

## บทที่ 7 เลือกใช้ JS เฟรมเวิร์ค

### Choose JS Frameworks

ผู้สอน อ.ปรีม ชฎารัตนจิตติ

## ขั้นตอนที่ 3 เลือกใช้เฟรมเวิร์ค (Choose Frameworks)

---

- ▶ ส่วนของ JavaScript ควรเรียนรู้อย่างน้อย 1 ตัว ที่เป็นเฟรมเวิร์คยุคปัจจุบัน ได้แก่ React.js, jQuery, Angular.js , Vue.js , W3.JS
- ▶ ในประเทศไทยมีหลากหลายบริษัทที่ต้องการนักพัฒนา AngularJS Vue.js และ React.js ค่าจ้างอยู่ในอัตราประมาณ 40,000-120,000 บาท



## jQuery

---

- ▶ jQuery เป็น JavaScript library
- ▶ jQuery ช่วยเพิ่มความสะดวกในการเขียน JavaScript อย่างมาก
- ▶ jQuery เรียนรู้ได้ง่าย
- ▶ jQuery มีขนาดเล็กและใช้ทรัพยากรน้อย “เขียนน้อย ได้ผลลัพธ์เยอะ”
- ▶ jQuery ใช้กับงานทั่วไปที่ปกติแล้วต้องเขียน JavaScript หลายบรรทัดเพื่อให้ได้ผลลัพธ์ เปลี่ยนมาเป็นการเรียกใช้ด้วยโค้ดเพียงบรรทัดเดียว
- ▶ jQuery ถูกนำมาใช้ในการพัฒนาเว็บไซต์ในบริษัทใหญ่ๆ หลายแห่ง เช่น
  - ▶ Google
  - ▶ Microsoft
  - ▶ IBM
  - ▶ Netflix
- ▶ `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>`



## ลักษณะการเขียน jQuery (jQuery Syntax)

---

- ▶ **jQuery syntax** มีรูปแบบเป็นการเลือก HTML elements และ ดำเนินการทำงาน(action) กับ element นั้นๆ

syntax พื้นฐานคือ : **\$(selector).action()**

- ▶ **\$** ใช้ประกาศ/เชื่อมต่อ jQuery
- ▶ **(selector)** ใช้ค้นหา HTML elements
- ▶ **action()** ใช้ดำเนินการตามที่ระบุกับ element



## ใช้ jQuery ซ่อนหรือแสดง element

---

- ▶ ด้วยการ ใช้ jQuery ช่วยให้สามารถซ่อนและแสดง HTML elements ด้วยการ ใช้ action ใน jQuery ที่ชื่อว่า hide() และ show()

- ▶ ตัวอย่างโค้ด

- ▶ 

```
$("#hide").click(function(){  
    $("#p").hide();  
});
```

### jQuery Hide and Show

- ▶ 

```
$("#show").click(function(){  
    $("#p").show();  
});
```

### jQuery Hide and Show

If you click on the "Hide" button, I will disappear.

Then, please click the "Show" button.

- ▶ ลิงค์ [https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis\\_jquery\\_hide](https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_jquery_hide)
- 



## jQuery Sliding Effect

---

- ▶ jQuery สามารถใช้สร้าง sliding effects สำหรับ HTML elements ได้ ด้วย slideDown() slideUp() และ slideToggle()

- ▶ ตัวอย่างโค้ด

- ▶ 

```
$("#flip").click(function(){  
    $("#panel").slideDown();  
});
```

### jQuery Slide

Click to slide down panel

London is the most populous city in the United Kingdom, with a metropolitan area of over 9 million inhabitants.

- ▶ 

```
$("#flip").click(function(){  
    $("#panel").slideUp();  
});
```

### jQuery Slide

Click to slide up panel

London is the most populous city in the United Kingdom, with a metropolitan area of over 9 million

---

▶

# jQuery Animation

---

▶ jQuery ใช้สร้างการเคลื่อนไหว โดยการกำหนดในลักษณะของ CSS ได้

▶ ตัวอย่างโค้ด

```
▶ $("button").click(function(){  
    $("div").animate({  
        left: '250px',  
        height: '+=150px',  
        width: '+=150px'  
    });  
});
```

## jQuery Animation

Start Animation

By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!



