



Tailwind CSS Framework:

Intermediate (Integrative-Generative AI Edition)

Student Price Book Center

Contents: Responsive Design, Hover, Focus, Pseudo States
Intermediate Typography, Color and Theme Management
Using Plugins, Bibliography

คำนำ

ในยุคของการพัฒนาเว็บไซต์ที่เปลี่ยนแปลงอย่างรวดเร็ว เฟรมเวิร์กสำหรับการจัดการส่วนติดต่อผู้ใช้งาน (UI) ได้เข้ามามีบทบาทสำคัญในการช่วยให้นักพัฒนาออกแบบและจัดวางโครงสร้างของเว็บได้อย่างมีประสิทธิภาพ และหนึ่งในเฟรมเวิร์กที่โดดเด่นที่สุดในยุคนี้คือ **Tailwind CSS** ซึ่งเป็น Utility-First CSS Framework ที่ให้ความยืดหยุ่นสูง เรียนรู้ง่าย และสามารถสร้าง UI ที่สวยงามได้โดยไม่ต้องเขียน CSS ใหม่ซ้ำซ้อน

หนังสือ **Tailwind CSS Framework: Intermediate** เล่มนี้ถูกออกแบบมาเพื่อผู้ที่มีพื้นฐานการใช้ Tailwind CSS อยู่แล้ว และต้องการต่อยอดไปสู่การใช้งานระดับกลางที่เน้นความเข้าใจเชิงลึก การปรับแต่ง การสร้างระบบ UI ที่ตอบสนองต่อผู้ใช้ และสามารถนำไปใช้ในโปรเจกต์จริงได้อย่างเป็นระบบ โดยมีการเรียบเรียงเนื้อหาให้เหมาะสำหรับการศึกษาทั้งด้วยตนเองและในชั้นเรียน

เริ่มต้นใน **บทที่ 6: Responsive Design** ผู้อ่านจะได้เข้าใจแนวคิด Mobile First Design และการใช้งาน Responsive Class ที่มีให้ใน Tailwind CSS เช่น md:flex, lg:grid เพื่อสร้างเว็บไซต์ที่ตอบสนองต่อหน้าจอขนาดต่าง ๆ ได้อย่างมีประสิทธิภาพ พร้อมตัวอย่างจริง เช่น Responsive Navbar และ Card Grid ที่สามารถนำไปประยุกต์ใช้ได้ทันที

บทที่ 7: Hover, Focus, Pseudo States จะนำเสนอการโต้ตอบกับผู้ใช้ผ่านสถานะขององค์ประกอบ เช่น hover:, focus:, active:, และ disabled: โดยมีทั้งแนวคิดและตัวอย่างการสร้าง Interactive UI อย่างปุ่มที่เปลี่ยนสีเมื่อ hover หรือ input ที่มี border เมื่อได้รับการ focus รวมถึงการใช้คลาส group-hover และ peer-* เพื่อควบคุมหลายองค์ประกอบพร้อมกัน

เข้าสู่ **บทที่ 8: Typography** ขั้นกลาง ผู้อ่านจะได้เรียนรู้การจัดการข้อความให้สื่อสารได้ชัดเจน และน่าอ่านมากขึ้น ไม่ว่าจะเป็นการกำหนดขนาด (text-lg), น้ำหนัก (font-bold), ระยะห่างบรรทัด (leading-*), หรือการปรับ letter spacing (tracking-*) และยังมีการใช้ plugin อย่าง @tailwindcss/typography ร่วมกับคลาส prose เพื่อจัดสไตล์บทความยาวให้อ่านง่ายและดูเป็นมืออาชีพ

ใน **บทที่ 9: การจัดการสีและ Theme** หนังสือจะพาผู้อ่านเข้าใจการใช้คลาสสีมาตรฐาน เช่น bg-blue-500, text-green-700 ตลอดจนการสร้าง custom color ด้วย tailwind.config.js เพื่อให้สอดคล้องกับ brand identity ของโปรเจกต์ และยังพูดถึงการใช้ opacity, state-based colors (เช่น hover:bg-blue-700) และแนวทางการจัดการ theme ที่ยืดหยุ่น รองรับการขยายในอนาคต

บทส่งท้ายที่สำคัญคือ **บทที่ 10: การใช้ Plugins** ซึ่งจะกล่าวถึงความสามารถของ Tailwind CSS ในการเพิ่มขยายผ่าน plugin ทั้งแบบติดตั้งจากภายนอกและสร้างขึ้นเอง โดยเน้น plugin ยอดนิยมอย่าง @tailwindcss/forms, @tailwindcss/aspect-ratio และ @tailwindcss/line-clamp รวมถึงการใช้ API เช่น addUtilities, addComponents เพื่อ Extend Components และ Utilities ตามความต้องการของโปรเจกต์จริง

ตลอดทั้งเล่ม ผู้อ่านจะพบกับ ตัวอย่างโค้ดที่ทันสมัย เข้าใจง่าย และพร้อมใช้งาน ซึ่งเน้นการประยุกต์ใช้ในสถานการณ์จริง โดยเฉพาะในงานพัฒนาเว็บไซต์หรือเว็บแอปพลิเคชันระดับ production ที่ต้องการความเป็นระบบ ความยืดหยุ่น และความสามารถในการปรับแต่งได้สูง

หนังสือเล่มนี้เหมาะสำหรับทั้งนักพัฒนาที่ต้องการยกระดับการใช้ Tailwind CSS ให้ลึกขึ้น และผู้ที่กำลังสร้าง Design System ของตนเอง โดยไม่ต้องเริ่มจากศูนย์ Tailwind CSS ไม่ได้เป็นเพียงเครื่องมือจัดวางเลย์เอาต์เท่านั้น แต่ยังเป็นเครื่องมือที่สามารถรวมแนวคิดการออกแบบ การพัฒนาที่สะอาด และการทำงานร่วมกันของทีมให้เป็นหนึ่งเดียวได้

หวังว่าหนังสือ **Tailwind CSS Framework: Intermediate** จะเป็นคู่มือที่ช่วยให้ผู้อ่านเข้าใจการใช้ Tailwind CSS อย่างมีประสิทธิภาพ ต่อยอดความสามารถจากระดับพื้นฐานสู่ระดับมืออาชีพ และสร้างสรรค์ประสบการณ์ผู้ใช้ที่โดดเด่นในทุกอุปกรณ์และทุกบริบทของการใช้งาน.

