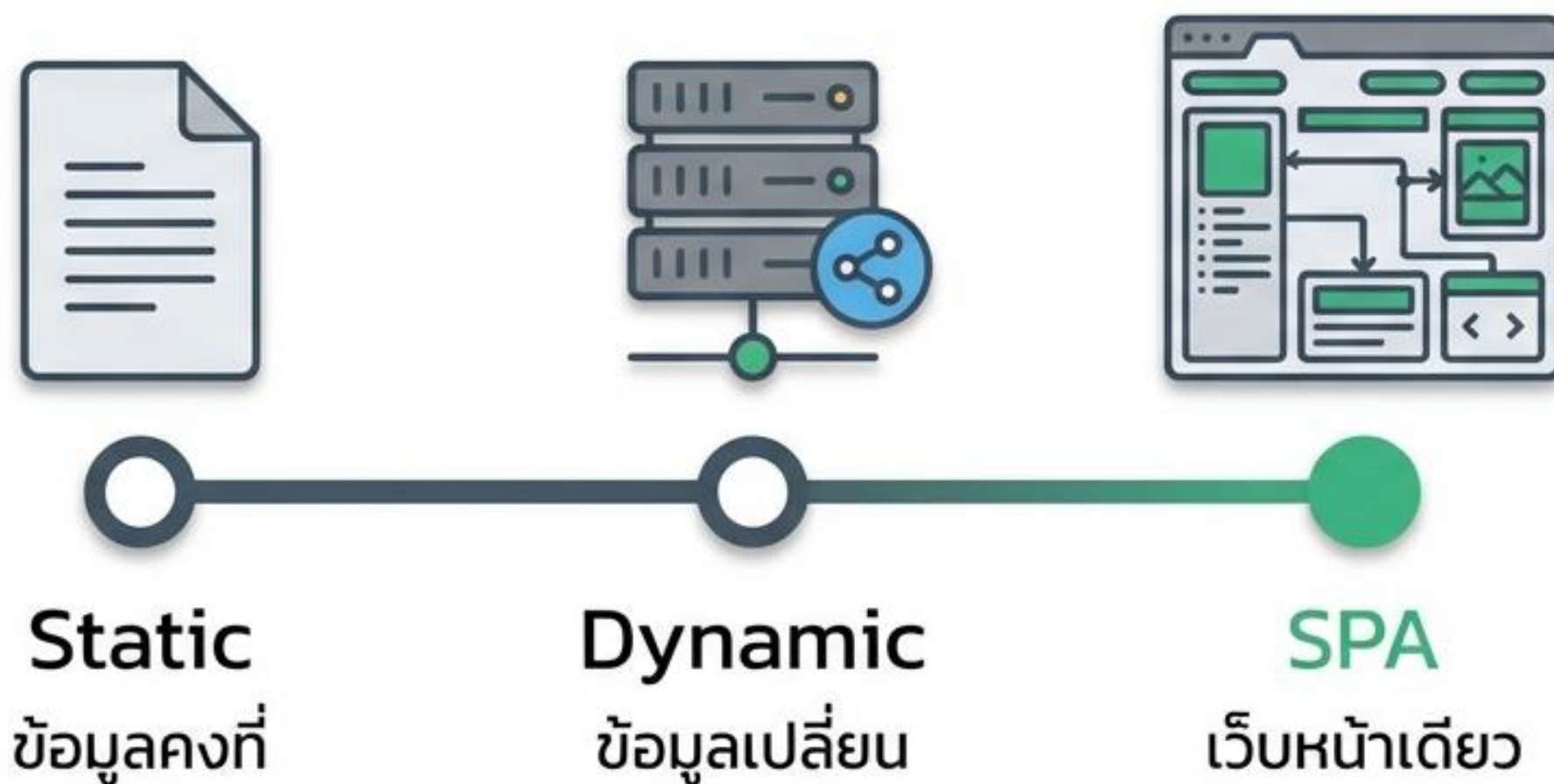




การพัฒนา Web UI ด้วย Vue Framework

วิชา: การออกแบบและเขียนโปรแกรมเว็บเพจ

ผู้สอน: พศ.สมเกียรติ ช่อเหมือน
Email: tko@webmail.npru.ac.th
สาขา: วิศวกรรมซอฟต์แวร์ NPRU



