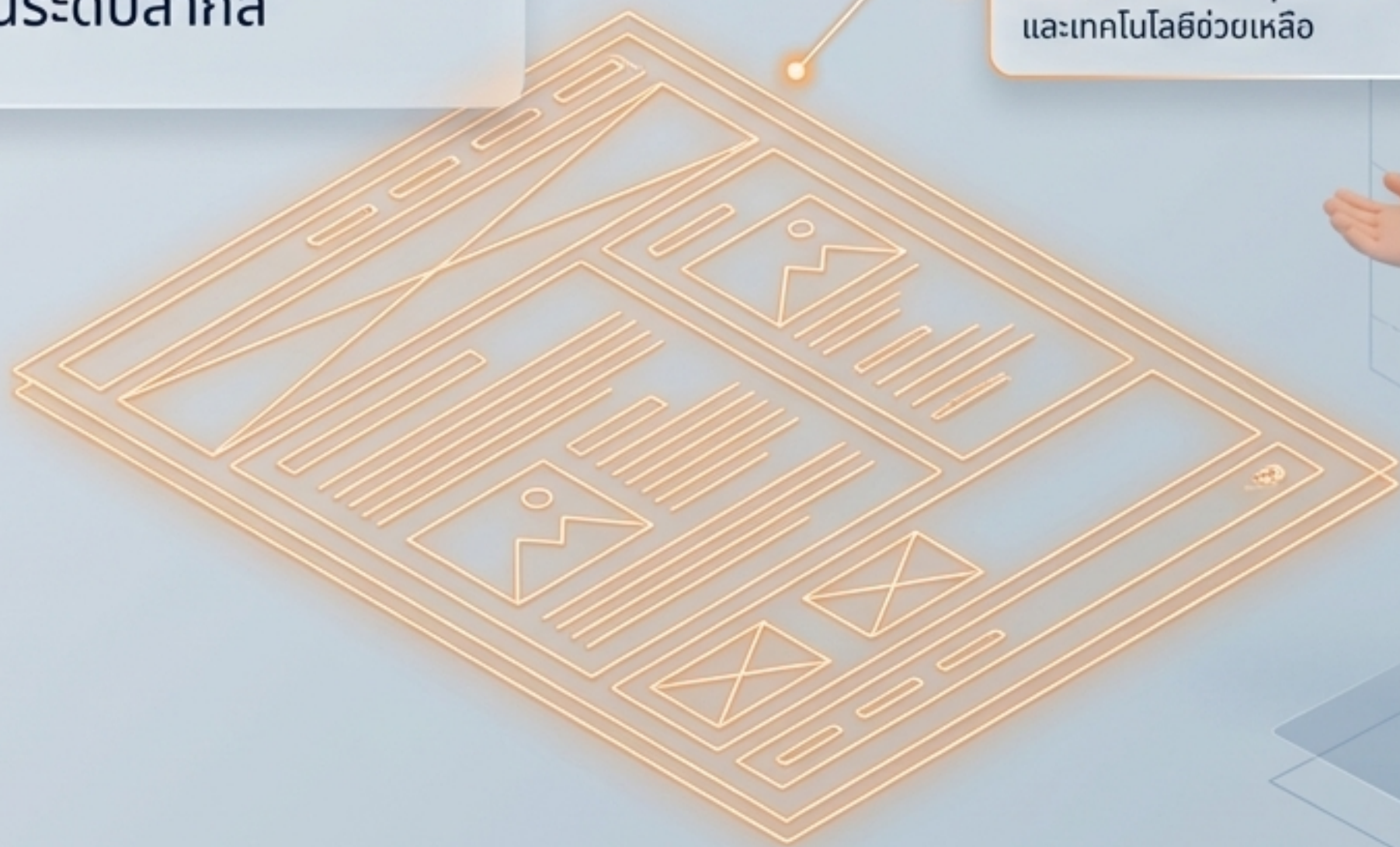


# สถาปัตยกรรมโครงสร้างเว็บฉบับพิมพ์เขียว

HTML ไม่ใช่เพียงชุดคำสั่ง  
แต่คือสถาปัตยกรรมพื้นฐานที่กำหนด  
วิธีสื่อสารข้อมูลในระดับสากล