ด้วยรักและปรารถนาดี
ศูนย์หนังสือราคาราคาเรียน

สารบัญ

หน้า

บทที่ 6 Responsive Design (Responsive Design).....	1
• Responsive Design ใน Tailwind CSS	
• Responsive Design (รายละเอียดเชิงลึก)	
• แนวคิด Mobile First Design (ใน Tailwind CSS)	
• การใช้งาน Responsive Class ใน Tailwind CSS	
• ตัวอย่าง Responsive Navbar และ Card Grid ด้วย Tailwind CSS	
บทที่ 7 Hover, Focus, Pseudo States (Hover, Focus, Pseudo States)	59
• Hover, Focus, Pseudo States	
• Hover, Focus, Pseudo States — รายละเอียดเชิงลึก	
• การใช้ hover:, focus:, active:, disabled: ใน Tailwind CSS	
• การทำ Interactive UI ด้วย Tailwind CSS	
• การใช้ group และ group-hover ใน Tailwind CSS	
• การใช้ peer และ peer-* ใน Tailwind CSS	
บทที่ 8 Typography ขั้นกลาง (Intermediate Typography)	126
• Typography	
• Typography ขั้นกลาง ของ Tailwind CSS	
• ขนาดตัวอักษรใน Tailwind CSS	
• น้ำหนักตัวอักษร (Font Weight) ใน Tailwind CSS	
• Line height (ระยะห่างบรรทัด) ด้วย leading-*	
• Letter Spacing ด้วย tracking-*	
• การใช้ plugin @tailwindcss/typography	
• ตัวอย่างบูรณาการ	
บทที่ 9 การจัดการสีและ Theme (Color and Theme)	194
• การจัดการสีและ Theme ใน Tailwind CSS	
• รายละเอียดเชิงลึกของการจัดการสีและ Theme ใน Tailwind CSS	
• การใช้สีใน Tailwind CSS	

●การสร้าง Custom Colors ใน Tailwind CSS	
●การใช้ Opacity ใน Tailwind CSS	
●สีแบบ State ใน Tailwind CSS	
●การปรับ Tone สีตาม Brand Design ใน Tailwind CSS	
●ตัวอย่างบูรณาการ	
บทที่ 10 การใช้ Plugins (Plugins)	249
●การใช้ Plugins	
●การใช้ Plugins ใน Tailwind CSS (เชิงลึก)	
●รายละเอียดเชิงลึกของ Plugin ยอดนิยมใน Tailwind CSS 3 ตัวหลัก ที่ควรรู้จัก	
●การติดตั้งและใช้งาน Plugin ใน Tailwind CSS	
●การสร้าง Tailwind CSS Plugin เองเบื้องต้น (Custom Plugin)	
●การ Extend Components และ Utilities ใน Tailwind CSS	
●ตัวอย่างบูรณาการ	
บรรณานุกรม	294

บทที่ 6

Responsive Design
(Responsive Design)

เนื้อหา

- Responsive Design ใน Tailwind CSS
- Responsive Design (รายละเอียดเชิงลึก)
- แนวคิด Mobile First Design (ใน Tailwind CSS)
- การใช้งาน Responsive Class ใน Tailwind CSS
- ตัวอย่าง Responsive Navbar และ Card Grid ด้วย Tailwind CSS

บทนำบทที่ 6: Responsive Design

ในยุคที่การเข้าถึงเว็บไซต์ผ่านอุปกรณ์พกพาอย่างสมาร์ทโฟนและแท็บเล็ตมีแนวโน้มสูงขึ้นอย่างต่อเนื่อง การออกแบบเว็บไซต์ให้สามารถปรับเปลี่ยนรูปแบบและขนาดตามอุปกรณ์ที่ผู้ใช้งานใช้จึงกลายเป็นสิ่งจำเป็นอย่างยิ่ง แนวคิดเรื่อง *Responsive Design* หรือการออกแบบที่ตอบสนองต่ออุปกรณ์หลากหลายประเภทได้ จึงเป็นหัวใจสำคัญของการพัฒนาเว็บแอปพลิเคชันยุคใหม่ โดยเฉพาะอย่างยิ่งเมื่อใช้ร่วมกับเฟรมเวิร์กอย่าง Tailwind CSS ที่ออกแบบมาเพื่อรองรับแนวคิดนี้อย่างมีประสิทธิภาพ

หนึ่งในแนวคิดหลักที่ขับเคลื่อน Responsive Design ให้เกิดขึ้นได้จริง คือ *Mobile First Design* หรือการออกแบบโดยยึดอุปกรณ์มือถือเป็นพื้นฐานก่อน แล้วค่อยขยายหรือปรับเปลี่ยนตามความสามารถของอุปกรณ์ที่มีหน้าจอใหญ่ขึ้น เช่น แท็บเล็ตหรือเดสก์ท็อป วิธีคิดแบบนี้ช่วยให้เว็บไซต์ทำงานได้อย่างราบรื่นในอุปกรณ์ขนาดเล็ก ซึ่งมักเป็นจุดเริ่มต้นของการใช้งานของผู้ใช้ส่วนใหญ่ในปัจจุบัน

Tailwind CSS สนับสนุนแนวคิด Mobile First ด้วยระบบ *Breakpoint* ที่มีมาให้ใช้งานอย่างครบถ้วน ได้แก่ sm, md, lg, xl และ 2xl ซึ่งแต่ละระดับมีหน้าที่ในการกำหนดเงื่อนไขการแสดงผลให้เหมาะสมกับขนาดหน้าจอที่แตกต่างกัน นักพัฒนาสามารถควบคุมการเปลี่ยนแปลงของ UI ได้อย่างแม่นยำโดยใช้เพียงชื่อ class prefix ที่สื่อความหมายตรงไปตรงมา