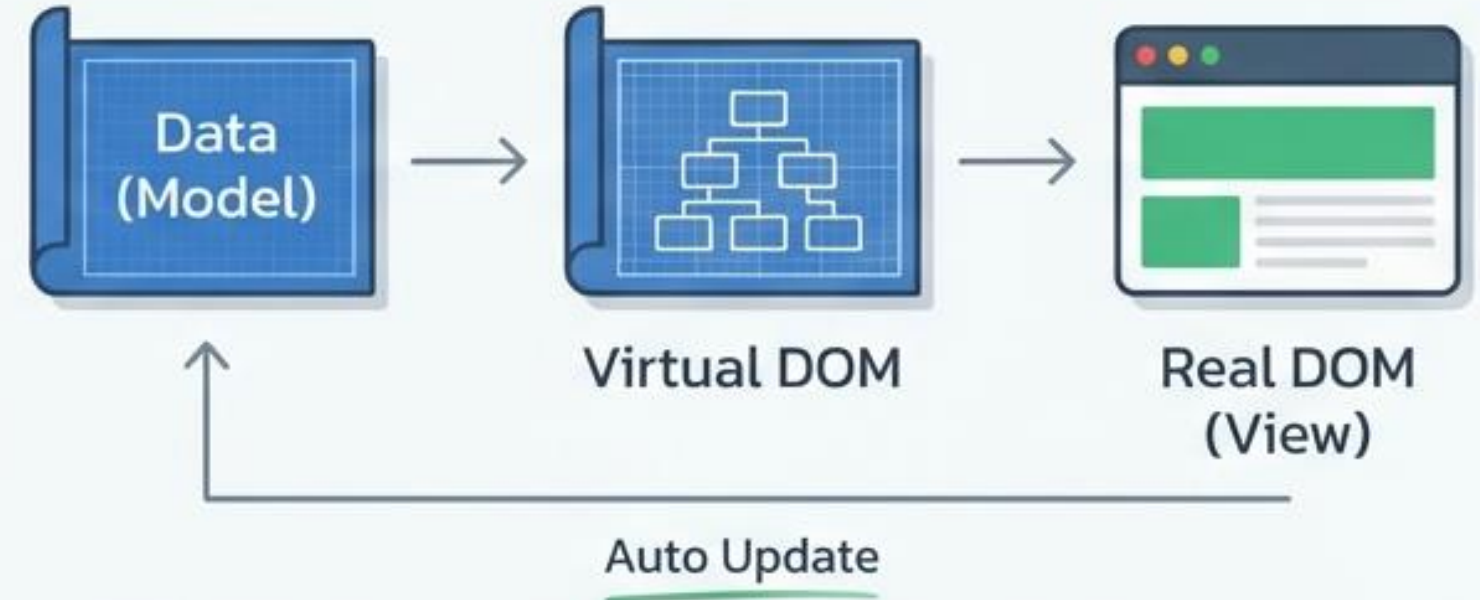
ทำไมต้องใช้ Framework?

- ทำงานอย่างเป็น Standard
- มีระบบ Template ที่ชัดเจน
- ช่วยให้นักพัฒนาโฟกัสที่ Logic

Problem: การจัดการสถานะที่ซับซ้อนต้องการเครื่องมือที่เป็นมาตรฐาน

Vue.js คืออะไร?