โครงสร้างหลัก (Skeleton)  
ที่เชื่อมต่อระหว่างมนุษย์ บอทค้นหา  
และเทคโนโลยีช่วยเหลือ



# กายวิภาคพื้นฐานขององค์ประกอบ HTML

**แท็กเปิด (Opening Tag):**  
กำหนดจุดเริ่มต้นและประเภทข้อมูล  
(ใช้วงเล็บมุม < >)

**เนื้อหา (Content):**  
ข้อมูลดิบที่ต้องการแสดงผล

**แท็กปิด (Closing Tag):**  
ระบุจุดสิ้นสุด (มีเครื่องหมาย//)

**<h1>หัวข้อรายงานวิจัย</h1>**

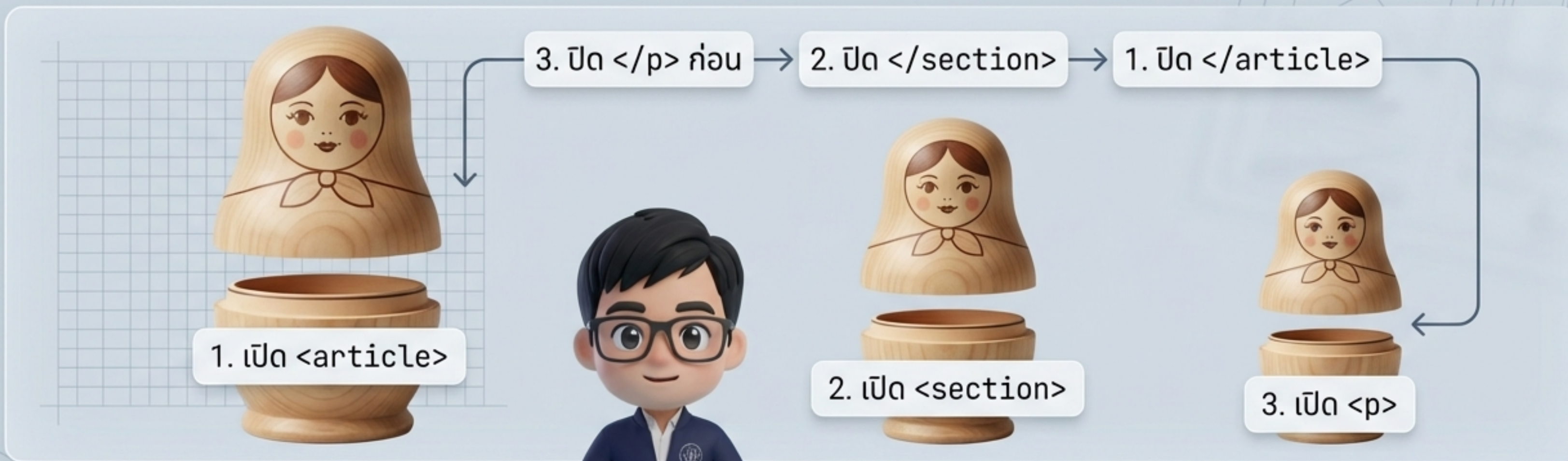
**แท็กปิด (Closing Tag):**  
ระบุจุดสิ้นสุด (มีเครื่องหมายสแลช /)

## ข้อควรระวัง:

การลืมใส่แท็กปิดคือข้อผิดพลาดร้ายแรง  
ทำให้เบราว์เซอร์ต้องใช้การคาดเดา (Parser  
Guessing) ซึ่งมักนำไปสู่การแสดงผลที่ผิดเพี้ยน