การใช้งาน *Responsive Class* ใน Tailwind CSS จึงเป็นเรื่องที่ง่ายและทรงพลัง ตัวอย่างเช่น md:flex จะเปลี่ยนองค์ประกอบให้เป็น flex container เมื่อหน้าจอมีขนาดกลางขึ้นไป หรือ lg:text-2xl จะปรับขนาดข้อความให้ใหญ่ขึ้นเมื่ออยู่บนหน้าจอขนาดใหญ่ ทั้งหมดนี้สามารถทำได้โดยไม่ต้องเขียน CSS เพิ่มเติม ซึ่งช่วยลดความซับซ้อนของโค้ดและเพิ่มความสามารถในการดูแลรักษาในระยะยาว

เพื่อให้ผู้อ่านเข้าใจและสามารถประยุกต์ใช้แนวคิด Responsive Design ได้อย่างชัดเจน บทนี้ได้นำเสนอ ตัวอย่างจริง ที่มักพบได้บ่อยในการพัฒนาเว็บ ได้แก่ *Responsive Navbar* และ *Card Grid Layout* ซึ่งเป็นโครงสร้างพื้นฐานของหลายเว็บไซต์ การเรียนรู้ผ่านตัวอย่างเหล่านี้จะช่วยให้ผู้อ่านมองเห็นภาพรวมของการปรับเปลี่ยนองค์ประกอบตามขนาดหน้าจอ และสามารถนำไปประยุกต์ใช้กับโปรเจกต์ของตนเองได้อย่างมั่นใจ

Responsive Navbar ที่นำเสนอ นั้น จะเน้นให้ผู้อ่านเห็นถึงการจัดองค์ประกอบของเมนูให้สามารถซ่อนหรือแสดงได้ตามขนาดหน้าจอ เช่น การเปลี่ยนจากเมนูแนวนอนบนเดสก์ท็อปไปสู่เมนูแบบ hamburger บนอุปกรณ์มือถือ ในขณะที่ *Card Grid Layout* จะอธิบายถึงการจัดเรียงกล่องข้อมูลให้เหมาะสมกับจำนวนคอลัมน์ที่ต่างกันในแต่ละระดับ Breakpoint

บทนี้จึงไม่ได้เพียงแค่แนะนำแนวคิด Responsive Design และการใช้ Breakpoints เท่านั้น แต่ยังเน้นการฝึกปฏิบัติจริง เพื่อให้ผู้อ่านเข้าใจกลไกของ Tailwind CSS ในการจัดการกับหน้าจอหลากหลายขนาด และสามารถสร้างเว็บไซต์ที่ดี ใช้งานได้จริง ไม่ว่าผู้ใช้งานจะเปิดด้วยอุปกรณ์ประเภทใดก็ตาม ทั้งหมดนี้คือทักษะที่จำเป็นในโลกของ Frontend Development ที่มุ่งสู่ประสบการณ์ผู้ใช้ที่ดีที่สุดในทุกบริบทการใช้งาน

Responsive Design ใน Tailwind CSS

☐ แนวคิด: Mobile First Design

Mobile First Design คือแนวคิดการออกแบบเว็บไซต์ให้เหมาะกับหน้าจอมือถือก่อน (หน้าจอเล็กที่สุด) แล้วจึงค่อย ๆ ขยายความสามารถหรือเปลี่ยน layout สำหรับหน้าจอที่ใหญ่ขึ้น (Tablet, Laptop, Desktop)

☐ ใน Tailwind CSS:

- จะใช้ class พื้นฐานเป็นค่าที่ใช้กับ mobile เป็นหลัก
- แล้วค่อยเพิ่ม **breakpoint modifier** สำหรับอุปกรณ์หน้าจอใหญ่กว่า เช่น md:, lg: เพื่อปรับ style

ตัวอย่าง:

```
<p class="text-sm md:text-lg lg:text-2xl">
```

ขนาดตัวอักษรจะใหญ่ขึ้นเมื่อหน้าจอใหญ่ขึ้น

```
</p>
```

☐ Breakpoints ที่มีให้ใน Tailwind CSS

ชื่อ Breakpoint	ความกว้างขั้นต่ำ (min-width)	ใช้กับ...
sm:	640px	โทรศัพท์แนวนอน

ชื่อ Breakpoint	ความกว้างขั้นต่ำ (min-width)	ใช้กับ...
md:	768px	Tablet ขึ้นไป
lg:	1024px	Laptop
xl:	1280px	Desktop
2xl:	1536px	หน้าจอขนาดใหญ่มาก

☐ ใช้ : ต่อท้ายเพื่อกำหนดว่า class จะทำงานเมื่อหน้าจอถึงขนาดนั้น

☐ การใช้งาน Responsive Class

Tailwind CSS ใช้รูปแบบ breakpoint:utility เพื่อกำหนดว่า utility class จะถูกใช้งานที่ breakpoint นั้น ตัวอย่างที่ 1:

```
<div class="text-center md:text-left">
```

ข้อความนี้จะอยู่ตรงกลางในมือถือ และชิดซ้ายบน tablet ขึ้นไป

```
</div>
```

ตัวอย่างที่ 2:

```
<div class="p-2 sm:p-4 md:p-6 lg:p-8">
```

Padding จะมากขึ้นตามขนาดหน้าจอ

```
</div>
```

ตัวอย่างที่ 3:

```
<div class="block md:flex lg:justify-between">
```

Layout จะเปลี่ยนจาก block → flex → justify-between ตามขนาดหน้าจอ

```
</div>
```

☐ ตัวอย่าง Responsive Navbar

☐ โค้ด:

```
<nav class="bg-gray-800 p-4">
```

```
<div class="max-w-7xl mx-auto flex flex-col sm:flex-row justify-between items-center">
```

```
<div class="text-white text-xl font-bold">MySite</div>
```

```
<div class="mt-2 sm:mt-0 flex gap-4">
```

```
<a href="#" class="text-gray-300 hover:text-white">Home</a>
```

```
<a href="#" class="text-gray-300 hover:text-white">About</a>
```

```
<a href="#" class="text-gray-300 hover:text-white">Contact</a>
```

```
</div>
```

```
</div>
```


</nav>

☐ คำอธิบาย:

- ใช้ flex-col ในมือถือ (ซ่อนแนวนอน)
- ใช้ sm:flex-row บนหน้าจอใหญ่ขึ้น (เรียงแนวนอน)
- gap-4 ใช้เว้นช่องว่างระหว่างลิงก์
- hover:text-white ทำให้สีเปลี่ยนเมื่อ hover

☐ ตัวอย่าง **Responsive Card Grid**

☐ โค้ด:

```
<div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-6 p-4">
  <div class="bg-white p-6 rounded shadow">Card 1</div>
  <div class="bg-white p-6 rounded shadow">Card 2</div>
  <div class="bg-white p-6 rounded shadow">Card 3</div>
  <div class="bg-white p-6 rounded shadow">Card 4</div>
</div>
```

☐ คำอธิบาย:

- เริ่มจาก grid-cols-1 (มือถือ: 1 คอลัมน์)
- sm:grid-cols-2: Tablet ขึ้นไปแสดง 2 คอลัมน์
- lg:grid-cols-4: Laptop ขึ้นไปแสดง 4 คอลัมน์
- gap-6: ช่องว่างระหว่าง card
- rounded + shadow: สร้างรูปลักษณะ card สวยงาม

☐ สรุป

แนวคิดหลัก	รายละเอียด
Mobile First	เขียน class ให้เหมาะกับมือถือก่อน แล้วค่อยใส่ breakpoint
Responsive Class	ใช้ breakpoint:class เช่น md:flex, lg:text-xl
Breakpoints	sm, md, lg, xl, 2xl
Layout Responsive	ใช้ flex, grid, col-span, hidden, block ร่วมกับ breakpoint
Component ตัวอย่าง	Navbar, Grid Card, Layout ปรับตามขนาดหน้าจอ

Responsive Design (รายละเอียดเชิงลึก)

☐ แนวคิดหลัก: Mobile First Design (Deep Dive)

Tailwind CSS ออกแบบตามแนวคิด **Mobile-First** ซึ่งหมายความว่า:

- ค่าที่ไม่มี **prefix** ใด ๆ (เช่น text-sm, p-4) จะถูกใช้กับอุปกรณ์ขนาดเล็กที่สุด (มือถือ)
- จากนั้นเมื่อหน้าจอขยายขึ้น Tailwind จะใช้ class ที่มี prefix เช่น sm:, md:, lg: เพื่อปรับเปลี่ยน style ตามหน้าจอ

☐ ข้อดีของ Mobile First

1. รองรับผู้ใช้งานมือถือซึ่งมีจำนวนมากในยุคปัจจุบัน
2. โหลด CSS น้อยที่สุดก่อน แล้วค่อยเพิ่มเมื่อจำเป็น
3. ทำให้การ debug responsive ง่ายขึ้น เพราะ "เริ่มจากน้อยไปมาก"

☐ Breakpoints ใน Tailwind (ขนาดที่ใช้ใน config เริ่มต้น)

// tailwind.config.js (ค่า default)

```
theme: {  
  screens: {  
    sm: '640px',  
    md: '768px',  
    lg: '1024px',  
    xl: '1280px',  
    '2xl': '1536px',  
  }  
}
```

☐ ความหมาย:

- sm: จะถูกใช้งานเมื่อ หน้าจอกว้างตั้งแต่ 640px ขึ้นไป
- ทุก breakpoint ใช้หลักการ min-width (ไม่ใช่ max)

☐ ดังนั้นคุณสามารถ “เขียนให้ mobile ก่อน” แล้วค่อย override บนหน้าจอใหญ่ขึ้นเรื่อย ๆ

☐ รูปแบบการเขียน Responsive Utility Class

รูปแบบมาตรฐาน:

<breakpoint>:<utility-class>

ตัวอย่าง:

<p class="text-sm md:text-base lg:text-lg xl:text-xl">

ปรับขนาดตัวอักษรตามขนาดหน้าจอ

</p>

Tailwind จะ generate CSS ต่อไปนี้:

```
.text-sm {  
  font-size: 0.875rem;  
}  
  
@media (min-width: 768px) {  
  .md\:text-base {  
    font-size: 1rem;  
  }  
}  
  
@media (min-width: 1024px) {  
  .lg\:text-lg {  
    font-size: 1.125rem;  
  }  
}
```

☐ เทคนิคและแนวทางปฏิบัติ (Best Practices)

1. ☐ เริ่มจาก Mobile Layout เสมอ

- เขียน layout แบบ column ด้วย flex-col หรือ grid-cols-1
- แล้วใช้ sm:flex-row, md:grid-cols-2 เพื่อเปลี่ยน layout ตามขนาด

2. ☐ ใช้ hidden, block, flex สำหรับ toggle layout

<!-- Menu ปกติซ่อน แต่แสดงเมื่อหน้าจอใหญ่ -->

<div class="hidden md:block">Main Menu</div>

3. ☐ อย่าซ่อน breakpoint โดยไม่จำเป็น

<!-- ไม่แนะนำ -->

<div class="sm:text-sm md:text-base lg:text-base xl:text-base">...</div>

<!-- แนะนำ -->

<div class="text-sm md:text-base">...</div>

☐ ตัวอย่างเจาะลึก: Responsive Navbar พร้อม Hamburger

<nav class="bg-white shadow p-4">

<div class="flex justify-between items-center max-w-7xl mx-auto">

<div class="text-lg font-bold">MySite</div>

```
<!-- ปุ่มเมนูสำหรับมือถือ -->
```

```
<button class="md:hidden block text-gray-700">
```



```
</button>
```

```
<!-- เมนูหลัก -->
```

```
<ul class="hidden md:flex space-x-6 text-gray-600">
```

```
<li><a href="#" class="hover:text-blue-500">Home</a></li>
```

```
<li><a href="#" class="hover:text-blue-500">About</a></li>
```

```
<li><a href="#" class="hover:text-blue-500">Contact</a></li>
```

```
</ul>
```

```
</div>
```

```
</nav>
```

☐ จุดสังเกต:

- ปุ่ม Hamburger block แด่บนมือถือ (md:hidden)
- เมนูหลัก hidden บนมือถือ, md:flex เมื่อหน้าจอ $\geq 768\text{px}$