- คือ Progressive Framework ที่เน้นส่วน View Layer เป็นหลัก
- สร้างโดย Evan You ในปี 2014
- จุดเด่น: **Reactivity**
 - ข้อมูลเปลี่ยน หน้าจอเปลี่ยนทันที
 - ไม่ต้องรีเฟรชหน้าเบราว์เซอร์



Vue.js Setup (JS)

```
const app = Vue.createApp({ ... })  
app.mount('#app')
```

ติดตั้งผ่าน CDN เพื่อเริ่มต้นเร็ว

Mustache Syntax {{ }}
ผูกข้อมูลจาก JS สู่ HTML

HTML Template

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

Hello World

v-bind: การผูกคุณสมบัติ

เราไม่สามารถใช้ {{ }} ใน HTML Attributes ได้

```
<!-- ผูกค่า src กับตัวแปร imageURL -->
```

```
<img v-bind:src='imageURL'>
```

```
<!-- หรือใช้ตัวย่อ (Shorthand) -->
```

```
<img :src='imageURL'>
```

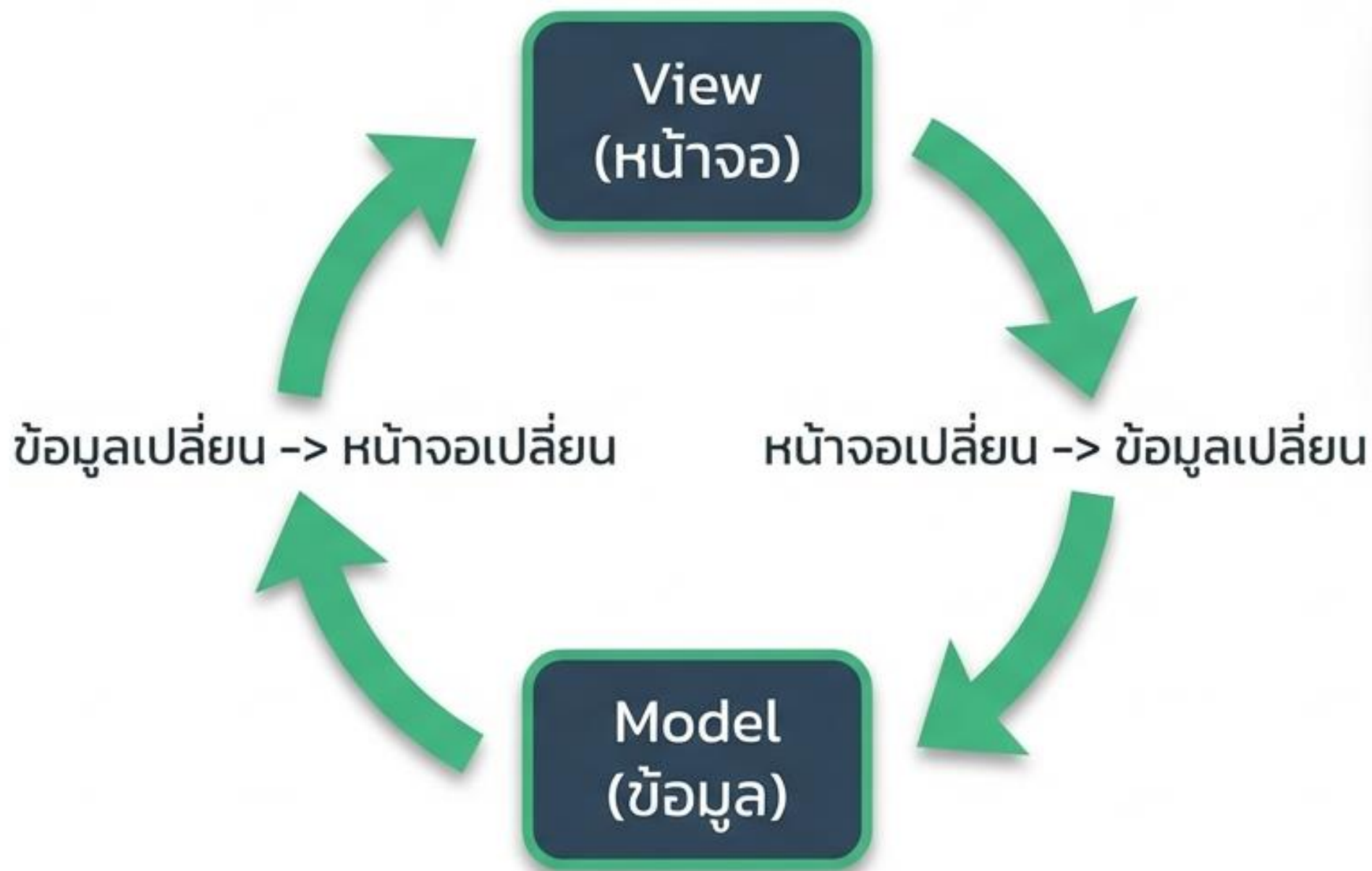
imageURL

'https://example.com/landscape.jpg'



เปลี่ยนรูปภาพแบบ Dynamic

v-model: ผูกข้อมูลสองทิศทาง (Two-way Binding)



```
<input v-model='username'>
<p>สวัสดี: {{ username }}</p>
```

Mario

สวัสดี: Mario

v-if: เงื่อนไขการแสดงผล (Logic)

ควบคุมการแสดงผลด้วย Boolean (True/False)



```
<p v-if='seen'>คุณเห็นฉันแล้ว</p>  
<p v-else>ฉันกำลังซ่อนอยู่</p>
```

ON



seen = true

คุณเห็นฉันแล้ว

Removed from DOM

ลบ/เพิ่ม แท็กใน DOM จริง (ไม่ใช่แค่ซ่อน CSS)

v-for: แสดงผลรายการ (Loop)



```
<ul>
  <li v-for='item in items'>
    {{ item.name }}
  </li>
</ul>
```

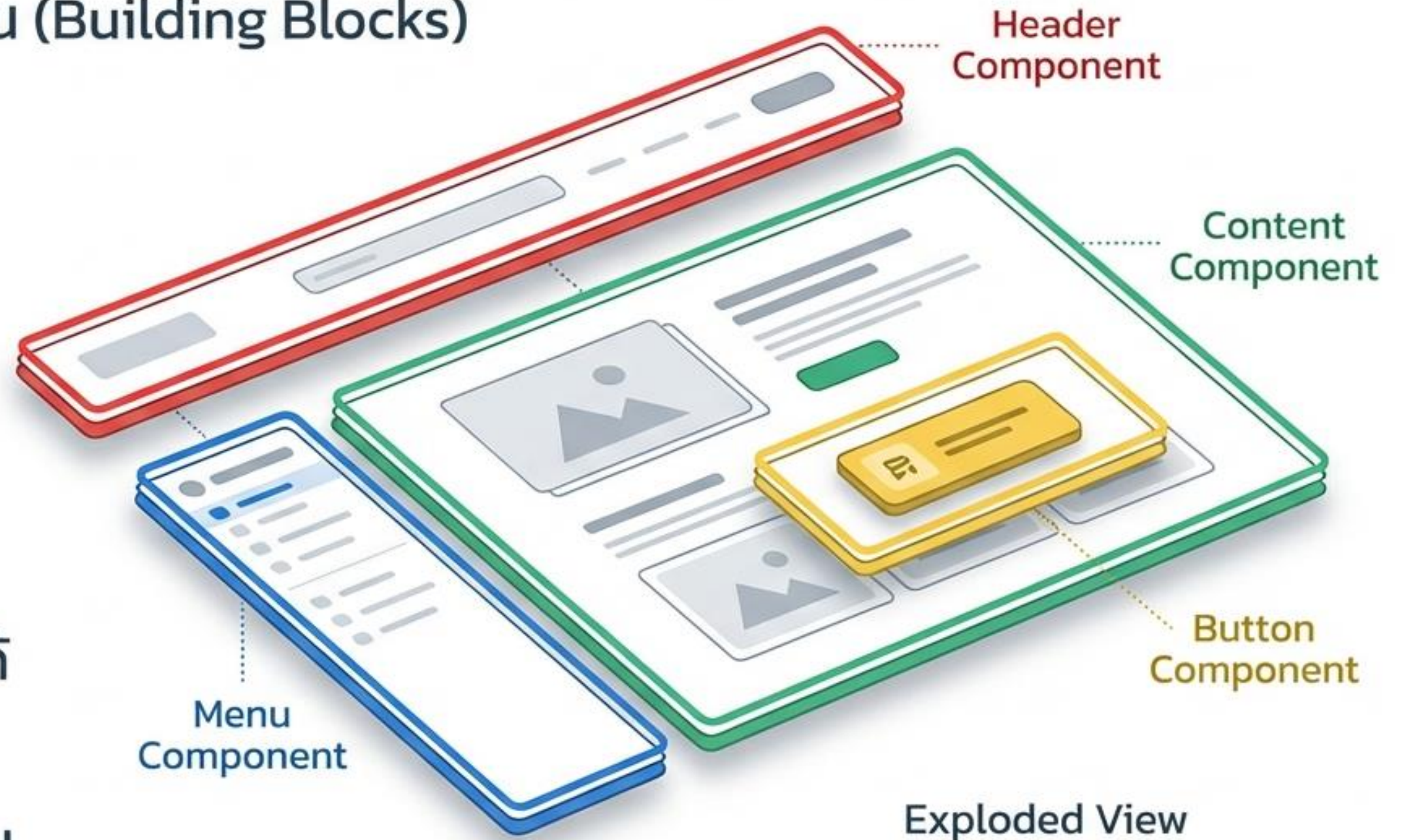
[{name: 'A'},
 {name: 'B'},
 {name: 'C'}]