# กฎหลักการซ้อนทับ: เข้าก่อนต้องออกหลัง



## Last-In-First-Out (LIFO):

องค์ประกอบที่เปิดลำดับสุดท้าย ต้องถูกปิดเป็นลำดับแรกเสมอ

## ! DOM Impact

หากปิดแท็กนอกลำดับ (เช่น ปิดตัวใหญ่ก่อนตัวเล็ก) จะทำให้โครงสร้าง Document Object Model (DOM) เสียหาย และส่งผลกระทบต่อการทำงานของ JavaScript ในภายหลัง

# การเพิ่มบริบทและฟังก์ชันด้วยแอตทริบิวต์

`class="main-title"`

ชื่อ

ค่า

## id

ระบุตัวตนเฉพาะ (Unique Identifier)  
เป้าหมายการลิงก์และอ้างอิงสคริปต์

## class

จัดกลุ่มเพื่อลดความซ้ำซ้อนของโค้ด CSS

## lang

จำเป็นยิ่งยวดสำหรับโปรแกรมอ่านหน้าจอ  
(Screen Readers) ในการออกเสียง

## tabindex

กำหนดลำดับโฟกัสคีย์บอร์ด (ค่า 0 =  
ลำดับธรรมชาติ, -1 = จัดการผ่านสคริปต์)

**!** **หมายเหตุ:** Boolean Attributes ทำงานทันทีที่ปรากฏชื่อ (เช่น disabled) โดยไม่ต้องมีค่า



# องค์ประกอบอิสระที่ไม่ต้องการแท็กปิด

องค์ประกอบว่าง (Void Elements): ไม่มีเนื้อหาภายใน ไม่ต้องการแท็กปิด  
ทำหน้าที่แทรกทรัพยากรหรือเปลี่ยนเลย์เอาต์ระดับบรรทัด



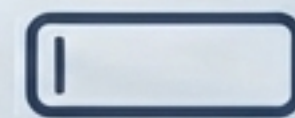
<img>



<hr>



<br>



<input>



**มาตรฐาน HTML5:** การใส่ Self-closing slash (เช่น <br />) เป็นเพียงตัวเลือกเสริม เบราวเซอร์จะเพิกเฉย

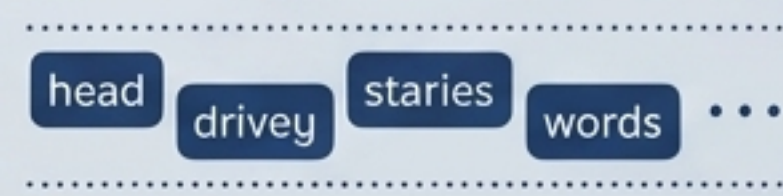


**ข้อควรระวัง:** ห้ามใส่สแลชในองค์ประกอบปกติ เช่น <script /> หรือ <ul /> เพราะเบราว์เซอร์จะมองว่าแท็กยังเปิดอยู่ ทำให้เลย์เอาต์หน้าเว็บพัง

# พฤติกรรมการจัดวาง: บล็อก ปะทะ อินไลน์



Block-level Elements



Inline Elements

การขึ้นบรรทัดใหม่

ขึ้นบรรทัดใหม่เสมอ

ต่อเนื่องในบรรทัดเดิม

ความกว้าง

เต็ม 100% ของ Parent

แคบเท่าขนาดเนื้อหา

การบรรจุ

ใส่ได้ทั้ง Block และ Inline

ใส่ได้แค่ Inline ด้วยกัน

การกำหนดขนาด CSS

รองรับทุกมิติ Width/Height

ไม่ส่งผลต่อการปลักเนื้อหาแนวตั้ง

**⚠ Tip:** ใช้ `display: inline-block` เพื่อรวมข้อดีของทั้งสองฝั่ง (วางแนวนอนได้แบบ Inline แต่กำหนดขนาดได้แบบ Block)

# อรรถศาสตร์ HTML: การให้ความหมายเหนือรูปแบบ



การใช้ <div> ทำปุ่ม อาจดูเหมือนปุ่มผ่าน CSS  
แต่ขาดความหมายเชิงบริบท



องค์ประกอบเชิงอรรถศาสตร์ (Semantic Elements)  
ทำหน้าที่สื่อสารหน้าที่ของข้อมูลอย่างชัดเจนแก่  
มนุษย์ เบราวเซอร์ บอทค้นหา และเทคโนโลยีช่วยเหลือ



