

ปฏิบัติการที่ 3: CSS Fundamentals

วัตถุประสงค์

1. เรียนรู้วิธีใช้ CSS ทั้ง 3 วิธี (Inline, Internal, External)
2. ฝึกใช้ CSS Selectors (Universal, Element, Class, ID, Attribute, Pseudo-class)
3. ออกแบบสี และจัดรูปแบบข้อความ (Colors & Text)
4. ทำความเข้าใจ Box Model (Content, Padding, Border, Margin)
5. ฝึก Positioning (Static, Relative, Absolute, Fixed, Sticky)
6. เรียนรู้ Display property สำหรับ Layout
7. บันทึกร่วมกับ Git ด้วยการให้ meaningful commits

เตรียม Git Repository

1. สร้างโฟลเดอร์ใหม่ wk03-lab

```
mkdir wk03-lab  
cd wk03-lab
```

```
# เปิด vs code  
code .
```

2. Initialize Git
git init

3. ตั้งค่า Git User (ครั้งแรก)

```
git config user.name "Your Name"  
git config user.email "your@email.com"
```

4. สร้างไฟล์ .gitignore

```
node_modules/  
.DS_Store  
*.log
```

5. Commit ครั้งแรก

```
git add .gitignore  
git commit -m "initial: add .gitignore"
```

Git Workflow ที่ใช้ในปฏิบัติการ

1. Commit ทุกขั้นตอนที่สำคัญ

```
git add  
git commit -m "commit messages"
```

Best Practices Commit Messages → <https://angsilacs.buu.ac.th/~wittawas/682/88726065/GIT-USECASE.html#best-practices>

2. Push ไปที่ remote repository เมื่อทำปฏิบัติการเสร็จแล้ว
-

เริ่มปฏิบัติการ

ตอนที่ 1: สร้างหน้าแรก

1. สร้างไฟล์ index.html ดังนี้

```
<!DOCTYPE html>
<html lang="th">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Week 3: CSS Fundamentals</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <div class="container">
      <h1>Week 3: CSS Fundamentals</h1>
      <p class="intro">ยินดีต้อนรับสู่ปฏิบัติการ CSS ของสัปดาห์ที่ 3</p>

      <section id="main-section">
        <h2>ตัวอย่างหน้าเพจ</h2>
        <p>
          หน้านี้แสดงตัวอย่างการใช้ CSS Selectors ทั้งหมดที่คุณจะได้เรียนรู้
        </p>

        <h3>ให้เข้าไปดู HTML files ต่อไปนี้:</h3>
        <ul>
          <li>
            <a href="units.html" class="btn">CSS Units - หน่วยวัด</a>
          </li>
          <li>
            <a href="box-model.html" class="btn">Box Model & Colors & Text</a>
          </li>
          <li><a href="positioning.html" class="btn">CSS Positioning</a></li>
          <li>
            <a href="display-layout.html" class="btn">Display & Layout</a>
          </li>
        </ul>
      </section>
    </div>
  </body>
</html>
```

2. ดูผลลัพธ์หน้าเว็บผ่าน Live Server

Week 3: CSS Fundamentals

ยินดีต้อนรับสู่ปฏิบัติการ CSS ของสัปดาห์ที่ 3

ตัวอย่างหน้าเพจ

หน้านี้แสดงตัวอย่างการใช้ CSS Selectors ทั้งหมดที่คุณจะได้เรียนรู้

ให้เข้าไปดู HTML files ต่อไปนี้:

- [CSS Units - หน่วยวัด](#)
- [Box Model & Colors & Text](#)
- [CSS Positioning](#)
- [Display & Layout](#)

3. commit "Initial commit: add index.html"