☐ ตัวอย่างเจาะลึก: **Responsive Card Grid**

```
<div class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6 p-4">
```

```
<div class="bg-white p-6 rounded shadow text-center">Card 1</div>
```

```
<div class="bg-white p-6 rounded shadow text-center">Card 2</div>
```

```
<div class="bg-white p-6 rounded shadow text-center">Card 3</div>
```

```
<div class="bg-white p-6 rounded shadow text-center">Card 4</div>
```

```
</div>
```

☐ เทคนิค:

- grid-cols-1: บังคับให้แสดงเป็น 1 คอลัมน์ในมือถือ
- sm:grid-cols-2: สองคอลัมน์เมื่อหน้าจอ $\geq 640\text{px}$
- เพิ่ม breakpoints ตามลำดับเพื่อขยาย layout

☐ เพิ่มเติม: **Responsive Variants** อื่น ๆ

Tailwind รองรับ responsive class หลายกลุ่ม:

ประเภท	ตัวอย่าง
--------	----------

ประเภท	ตัวอย่าง
Layout	sm:flex, lg:grid, xl:block
Spacing	sm:p-4, md:mx-6
Typography	md:text-xl, lg:font-semibold
Visibility	hidden, md:block
Grid System	sm:grid-cols-2, lg:col-span-2
Sizing	md:w-1/2, lg:h-64

⚙️ ปรับแต่ง Breakpoints เอง (Advanced)

ใน tailwind.config.js:

```
module.exports = {
  theme: {
    screens: {
      mobile: '480px',
      tablet: '768px',
      laptop: '1024px',
      desktop: '1280px',
    }
  }
}
```

แล้วสามารถใช้ tablet:text-xl, laptop:grid-cols-3 ได้เลย

□ สรุป (Deep Summary)

หัวข้อ	อธิบาย
Mobile First Design	เขียน style สำหรับ mobile ก่อน แล้ว override ด้วย breakpoints
Responsive Class	ใช้รูปแบบ breakpoint:utility
Breakpoints	มี 5 ค่า default (sm, md, lg, xl, 2xl)
Responsive Navbar	ใช้ flex, hidden, block ร่วมกับ breakpoint
Responsive Grid	ใช้ grid-cols-* + sm:, md: เพื่อปรับ layout
Best Practices	เขียนแบบน้อย → มาก, ไม่ซ้อน class เกินจำเป็น

หัวข้อ	อธิบาย
Custom Screens	ปรับแต่ง breakpoint ใหม่ได้ใน tailwind.config.js

แนวคิด Mobile First Design (ใน Tailwind CSS)

☐ คืออะไร?

Mobile First Design คือ แนวคิดการออกแบบเว็บให้เหมาะสมกับหน้าจอมือถือก่อน (ที่เล็กที่สุด) แล้วค่อยขยายพีเจอร์หรือเปลี่ยน layout สำหรับหน้าจอที่ใหญ่ขึ้น เช่น tablet, laptop, หรือ desktop โดยใช้เทคนิคของ Media Queries (min-width) ใน CSS

☐ เหตุผลที่ต้องใช้ Mobile First

เหตุผล	คำอธิบาย
☐ ผู้ใช้งานมือถือมากที่สุด	ปัจจุบันผู้ใช้งานเว็บผ่านมือถือมีสัดส่วนสูงที่สุด
☐ โหลดเร็ว	CSS เริ่มจากขนาดเล็กสุด โหลดน้อย ใช้ทรัพยากรน้อย
☐ ปรับขยายง่าย	เริ่มจากโครงสร้างพื้นฐานที่สุด แล้วเพิ่มได้เรื่อย ๆ
☐ เขียนโค้ดง่ายและจัดการง่าย	โดยเฉพาะเมื่อใช้ร่วมกับ utility framework เช่น Tailwind CSS

☐ การนำไปใช้ใน Tailwind CSS

Tailwind CSS ถูกออกแบบมาให้ ยึดตามแนวคิด **Mobile First** โดยอัตโนมัติ

☐ หลักการทำงาน:

- class ที่ไม่มี prefix (text-sm, p-4, flex) = ใช้กับมือถือ (default)
- class ที่มี prefix (md:flex, lg:text-lg) = ใช้กับหน้าจอที่ใหญ่กว่า breakpoint นั้นขึ้นไป

☐ ตัวอย่างเปรียบเทียบ

☐ แบบ Desktop First (แนวทางเก่า):

/* เริ่มจาก Desktop แล้วค่อยแก้สำหรับ mobile */

```
.box {
  font-size: 20px;
}
```

```
@media (max-width: 768px) {
```

```
  .box {
```

```
font-size: 16px;
}
```

☐ แบบ **Mobile First (Tailwind ใช้แบบนี้)**:

```
/* เริ่มจาก mobile แล้วเพิ่มเงื่อนไขที่หน้าจอใหญ่ขึ้น */
.box {
  font-size: 16px;
}
```

```
@media (min-width: 768px) {
  .box {
    font-size: 20px;
  }
}
```

☐ ตัวอย่างการใช้ **Mobile First** ใน **Tailwind CSS**

```
<p class="text-sm md:text-base lg:text-xl">
  ข้อความนี้จะใหญ่ขึ้นเมื่อหน้าจอขยาย
</p>
```

อธิบาย:

- text-sm: ใช้กับหน้าจอ mobile (เล็กที่สุด)
- md:text-base: ใช้เมื่อหน้าจอกว้าง $\geq 768\text{px}$
- lg:text-xl: ใช้เมื่อหน้าจอกว้าง $\geq 1024\text{px}$

Tailwind จะ compile เป็น media query แบบ Mobile First:

```
.text-sm { font-size: 0.875rem; }
```

```
@media (min-width: 768px) {
  .md\:text-base { font-size: 1rem; }
}
```

```
@media (min-width: 1024px) {
  .lg\:text-xl { font-size: 1.25rem; }
}
```

⚙️ ตัวอย่าง Layout จริง

```
<div class="p-4 sm:p-6 md:p-8 lg:p-10">
  <p class="text-base sm:text-lg md:text-xl lg:text-2xl">
    Welcome to our site!
  </p>
</div>
```