- Item A
- Item B
- Item C



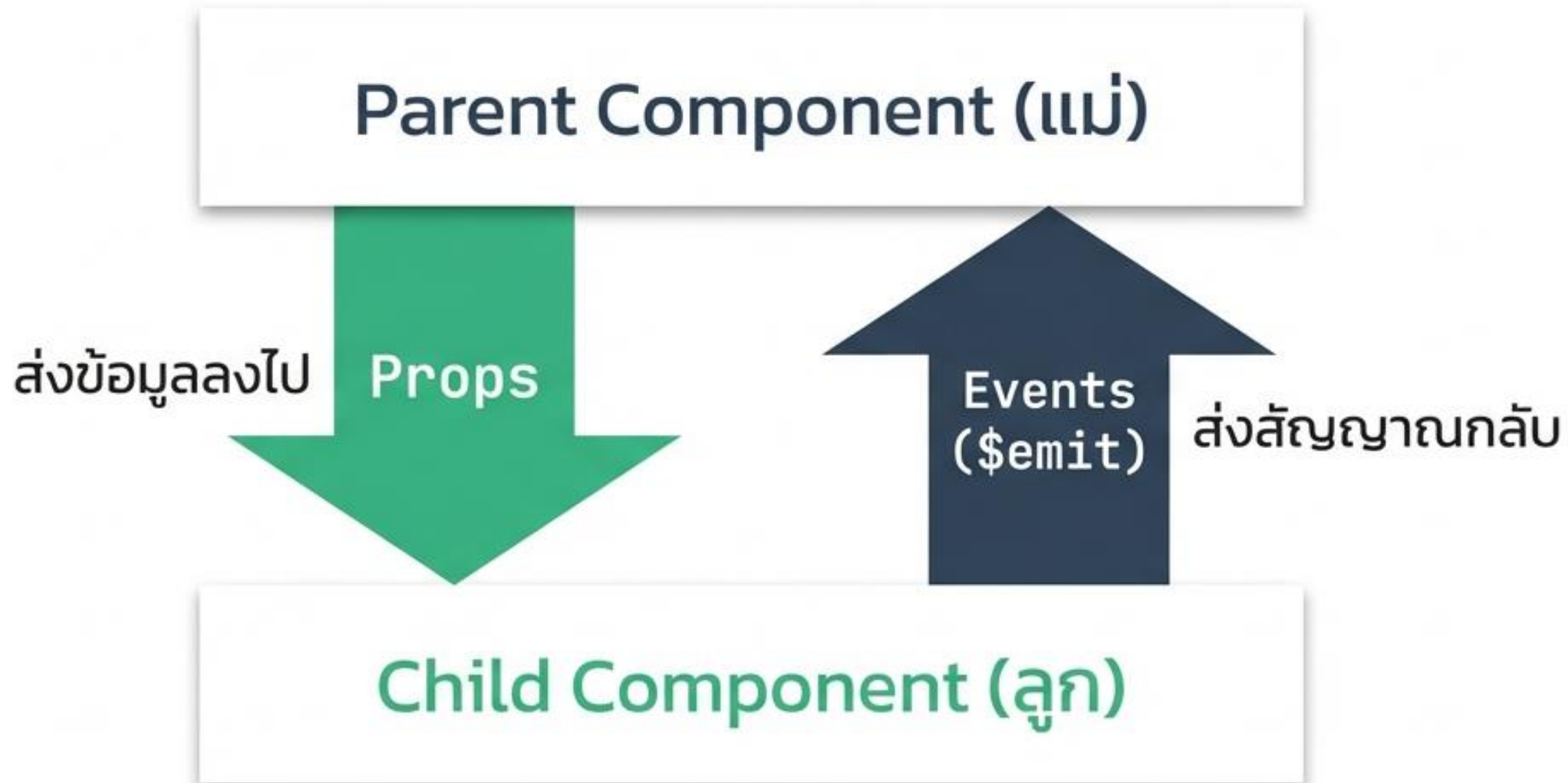
ระบบ Components

แยกส่วนหน้าจอเป็นชิ้นส่วน (Building Blocks)



- นำชิ้นส่วนกลับมาใช้ซ้ำได้ (Reusable)
- จัดการโค้ดได้เป็นระเบียบ

การส่งข้อมูลระหว่างคอมโพเนนต์



One-Way Data Flow (ข้อมูลไหลทางเดียว)

วงจรชีวิต (Lifecycle Hooks)



onMounted

เมื่อโหลดเสร็จ (ถึง API)

onUpdated

เมื่อข้อมูลเปลี่ยน

onUnmounted

เมื่อปิดหน้า (Cleanup)

Computed Properties: ฉลาดกว่า Method

- **Caching:** คำนวณค่าและเก็บไว้
- **Performance:**
เร็วกว่าการใช้ Method ปกติ
- **Reactive:**
อัปเดตเมื่อข้อมูลตั้งต้นเปลี่ยน

```
computed: {  
  totalPrice() {  
    return this.price * this.quantity  
  }  
}
```



ปฏิบัติ: แอปคำนวณเงิน (Calculator)

Price Calculator

ราคาสินค้า (Price)