ตอนที่ 2: External Style Sheet

1. สร้างไฟล์ styles.css

```
/* =====  
UNIVERSAL SELECTOR (*)  
===== */  
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}  
  
/* =====  
ELEMENT SELECTORS  
===== */  
  
/* Body - ตั้งค่า default ทั้งหมด */  
body {  
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;  
  line-height: 1.6;  
  color: #333;  
  background-color: #f5f5f5;  
}  
  
/* Heading */  
h1 {  
  font-size: 2.5em;  
  font-weight: 700;  
  color: #264de4;  
  text-align: center;  
  line-height: 1.2;  
  margin-bottom: 20px;  
}  
  
h2 {  
  font-size: 2em;  
  color: #3498db;  
  margin-top: 30px;  
  margin-bottom: 15px;  
  border-bottom: 2px solid #3498db;  
  padding-bottom: 10px;  
}  
  
h3 {  
  font-size: 1.5em;  
  color: #555;  
  margin-top: 20px;  
  margin-bottom: 10px;  
}  
  
/* Paragraph */  
p {  
  font-size: 1em;  
  margin-bottom: 15px;  
}  
  
/* Container */  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 40px 20px;  
  background: white;  
  border-radius: 8px;  
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  
}
```

```
/* =====  
CLASS SELECTORS (.)  
===== */  
  
/* Intro paragraph */  
.intro {  
  font-size: 1.2em;  
  color: #666;  
  text-align: center;  
  font-style: italic;  
  margin-bottom: 30px;  
}  
  
/* Highlight text */  
.highlight {  
  background-color: #fff3cd;  
  padding: 2px 6px;  
  border-radius: 3px;  
  color: #856404;  
}  
  
/* Code block styling */  
.code-box {  
  background-color: #f4f4f4;  
  border-left: 4px solid #264de4;  
  padding: 15px;  
  margin: 20px 0;  
  border-radius: 4px;  
  overflow-x: auto;  
}  
  
.code-box code {  
  font-family: "Courier New", monospace;  
  font-size: 0.9em;  
  color: #d63031;  
}  
  
/* Button styling */  
.btn {  
  display: inline-block;  
  padding: 10px 20px;  
  background-color: #dddddd;  
  color: rgb(0, 0, 0);  
  text-decoration: none;  
  border-radius: 5px;  
  transition: all 0.3s ease;  
  border: none;  
  cursor: pointer;  
  font-size: 1em;  
  margin: 10px 5px 10px 0;  
}  
  
.btn:hover {  
  background-color: #c8ff00;  
  transform: translateY(-2px);  
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);  
}  
  
.btn:active {  
  transform: translateY(0);  
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);  
}
```

```

/* =====
ID SELECTORS (#)
===== */
#main-section {
  margin: 40px 0;
}

/* =====
ATTRIBUTE SELECTORS []
===== */
input[type="text"],
input[type="email"],
textarea {
  width: 100%;
  padding: 10px;
  margin: 10px 0;
  border: 1px solid #ddd;
  border-radius: 4px;
  font-family: inherit;
}

input[type="text"]:focus,
input[type="email"]:focus,
textarea:focus {
  outline: none;
  border-color: #264de4;
  box-shadow: 0 0 5px rgba(38, 77, 228, 0.5);
}

input[disabled] {
  background-color: #e9ecef;
  cursor: not-allowed;
}

a[target="_blank"]::after {
  content: " ↗";
  font-size: 0.8em;
}

/* =====
PSEUDO-CLASS SELECTORS
===== */
a {
  color: #264de4;
  text-decoration: none;
  transition: color 0.3s ease;
}

a:hover {
  color: #3498db;
  text-decoration: underline;
}

a:visited {
  color: #8e44ad;
}

a:active {
  color: #c0392b;
}

li:first-child {
  font-weight: bold;
}