❑ จากตัวอย่าง:

- mobile จะเห็น p-4, text-base
- tablet จะใช้ p-6, text-lg
- laptop จะได้ p-8, text-xl
- desktop ได้ p-10, text-2xl

❑ ประโยชน์ที่แท้จริงของ Mobile First Design

ประโยชน์	รายละเอียด
ประสบการณ์ผู้ใช้ดี	เว็บไซต์โหลดไวและเข้าถึงง่ายจากมือถือ
ปรับแต่ง UI ได้ง่าย	เมื่อทำงานกับ Tailwind, การเพิ่ม breakpoints ทำให้ควบคุม layout ได้ง่าย
รองรับอุปกรณ์หลากหลาย	ออกแบบครั้งเดียว รองรับทั้งมือถือ, tablet, desktop
ลด CSS ที่ซ้ำซ้อน	ใช้ class utility เพียงชุดเดียวสำหรับทุกอุปกรณ์

❑ แนวทางปฏิบัติ (Best Practices)

- ❑ เริ่มจาก **Mobile Layout** ก่อน: อย่าใช้ lg: ก่อน sm: หรือ md:
- ❑ ใช้ hidden, block, flex กับ breakpoint เพื่อปรับ visibility
- ❑ ทดสอบ UI บนหน้าจอเล็กก่อน แล้วค่อยขยายไปสู่หน้าจอใหญ่

❑ กลุ่มที่ 1: ตัวอย่างโปรแกรมพื้นฐาน (3 ตัวอย่าง)

แนวคิด Mobile First Design + Tailwind CSS

ไฟล์เดี่ยว (HTML) พร้อมคำอธิบายโค้ดและผลลัพธ์

❑ ตัวอย่างที่ 1: Responsive Text Section

❑ โครงสร้าง

project-1/

└── index.html

❑ index.html


```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Responsive Text</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="p-6">
  <h1 class="text-lg sm:text-xl md:text-2xl lg:text-4xl font-bold text-blue-600">
    Welcome to Mobile First Design
  </h1>
  <p class="mt-2 text-sm sm:text-base md:text-lg">
    This text scales depending on screen size.
  </p>
</body>
</html>

```

☐ คำอธิบาย

- text-lg คือค่าพื้นฐาน (มือถือ)
- เพิ่มขนาด font ที่ละขั้นผ่าน sm:, md:, lg:
- viewport tag ช่วยให้ Responsive จริง

☐ ผลการรัน

- มือถือ: ข้อความขนาดเล็ก
- Desktop: ขนาดใหญ่ขึ้นตาม breakpoint

☐ ตัวอย่างที่ 2: Responsive Navigation Menu

☐ โครงสร้าง

project-2/

└── index.html

☐ index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Responsive Navbar</title>
<script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <nav class="bg-gray-800 p-4 text-white flex justify-between items-center">
    <div class="text-xl font-semibold">Brand</div>
    <div class="hidden md:flex space-x-4">
      <a href="#">Home</a>
      <a href="#">Services</a>
      <a href="#">Contact</a>
    </div>
    <div class="block md:hidden">☰</div>
  </nav>
</body>
</html>

```

☐ คำอธิบาย

- เมนูถูกซ่อนบนมือถือ (hidden md:flex)
- ปุ่ม hamburger แสดงเฉพาะมือถือ (block md:hidden)
- ใช้ justify-between และ space-x-4 จัด layout

☐ ผลการรัน

- มือถือ: แสดงปุ่ม ☰
- Desktop: แสดงเมนูแนวนอนเต็มรูปแบบ

☐ ตัวอย่างที่ 3: Responsive Grid Layout

☐ โครงสร้าง

project-3/

├── index.html

☐ index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

```

```

<title>Responsive Grid</title>
<script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="p-6 bg-gray-50">
  <div class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 gap-4">
    <div class="bg-white p-4 shadow">Card 1</div>
    <div class="bg-white p-4 shadow">Card 2</div>
    <div class="bg-white p-4 shadow">Card 3</div>
  </div>
</body>
</html>

```

☐ คำอธิบาย

- Layout เริ่มจาก 1 คอลัมน์
- Breakpoints: 2 คอลัมน์ (tablet), 3 คอลัมน์ (desktop)
- ใช้ grid-cols-* + gap-4 สร้าง grid ที่ flexible

☐ ผลการรัน

- Mobile: 1 card ต่อบรรทัด
- Tablet: 2 cards
- Desktop: 3 cards

☐ กลุ่มที่ 2: ตัวอย่างแนวประยุกต์ — แนวคิด Mobile First Design

☐ โปรแกรมที่ 1: Responsive Landing Page Hero Section

☐ โครงสร้าง

hero-landing/

└── index.html

☐ index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Hero Section</title>
  <script src="https://cdn.tailwindcss.com"></script>

```

```

</head>
<body class="bg-white">
  <section class="p-6 md:p-12 text-center md:text-left max-w-5xl mx-auto">
    <h1 class="text-3xl sm:text-4xl md:text-5xl font-bold text-gray-900">
      Build Your Idea <br class="hidden md:block" />
      <span class="text-blue-600">Faster than Ever</span>
    </h1>
    <p class="mt-4 text-gray-600 text-base sm:text-lg md:text-xl">
      Create responsive web experiences with Tailwind CSS and mobile-first design.
    </p>
    <div class="mt-6">
      <button class="px-6 py-2 bg-blue-600 text-white rounded hover:bg-blue-700">Get
Started</button>
    </div>
  </section>
</body>
</html>

```

☐ คำอธิบาย

- text-center บนมือถือ → md:text-left สำหรับ desktop
- text-3xl → sm:4xl → md:5xl: ขนาดหัวเรื่องที่ responsive
- ปุ่มจะอยู่ใต้เนื้อหาแบบแนวนอนบนมือถือ และจัดระยะห่างดีบน desktop

☐ ผลลัพธ์

- Mobile: layout ตรงกลาง จัดเรียงแนวนอน
- Desktop: layout ชิดซ้าย ขนาดหัวเรื่องใหญ่ขึ้น

☐ โปรแกรมที่ 2: Responsive Resume/Portfolio Section

☐ โครงสร้าง

resume-section/

|— index.html

☐ index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8" />
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Responsive Resume</title>
<script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100 p-6">
  <section class="bg-white p-6 rounded shadow max-w-3xl mx-auto">
    <div class="md:flex md:items-center md:space-x-6">
      
      <div class="text-center md:text-left mt-4 md:mt-0">
        <h2 class="text-xl font-bold text-gray-900">Jane Doe</h2>
        <p class="text-gray-600">Frontend Developer</p>
        <p class="text-sm mt-2 text-gray-500">Passionate about building accessible and
performant interfaces.</p>
      </div>
    </div>
  </section>
</body>
</html>