5,000

จำนวน (Qty)

1

ภาษี (Tax)

350

ยอดรวม: 5,350 บาท

- 1. รับราคาสินค้าและภาษี
-> v-model
- 2. ใช้ Computed หาผลรวม
-> $(Price * Qty) + Tax$
- 3. แสดงผลลัพธ์แบบเรียลไทม์
-> Reactivity

ข้อแนะนำในการพัฒนา (Best Practices)



แยก **CSS** และ **Logic** ชัดเจน



ใช้ **SFC (.vue files)** เมื่อแอปใหญ่



ตั้งชื่อคอมโพเนนต์ **สองคำ** (Two-word naming)



ห้ามแก้ไข **Props** โดยตรง



หมั่นตรวจสอบผลผ่าน **Vue DevTools**

สรุปบทเรียน

-  Vue คือเครื่องมือสมัยใหม่ (Modern)
-  เน้นความ Reactive และง่าย
-  โครงสร้างแบบ Components
-  หัวใจคือการจัดการสถานะเว็บ

แหล่งข้อมูล & ติดต่อ

-  Docs: vuejs.org
-  Book: Mastering Vue.js 3
-  ผู้สอน: พศ.สมเกียรติ ช่อเหมือน
-  Email: tko@webmail.npru.ac.th

ลงมือทำ คือกุญแจสำคัญ (Action is Key)