```

```
li:last-child {
  margin-bottom: 0;
}

/* =====
COLOR & TEXT STYLING
===== */

/* Text colors */
.text-primary {
  color: #264de4;
}

.text-success {
  color: #27ae60;
}

.text-warning {
  color: #f39c12;
}

.text-danger {
  color: #e74c3c;
}

/* Background colors */
.bg-light {
  background-color: #ecf0f1;
  padding: 15px;
  border-radius: 4px;
}

.bg-info {
  background-color: #d1ecf1;
  border-left: 4px solid #0c5460;
  padding: 15px;
  border-radius: 4px;
}

/* Font styling */
.font-bold {
  font-weight: 700;
}

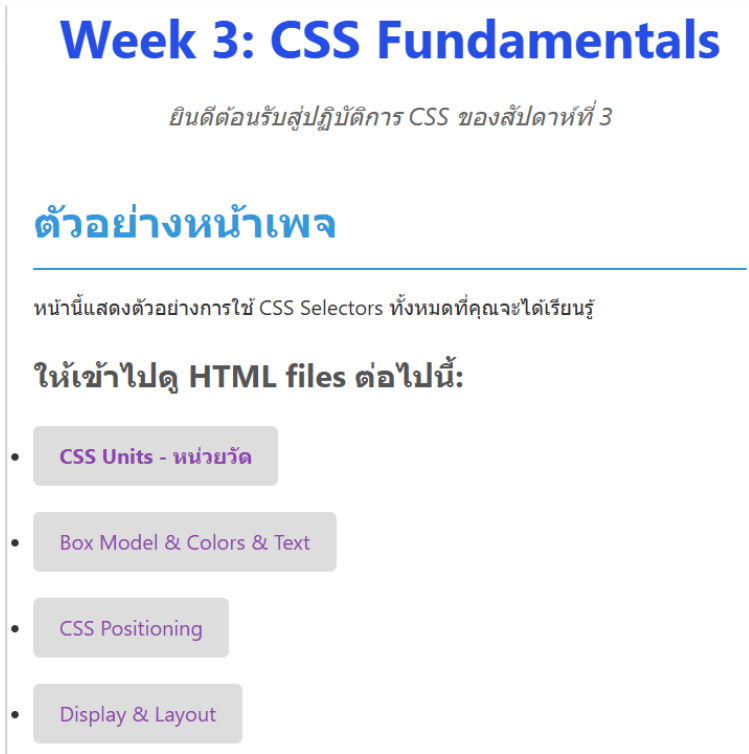
.font-italic {
  font-style: italic;
}

.text-uppercase {
  text-transform: uppercase;
  letter-spacing: 1px;
}

.text-center {
  text-align: center;
}

.text-justify {
  text-align: justify;
}
```

2. ดูผลลัพธ์หน้าเว็บผ่าน Live Server



3. commit "add: Add external CSS with selectors styling"