## jQuery Animation

Start Animation

By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!



## jQuery Manipulating CSS

---

- ▶ jQuery ใช้จัดการ CSS ได้ เช่น `addClass()` `removeClass()` หรือ `toggleClass()`

- ▶ ตัวอย่างโค้ด

- ▶ 

```
$("#button").click(function(){  
    $("#h1, h2, p").toggleClass("blue");  
});
```

### Heading 1

This is a paragraph.

Toggle class

### Heading 1

This is a paragraph.

Toggle class

---

▶



## AngularJS

---

- ▶ **AngularJS** ช่วยเพิ่มเติม HTML attributes ที่เรียกว่า directives
- ▶ **AngularJS directives** เพิ่มฟังก์ชันลงไปใน HTML ธรรมดา
- ▶ **AngularJS** มี directives แบบพื้นฐาน และ directives แบบที่ผู้ใช้เขียนเอง



- ▶ `<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">`



## AngularJS Directives

---

- ▶ **AngularJS** ใช้ `{{ }}` ใช้แสดงตำแหน่งข้อมูลลักษณะคล้ายตัวแปร
- ▶ **AngularJS directives** เป็น HTML attributes มีคำนำหน้าด้วย `ng-`
- ▶ ต้องใส่ `ng-app` เพื่อให้มีการประมวลผลแบบ AngularJS แอปพลิเคชัน
- ▶ ใส่ `ng-init` เพื่อประมวลข้อมูล
- ▶ ตัวอย่างโค้ด
- ▶ 

```
<div ng-app="" ng-init="message='Hello AngularJS!'">  
  <h1>{{ message }}</h1>  
</div>
```

**Hello AngularJS!**

---

## ng-model Directive

---

- ▶ ใช้ **ng-model** เพื่อเชื่อมค่าของข้อมูลใน HTML elements กับข้อมูลใน application data ของ AngularJS

- ▶ ตัวอย่างโค้ด

- ▶ 

```
<div ng-app="" ng-init="firstName='John'">  
<p>Name: <input type="text" ng-model="firstName"></p>  
<p>You wrote: <b>{{ firstName }}</b></p>  
</div>
```

### AngularJS

Input something in the input box:

Name:

You wrote: **John**

### AngularJS

Input something in the input box:

Name:

You wrote: **Johny boy**

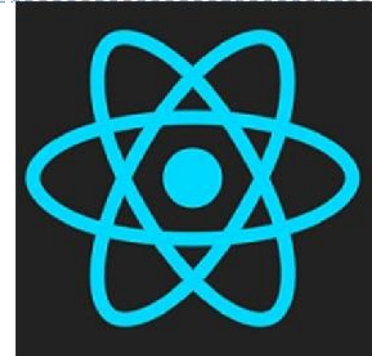
---



## React

---

- ▶ **React** เป็น JavaScript library สร้างโดย Facebook
- ▶ **React** เป็น **User Interface (UI)** library
- ▶ **React** เป็นเครื่องมือใช้ในการสร้าง **UI component**
- ▶ ในการเขียน React ควรจะมีความเข้าใจพื้นฐานเกี่ยวกับ
  - ▶ HTML
  - ▶ CSS
  - ▶ DOM
  - ▶ ES6
  - ▶ Node.js
  - ▶ npm



## React CDN

---

▶ **<!-- Load React API -->**

```
<script src="https://unpkg.com/react@16/umd/react.production.min.js"></script>
```

**<!-- Load React DOM-->**

```
<script src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"></script>
```

**<!-- Load Babel Compiler -->**

```
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
```

▶ `<script type="text/babel">`

```
  // JSX Babel code goes here
```

```
</script>
```

---



## Babel

---

- ▶ **Babel** เป็น JavaScript compiler ที่ใช้แปลงข้อมูลหรือ ภาษาโปรแกรมต่างๆ ให้อยู่ในรูปแบบ JavaScript.
- ▶ ด้วยการ ใช้ Babel จะทำให้สามารถใช้ฟีเจอร์ใหม่ๆ ใน JavaScript (ECMAScript 6 หรือ ES6) ได้
- ▶ React ใช้ Babel เพื่อแปลง JSX ไปเป็น JavaScript จึงสำคัญมากที่เวลาเขียนต้องมี `<script type="text/babel"> </script>`