```

☐ คำอธิบาย

- md:flex สำหรับ layout แนวนอน
- mx-auto → จัดกลางใน mobile, md:mx-0 จัดซ้ายใน desktop
- text-center → md:text-left: เปลี่ยนการจัดข้อความ

☐ ผลลัพธ์

- Mobile: รูปภาพอยู่ด้านบน ชื่ออยู่ด้านล่าง
- Desktop: รูปภาพซ้าย ข้อความขวา เป็น row layout

☐ โปรแกรมที่ 3: Responsive Pricing Cards

☐ โครงสร้าง

pricing-cards/

|—— index.html

☐ index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Responsive Pricing</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-50 p-6">
  <section class="max-w-6xl mx-auto grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
    <div class="bg-white p-6 rounded shadow text-center">
      <h3 class="text-xl font-bold">Basic</h3>
      <p class="text-gray-600">$9/month</p>
      <ul class="mt-4 text-sm text-gray-500 space-y-1">
        <li>✓ 1 Project</li>
        <li>✓ Email Support</li>
        <li>✓ Community Access</li>
      </ul>
    </div>

    <div class="bg-blue-100 p-6 rounded shadow text-center">
      <h3 class="text-xl font-bold">Pro</h3>
      <p class="text-gray-600">$29/month</p>
      <ul class="mt-4 text-sm text-gray-700 space-y-1">
        <li>✓ 5 Projects</li>
        <li>✓ Priority Support</li>
        <li>✓ Team Access</li>
      </ul>
    </div>

    <div class="bg-white p-6 rounded shadow text-center">
      <h3 class="text-xl font-bold">Enterprise</h3>
      <p class="text-gray-600">Contact Us</p>
      <ul class="mt-4 text-sm text-gray-500 space-y-1">
        <li>✓ Unlimited Projects</li>
      </ul>
    </div>
  </section>
```

```

</li> ✓ Dedicated Manager</li>
</li> ✓ SLA Guarantee</li>
</ul>
</div>
</section>
</body>
</html>

```

☐ คำอธิบาย

- grid-cols-1: เริ่มต้นด้วย layout 1 card ต่อแถว
- sm:grid-cols-2: 2 cards บน tablet
- lg:grid-cols-3: 3 cards บน desktop
- การใช้พื้นหลังต่างสีช่วยเน้น plan กลาง (Pro)

☐ ผลลัพธ์

- Mobile: เรียงแนวตั้ง
- Tablet: 2 คอลัมน์
- Desktop: 3 คอลัมน์ layout สวยงาม

☐ สรุปกลุ่มแนวประยุกต์

โปรแกรม	แนวคิดหลัก
Hero Section	Responsive typography และ alignment
Resume Card	Flex layout ที่เปลี่ยนจาก column → row
Pricing Cards	Grid layout + scaling structure

☐ กลุ่มที่ 3: ตัวอย่างบูรณาการ – Mobile First + JavaScript + Interactive

☐ โปรแกรมที่ 1: Responsive Mobile Navbar + Toggle Menu (JS)

☐ โครงสร้าง

mobile-navbar/

```

├── index.html
├── script.js

```

☐ index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Responsive Navbar</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-white">
  <nav class="bg-blue-600 p-4 text-white flex justify-between items-center">
    <div class="text-xl font-bold">Brand</div>
    <button id="menu-btn" class="block md:hidden text-2xl focus:outline-none">☰</button>
    <ul id="menu" class="hidden md:flex space-x-4 text-sm">
      <li><a href="#">Home</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>

  <script src="script.js"></script>
</body>
</html>

```

☐ script.js

```

const btn = document.getElementById('menu-btn');
const menu = document.getElementById('menu');

```

```

btn.addEventListener('click', () => {
  menu.classList.toggle('hidden');
  menu.classList.toggle('flex');
  menu.classList.toggle('flex-col');
  menu.classList.toggle('mt-4');
});

```

☐ ผลลัพธ์

- Mobile: มีปุ่ม ☰ กดแล้วแสดงเมนู
- Desktop: เมนูแสดงแบบปกติแนวนอน

☐ โปรแกรมที่ 2: Responsive Accordion Section (FAQ)☐ โครงสร้าง

faq-accordion/

├── index.html

└── script.js

☐ index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Responsive FAQ</title>

<script src="https://cdn.tailwindcss.com"></script>

</head>

<body class="p-6 bg-gray-100">

<div class="max-w-xl mx-auto">

<h1 class="text-2xl font-bold mb-4 text-center sm:text-left">FAQs</h1>

<div class="space-y-4">

<div class="bg-white p-4 rounded shadow">

<button class="w-full text-left font-semibold faq-toggle">What is Tailwind?</button>

<div class="faq-content hidden mt-2 text-gray-600">

Tailwind is a utility-first CSS framework for rapidly building UI.

</div>

</div>

<div class="bg-white p-4 rounded shadow">

<button class="w-full text-left font-semibold faq-toggle">What is Mobile First?</button>

<div class="faq-content hidden mt-2 text-gray-600">

Mobile First means designing for small screens first, then expanding.