ตอนที่ 3 : CSS Units - หน่วยวัด

1. สร้างไฟล์ units.html

```
<!DOCTYPE html>
<html lang="th">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CSS Units - หน่วยวัด</title>
    <link rel="stylesheet" href="styles.css" />
    <style>
      /* =====
      CSS UNITS DEMONSTRATION
      ===== */

      .units-section {
        margin: 40px 0;
      }

      /* Pixel (px) - Fixed Unit */
      .px-demo {
        width: 300px;
        padding: 20px;
        margin: 10px;
        background-color: #e8f4f8;
        border: 2px solid #264de4;
        border-radius: 5px;
      }
    </style>
  </head>
  <body>
    <div class="units-section">
      <div class="px-demo">
        <h3>CSS Units - หน่วยวัด</h3>
      </div>
    </div>
  </body>
</html>
```

```
/* Percentage (%) - Relative to Parent */
.percentage-container {
  width: 400px;
  background-color: #f0f0f0;
  border: 2px solid #666;
  margin: 20px 0;
}

.percentage-demo {
  width: 100%;
  padding: 20px;
  background-color: #d5f4e6;
  border: 2px solid #27ae60;
  margin: 10px 0;
}

.percentage-demo-50 {
  width: 50%;
  padding: 20px;
  background-color: #fef5e7;
  border: 2px solid #f39c12;
  margin: 10px 0;
}

.percentage-demo-75 {
  width: 75%;
  padding: 20px;
  background-color: #fadbd8;
  border: 2px solid #e74c3c;
  margin: 10px 0;
}

/* Em - Relative to Font Size */
.em-demo {
  font-size: 16px;
  background-color: #d1ecf1;
  border: 2px solid #0c5460;
  padding: 20px;
  margin: 20px 0;
}

.em-demo-small {
  font-size: 0.8em;
  color: #666;
}

.em-demo-large {
  font-size: 1.5em;
  color: #264de4;
  font-weight: bold;
}

.em-demo-margin {
  margin: 2em 0;
  padding: 1em;
  background-color: #f4ecf7;
  border: 2px solid #8e44ad;
}
```

```
/* Rem - Relative to Root Font Size */
.rem-demo {
  background-color: #ecf0f1;
  border: 2px solid #95a5a6;
  padding: 1rem;
  margin: 1rem 0;
}

.rem-demo-h {
  font-size: 2rem;
  color: #264de4;
  margin-bottom: 0.5rem;
}

.rem-demo-p {
  font-size: 1rem;
  line-height: 1.5rem;
  color: #333;
}

/* Viewport Units */
.viewport-demo {
  height: 50vh;
  background: linear-gradient(135deg, #3498db, #2ecc71);
  color: white;
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 20px 0;
  border-radius: 8px;
  font-size: 2rem;
  font-weight: bold;
}

.viewport-demo-small {
  height: 25vh;
  background: linear-gradient(135deg, #e74c3c, #f39c12);
  color: white;
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 20px 0;
  border-radius: 8px;
  font-weight: bold;
}

/* Comparison Box */
.units-comparison {
  display: flex;
  gap: 20px;
  flex-wrap: wrap;
  margin: 20px 0;
}

.unit-box {
  flex: 1;
  min-width: 200px;
  padding: 20px;
  border-radius: 8px;
}
```

```
    text-align: center;
    color: white;
    font-weight: bold;
}

.unit-box.px {
    background-color: #264de4;
    width: 200px;
}

.unit-box.em {
    background-color: #3498db;
    font-size: 1.2em;
}

.unit-box.rem {
    background-color: #27ae60;
    font-size: 1.2rem;
}

.unit-box.percent {
    background-color: #f39c12;
    width: 80%;
}

.unit-box.vw {
    background-color: #e74c3c;
    width: 60vw;
}

/* Units Characteristics Table */
.units-table {
    width: 100%;
    border-collapse: collapse;
    margin: 20px 0;
}

.units-table th,
.units-table td {
    padding: 15px;
    text-align: left;
    border: 1px solid #ddd;
}

.units-table th {
    background-color: #264de4;
    color: white;
    font-weight: bold;
}

.units-table tr:nth-child(even) {
    background-color: #f9f9f9;
}

.units-table tr:hover {
    background-color: #f0f0f0;
}
```

```
/* Use Cases */
.use-case {
  background-color: #d5f4e6;
  border-left: 4px solid #27ae60;
  padding: 15px;
  margin: 15px 0;
  border-radius: 4px;
}

.use-case strong {
  color: #27ae60;
}

/* Responsive Box Example */
.responsive-demo {
  display: flex;
  gap: 20px;
  flex-wrap: wrap;
  margin: 20px 0;
}

.responsive-item {
  flex: 1;
  min-width: 150px;
  padding: 20px;
  background-color: #d1ecf1;
  border: 2px solid #0c5460;
  border-radius: 4px;
  text-align: center;
}

.responsive-item-fixed {
  flex: 0 0 auto;
  width: 200px;
  padding: 20px;
  background-color: #fadbd8;
  border: 2px solid #e74c3c;
  border-radius: 4px;
  text-align: center;
}

/* Code Example */
.code-example {
  background-color: #f4f4f4;
  border-left: 4px solid #264de4;
  padding: 15px;
  margin: 20px 0;
  border-radius: 4px;
  overflow-x: auto;
}

.code-example code {
  font-family: "Courier New", monospace;
  font-size: 0.9em;
  color: #d63031;
}
</style>
</head>
<body>
```

```

<div class="container">
  <h1>CSS Units - หน่วยวัด ใน CSS</h1>
  <p class="intro">
    เรียนรู้หน่วยวัดต่าง ๆ ที่ใช้ใน CSS และเมื่อไหร่ที่ควรใช้
  </p>

  <!-- Introduction -->
  <section class="units-section">
    <h2>บทนำ CSS Units</h2>
    <p>
      CSS มีหน่วยวัดมากมาย แต่ละหน่วยมีข้อดีและข้อเสียที่แตกต่างกัน
      การเลือกหน่วยที่ถูกต้องจะช่วยให้เว็บไซต์ของคุณสะดวกใช้งานและตอบสนองต่อขนาดจอได้ดี
    </p>
  </section>

  <!-- Absolute Units -->
  <section class="units-section">
    <h2>1. Absolute Units (หน่วยคงที่)</h2>
    <p>หน่วยที่มีขนาดคงที่ ไม่ขึ้นกับค่าอื่น</p>

    <h3>Pixel (px) - หน่วยพิกเซล</h3>
    <p>หน่วยที่นิยมใช้มากที่สุด 1px = 1/96 inch</p>
    <div class="px-demo">
      <h4>ตัวอย่าง: width: 300px, padding: 20px</h4>
      <p>Element นี้มีความกว้างคงที่ 300px</p>
      <p>ขนาดจะไม่เปลี่ยน ไม่ว่าหน้าจอมีขนาดเท่าไร</p>
    </div>

    <div class="code-example">
      <code>
        >/* Pixel Unit */<br />
        .box {<br />
          &nbsp;&nbsp;&nbsp;width: 300px;<br />
          &nbsp;&nbsp;&nbsp;padding: 20px;<br />
          &nbsp;&nbsp;&nbsp;margin: 10px;<br />
          &nbsp;&nbsp;&nbsp;font-size: 16px;<br />
        }
      </code>
    </div>

    <div class="use-case">
      <strong>☑ ใช้ px เมื่อ:</strong>
      <ul>
        <li>ต้องการความแม่นยำสูง (border, shadows)</li>
        <li>ใช้ fixed layouts</li>
        <li>ออกแบบ icons หรือ small elements</li>
      </ul>
    </div>
  </section>

  <!-- Relative Units -->
  <section class="units-section">
    <h2>2. Relative Units (หน่วยสัมพันธ์)</h2>
    <p>หน่วยที่ขึ้นกับค่าอื่น (parent element, root element, viewport)</p>

  <!-- Percentage -->