## JSX

---

- ▶ **JSX** มาจากคำว่า **JavaScript XML**.
- ▶ **JSX** เป็น XML/HTML เป็นเหมือนส่วนเสริมสำหรับ JavaScript

- ▶ ตัวอย่างโค้ด

```
const element = <h1>Hello World!</h1>
```

- ▶ JSX ไม่ใช่ทั้ง JavaScript หรือ HTML คล้ายกับ HTML JSX tags มีได้ทั้ง tag names, attributes และ children



## React DOM Render

---

- ▶ React จะใช้ method ชื่อ ReactDOM.render() เพื่อแสดงผล HTML elements:

- ▶ ตัวอย่างโค้ด

- ▶ `<div id="id01"></div>`

```
<script type="text/babel">
```

```
  ReactDOM.render(
```

```
    <h1>Hello React!</h1>,
```

```
    document.getElementById('id01'));
```

```
</script>
```

**Hello React!**





## JSX Expressions

---

- ▶ หากต้องใช้ Expressions (รับค่าตัวแปร) ให้ใช้ JSX ด้วยการเขียนลงไปนใน {}

- ▶ ตัวอย่างโค้ด

- ▶ `<div id="id01">Hello World!</div>`

```
<script type="text/babel">
```

```
const name = 'John Doe';
```

```
ReactDOM.render(
```

```
  <h1>Hello {name}!</h1>,
```

```
  document.getElementById('id01'));
```

```
</script>
```

**Hello John Doe**



## React Code

---

- ▶ React elements จะไม่มีการเปลี่ยนแปลงตัวเองเมื่อมีการเปลี่ยนข้อมูล
- ▶ หากต้องการให้ React แสดงข้อมูลใหม่ ต้องสั่ง render ใหม่ทุกครั้ง

### ▶ ตัวอย่างโค้ด

```
function tick() {  
  const element = (<h1>{new Date().toLocaleTimeString()}</h1>);  
  ReactDOM.render(element, document.getElementById('root'));  
}  
setInterval(tick, 1000);
```

**11:24:13 AM**



## React Component Properties

---

- ▶ ตัวอย่าง การเขียน React **component** ชื่อว่า "Welcome" โดยมีค่าตัวแปร แบบ ES6

- ▶ `class Welcome extends React.Component {`

```
  render() { return(<h1>Hello {this.props.name}</h1>); }
```

**Hello John Doe!**

```
}
```

```
ReactDOM.render(<Welcome name="John Doe"/>, document.getElementById('root'));
```



## Vue.js

---

- ▶ **Vue.js** เสริมการเขียน HTML ด้วย HTML attributes ที่เรียกว่า **directives**
- ▶ **Vue.js directives** เสริมการเขียนฟังก์ชันลงไป ในภาษา HTML
- ▶ **Vue.js** มี directives แบบพื้นฐาน และ directives แบบที่ผู้ใช้เขียนเอง



## Vue.js Directives

---

- ▶ **Vue.js** ใช้ `{{ }}` แทนส่วนของข้อมูล (ตัวแปร)
- ▶ **Vue.js directives** เป็น HTML attributes ที่นำหน้าด้วย `v-`

- ▶ ตัวอย่างโค้ด

- ▶ `<div id="app">`

- `<h1>{{ message }}</h1>`

**Hello Vue!**

- `</div>`

- `<script>`

- `var myObject = new Vue({`

- `el: '#app',`

- `data: {message: 'Hello Vue!'}`

- `})`

- `</script>`

---



## Vue.js Binding

---

- ▶ สามารถเชื่อม Vue object เข้ากับ HTML element เมื่อมีการเปลี่ยนแปลงข้อมูลทั้ง HTML element ก็จะเปลี่ยนตาม

- ▶ ตัวอย่างโค้ด

```
<div id="app">{{ message }}</div>
```

```
<button onclick="myFunction()">Click Me!</button>
```

```
<script>
```

```
var myObject = new Vue({el: '#app', data: {message: 'Hello Vue!'}})
```

```
function myFunction() {myObject.message = "John Doe";}
```

```
</script>
```

**Vue.js**

Hello Vue!

Click Me!

**Vue.js**

John Doe

Click Me!

---