</div>

</div>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

☐ **script.js**

```
document.querySelectorAll('.faq-toggle').forEach((btn) => {
  btn.addEventListener('click', () => {
    const content = btn.nextElementSibling;
    content.classList.toggle('hidden');
  });
});
```

☐ **ผลลัพธ์**

- Mobile/Tablet/Desktop: ปรับขนาดอัตโนมัติ
- กดคำถามแต่ละข้อเพื่อเปิด-ปิดคำตอบ

☐ **โปรแกรมที่ 3: Responsive Form Validation (JS Dynamic Feedback)**
☐ **โครงสร้าง**

responsive-form/

```
|
|_____ index.html
|_____ script.js
```

☐ **index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Form Validation</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-50 p-6">
  <form id="contact-form" class="bg-white p-6 rounded shadow max-w-lg mx-auto space-y-4">
    <div>
      <label class="block font-medium">Email</label>
      <input type="email" id="email" class="w-full border border-gray-300 p-2 rounded mt-1"
required />
      <p id="email-error" class="text-red-600 text-sm mt-1 hidden">Invalid email address</p>
    </div>
```

```

<button type="submit" class="bg-blue-600 text-white px-4 py-2 rounded hover:bg-blue-700">
  Submit
</button>

</form>

<script src="script.js"></script>
</body>
</html>

```

❑ script.js

```

const form = document.getElementById('contact-form');
const email = document.getElementById('email');
const error = document.getElementById('email-error');

```

```

form.addEventListener('submit', (e) => {
  e.preventDefault();
  if (!email.value.includes('@')) {
    error.classList.remove('hidden');
  } else {
    error.classList.add('hidden');
    alert("Submitted!");
  }
});

```

❑ ผลลัพธ์

- Mobile: ฟอร์มแสดงแบบเรียง column
- Desktop: layout กว้างขึ้นอัตโนมัติ
- เมื่อกรอก email ผิด จะมีข้อความแจ้งเตือนแบบ dynamic

❑ สรุปกลุ่มบูรณาการ

โปรแกรม	สิ่งที่บูรณาการ
Navbar Toggle	Mobile First + JavaScript toggle menu
FAQ Accordion	Responsive + JS interactivity
Form Validation	Mobile First + Dynamic Validation Feedback

☐ รายการ Breakpoints ที่ Tailwind ให้มา (ค่าเริ่มต้น)

ชื่อ	ความกว้างขั้นต่ำ (min-width)	คำอธิบาย
sm	640px	สำหรับ โทรศัพท์จอใหญ่ หรือ Tablet แนวตั้ง
md	768px	สำหรับ Tablet แนวนอน หรือ หน้าจอเล็ก
lg	1024px	สำหรับ หน้าจอแล็ปท็อปขนาดกลาง
xl	1280px	สำหรับ หน้าจอเดสก์ท็อปใหญ่
2xl	1536px	สำหรับ หน้าจอ ultra-wide หรือ 4K

☐ แนวคิดสำคัญ: Tailwind ใช้แนวทาง min-width (Mobile First)

หมายความว่า styles จะใช้ ตั้งแต่ breakpoint ขึ้นไป

☐ วิธีใช้งาน Breakpoint ใน Class

คุณสามารถเขียน class แบบนี้:

```
<p class="text-base md:text-lg lg:text-xl">ข้อความ</p>
```

☐ ความหมาย:

- text-base: ใช้กับจอมือถือ (ต่ำกว่า md)
- md:text-lg: ตั้งแต่ 768px ขึ้นไปใช้ text-lg
- lg:text-xl: ตั้งแต่ 1024px ขึ้นไปใช้ text-xl

☐ ตัวอย่างการใช้งาน Breakpoints

```
<div class="bg-gray-200 p-4 sm:bg-blue-100 md:bg-green-100 lg:bg-yellow-100 xl:bg-red-100 2xl:bg-purple-100">
```

สวัสดี Tailwind!

```
</div>
```

คำอธิบาย

- มือถือเล็ก: bg-gray-200
- >= 640px: bg-blue-100
- >= 768px: bg-green-100
- >= 1024px: bg-yellow-100
- >= 1280px: bg-red-100
- >= 1536px: bg-purple-100

☐ การปรับแต่ง Breakpoints

คุณสามารถแก้ไขหรือเพิ่ม breakpoints ได้ในไฟล์ `tailwind.config.js`:

```
module.exports = {
  theme: {
    extend: {},
    screens: {
      'sm': '640px',
      'md': '768px',
      'lg': '1024px',
      'xl': '1280px',
      '2xl': '1536px',
      '3xl': '1920px' // เพิ่มเอง
    }
  }
}
```

□ สรุป

- Tailwind มี breakpoints 5 ระดับหลัก: sm, md, lg, xl, 2xl
- ใช้งานโดยเติม prefix หน้า class เช่น md:flex, lg:text-xl
- แนวทาง Mobile First: ทุก class จะทำงานกับมือถือก่อน แล้วค่อย override ด้วย breakpoint ถัดไป
- ปรับแต่งได้เองใน config

กลุ่มที่ 1: ตัวอย่างโปรแกรมพื้นฐาน — Breakpoints (sm, md, lg, xl, 2xl)

โปรแกรมที่ 1: Responsive Text Size Page

โครงสร้างไฟล์

```
responsive-text/
└── index.html
```

index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8" />
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1" />
```

```

<title>Responsive Text Size</title>
<script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="p-6">
  <h1 class="text-base sm:text-lg md:text-xl lg:text-2xl xl:text-3xl 2xl:text-4xl font-bold text-center">
    Responsive Text Size with Tailwind Breakpoints
  </h1>
  <p class="mt-4 text-sm sm:text-base md:text-lg lg:text-xl xl:text-2xl 2xl:text-3xl text-gray-700 text-center">
    This paragraph text size changes based on screen size using Tailwind's breakpoints.
  </p>
</body>
</html>

```

คำอธิบาย

- text-base เป็นขนาดเริ่มต้นสำหรับมือถือ (ไม่มี prefix)
- ขยายขนาดตาม breakpoint: sm:text-lg ($\geq 640\text{px}$), md:text-xl ($\geq 768\text{px}$), lg:text-2xl ($\geq 1024\text{px}$), xl:text-3xl ($\geq 1280\text{px}$), 2xl:text-4xl ($\geq 1536\text{px}$)
- เน้นการใช้ Mobile First Design

โปรแกรมที่ 2: Responsive Grid Layout

โครงสร้างไฟล์

```

responsive-grid/
└── index.html

```

index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Responsive Grid</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="p-6 bg-gray-50">

```