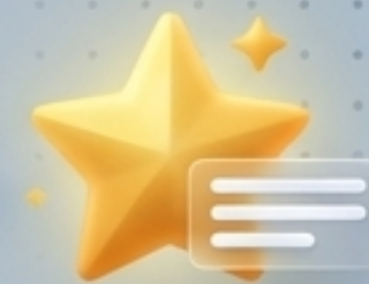
# ป้ายบอกทางสำหรับปัญญาประดิษฐ์และใช้งาน



SEO



```
<article>  
<h1>  
..
```



SEO: เครื่องมือค้นหาให้ความสำคัญในแท็ก Semantic มีน้ำหนักสูงกว่าแท็กทั่วไป

Accessibility

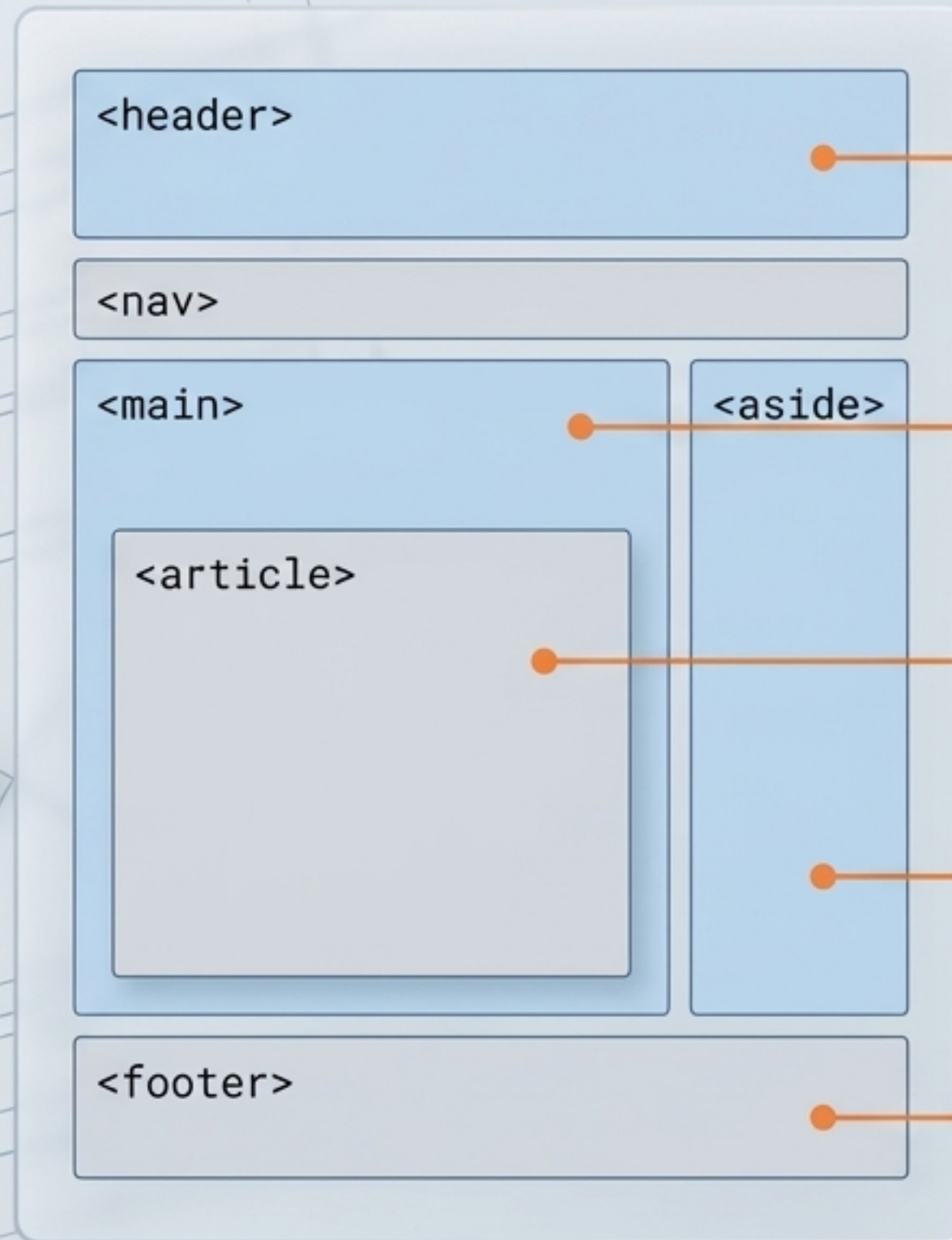


<nav>

<main>

Accessibility: โปรแกรมอ่านหน้าจอใช้แท็กเป็นป้ายบอกทางให้ผู้พิการทางสายตาระโดดข้ามเนื้อหาหรือนำทางได้อย่างอิสระ

# พิมพ์เขียวโครงสร้างอรรถศาสตร์บนหน้าเว็บ



ส่วนหัว รวบรวมโลโก้และชื่อไซต์

เนื้อหาหลักที่เป็นเอกลักษณ์ (ควรมีแค่ 1 อันต่อหน้า)

เนื้อหาอิสระ นำไปแชร์ต่อได้

เนื้อหาเสริม ชิวประวัติผู้เขียน  
โฆษณา

ข้อมูลลิขสิทธิ์และการติดต่อ

# สถาปัตยกรรมตารางเพื่อข้อมูลที่ซับซ้อน

โครงสร้าง `<table>`  
แบบใหม่แยกส่วนหัว  
(Header) เนื้อหา (Body)  
และส่วนท้าย (Footer)  
ชัดเจน เพื่อประสิทธิภาพเวลา  
เลื่อนหน้าจอ (Scrolling)  
หรือสั่งพิมพ์

