်မှုမှာ မူလဲ၏ Header

မျှမှန်၍ မူလဲ၏ Header

# State & Props

	Props	State
Usage  Passing data  ส่งค่าจาก non UI component	Passing data  inside the component only	Inside the component only
Data modification  data ที่เปลี่ยน  modification	Read only  เข้าถึงแต่เดียว	Yes, locally  set ค่าโดยใช้ฟังก์ชัน function set state

Ex

# More with state

Result

Sat Feb 19 2022 16:25:50 GMT+0700 (Indochina Time)

Headline Headline 01 10++

1  
2  
3

```
class ShowTitle extends Component {  
  state = {  
    ticker: false,  
    count: 0,  
    datetime: new Date()  
  }  
  // 3 3 field  
  // 10++ m. วน: ที่ส่วน component ที่มี state
```

```
updatetime() {  
  var num = this.state.count;  
  
  num++;  
  if (num>10)  
    this.setState({ticker: true});  
  
  this.setState({  
    count: num,  
    datetime: new Date()  
  });  
}
```

```
render() {  
  var textval="";  
  if (this.state.ticker==true)  
    textval = '10++';  
  else  
    textval = this.state.count;
```

```
return (  
  <div>    convert datetime to str  
    <p>{this.state.datetime.toString()}</p>  
    <a href="https://sv1.picz.in.th/images/2022/02/06/nM1SyZ.jpg" 2  
      onMouseOver={this.updatetime.bind(this)}>Headline {this.props.title} </a>  
    <input type="text" value={textval} readOnly></input>  
  </div>  
);  
}
```

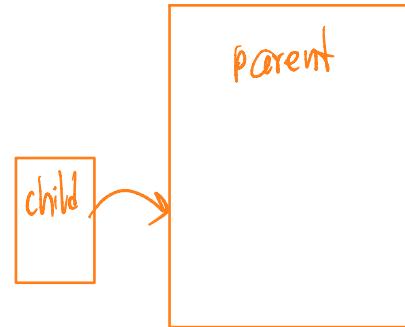
10:25:00 Jul 2061 ①  
Headline ②  
1 ③ ตัวเลขที่ไม่ตรง  
10++ ยกเว้น 11 ตัวเลข

# Passing data

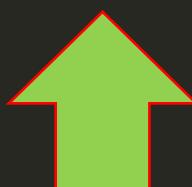
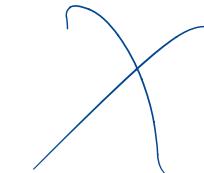
- From parent to child
  - Props

# Passing data

- From child to parent *给孩子一个* →
  - Callback function *回调*



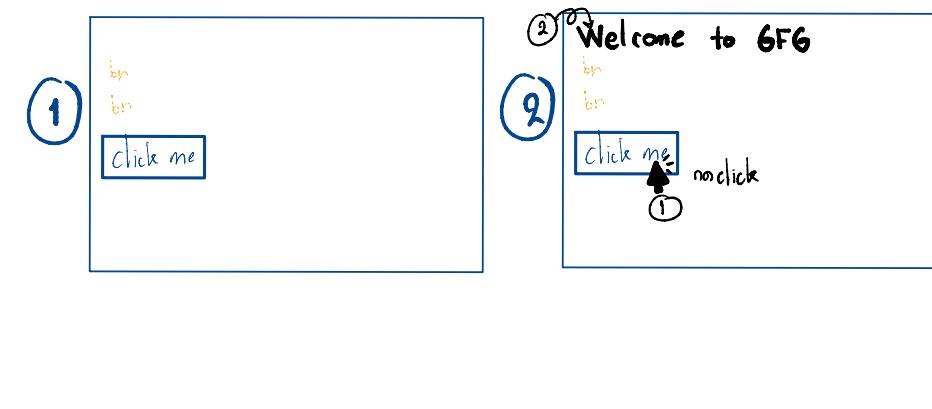
```
class Parent extends React.Component{  
    state = {  
        msg: "",  
    }  
  
    handleCallback = (childData) =>{  
        this.setState({msg: childData})  
    }  
  
    render() {  
        const {msg} = this.state;  
        return(  
            <div>  
                <h1> {msg}</h1>  
                <Child parentCallback = {this.handleCallback}/>  
            </div>  
        );  
    }  
}
```



```
class Child extends React.Component {  
    →onTrigger = () => {  
        this.props.parentCallback("Welcome to GFG");  
    };  
  
    render() {  
        return (  
            <div>  
                <br></br> <br></br>  
                <button onClick={this.onTrigger}>Click me</button>  
            </div>  
        );  
    }  
}
```



```
class Parent extends React.Component{  
    state = {  
        msg: "",  
    }  
  
    handleCallback = (childData) =>{  
        this.setState({msg: childData})  
    }  
  
    render() {  
        const {msg} = this.state;  
        return(  
            <div>  
                <h1> {msg}</h1>  
                <Child parentCallback = {this.handleCallback}/>  
            </div>  
        );  
    }  
}
```



```
class Child extends React.Component {  
    onTrigger = () => {  
        this.props.parentCallback("Welcome to GFG");  
    };  
  
    render() {  
        return (  
            <div>  
                <br></br> <br></br> ① click  
                <button onClick={this.onTrigger}>Click me</button>  
            </div>  
        );  
    }  
}
```

# exercise

uic平臺上：如何在滑動  
視窗上，加上進度條