`<thead>`

`<tbody>`

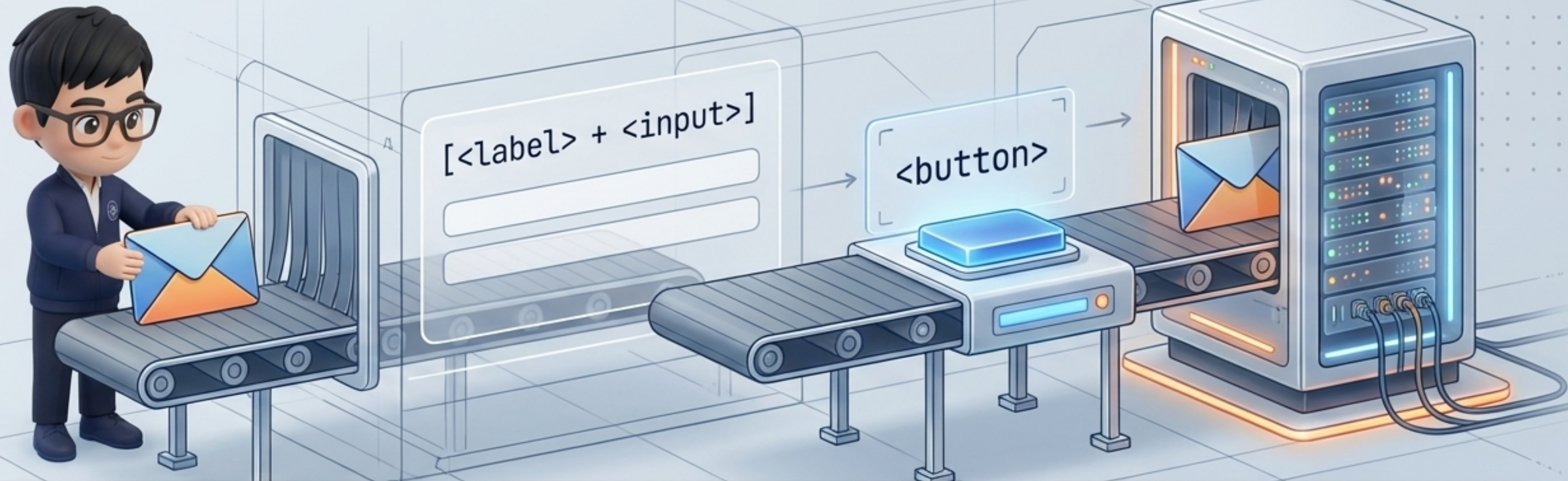
`<tfoot>`



**Accessibility Note:** การใช้แอตทริบิวต์ `scope` ในแท็ก `<th>` ช่วยระบุ  
ความสัมพันธ์ของแถวและคอลัมน์ ป้องกันความสับสนสำหรับโปรแกรมอ่านหน้าจอ



# กลไกโต้ตอบและส่งข้อมูลผู้ใช้



## <form>

ภาษาระบบจัดตัวควบคุม  
กำหนดปลายทางด้วย action  
และรูปแบบส่งข้อมูลด้วย method

## UX & Accessibility

การผูก <label> เข้ากับ <input>  
ไม่เพียงช่วย Screen Reader แต่ยัง  
ช่วยขยายพื้นที่การคลิก (Click Target)

## Tip

ใช้ <fieldset> และ <legend>  
จัดกลุ่มข้อมูลในฟอร์มยาวๆ  
เพื่อลดภาระทางพุทธิปัญญา

# กฎข้อห้ามเพื่อรักษาความถูกต้องของโครงสร้าง



## ข้อห้าม (Invalid)

**ห้าม:** ใส่ Block ไว้ใน `<p>`  
(**Why:** `<p>` รับผิดชอบข้อความ)

**ห้าม:** ซ้อนลิงก์ `<a>` ไว้ใน `<a>`  
(**Why:** พื้นที่คลิกทับซ้อน เมราว์เซอร์รวม)

**ห้าม:** ใส่ Block ไว้ใน `<span>`  
(**Why:** ฝ่าฝืนการไหลของ Inline)

**ห้าม:** ซ้อนแบบฟอร์ม `<form>`  
(**Why:** สับสนการประมวลผล)



## ทางออก (Valid)

**ทางออก:** ใช้ `<div>`

**ทางออก:** แยกปุ่มหรือจัดลำดับโครงสร้างใหม่

**ทางออก:** เปลี่ยนเป็น `<div>`

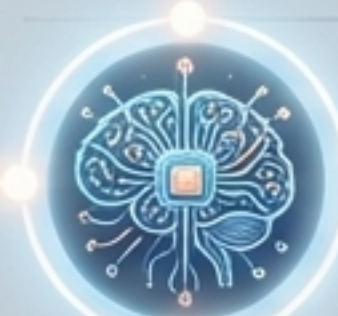
**ทางออก:** แยกฟอร์มให้ชัดเจน

# จากโค้ดมาร์กอัปสู่ภาษาสากลแห่งโลกอนาคต

มนุษย์  
(Responsive)



ปัญญาประดิษฐ์  
(Generative Search)



ผู้พิการ  
(Screen Readers)



อุปกรณ์อนาคต  
(Screenless Devices)



Valid  
Semantic  
HTML



- HTML คือระบบนิเวศข้อมูลที่เน้นความเป็นมนุษย์ (Human-centric)
- การเปลี่ยนผ่านสู่ **Semantic Era** ทำให้เว็บไซต์พร้อมสื่อสารกับเทคโนโลยี AI
- โครงสร้างเว็บที่ถูกต้องคือพันธกิจในการสร้างพื้นที่ดิจิทัลที่ยั่งยืน และพร้อมรองรับอุปกรณ์ไร้หน้าจอทุกแพลตฟอร์ม