```

<h3>Percentage (%) - เปอร์เซ็นต์</h3>

<p>ขนาดสัมพันธ์กับ parent element (ค่าเริ่มต้นคือ 100%)</p>

```
<div class="percentage-container">
  <p style="color: #666; margin: 10px 0">
    Parent Container: width: 400px
  </p>
```

```
<div class="percentage-demo">
  <p>width: 100% = 400px</p>
</div>
```

```
<div class="percentage-demo-50">
  <p>width: 50% = 200px</p>
</div>
```

```
<div class="percentage-demo-75">
  <p>width: 75% = 300px</p>
</div>
```

```
<div class="code-example">
  <code>
    >/* Percentage Unit */<br />
    .container { width: 400px; }<br />
    .full { width: 100%; } /* = 400px */<br />
    .half { width: 50%; } /* = 200px */<br />
    .three-quarters { width: 75%; } /* = 300px */
  </code>
</div>
```

```
<div class="use-case">
  <strong><input checked="" type="checkbox"/> ใช้ % เมื่อ:</strong>
  <ul>
    <li>ต้องการ responsive layout</li>
    <li>ต้องการให้ element ปรับขนาดตามที่มี</li>
    <li>ทำ column layouts (น้ำหนักสัมพันธ์)</li>
  </ul>
</div>
```

<!-- Em Unit -->

<h3>Em (em) - เทียบกับ Font Size ปัจจุบัน</h3>

```
<p>
  1em = current font-size; ถ้า parent มี font-size 16px แล้ว 1em = 16px
</p>
```

```
<div class="em-demo">
  <p style="margin-bottom: 20px">
    <strong>Parent: font-size: 16px</strong>
  </p>
```

```
<p class="em-demo-small">Small text (font-size: 0.8em = 12.8px)</p>
```

```
<p style="font-size: 1em; margin: 10px 0">
  Normal text (font-size: 1em = 16px)
</p>
```

```
<p class="em-demo-large">Large text (font-size: 1.5em = 24px)</p>
```

```

<div class="em-demo-margin">
  <p>Element นี้มีmargin: 2em = 32px (เทียบกับ parent)</p>
  <p>padding: 1em = 16px</p>
</div>
</div>

<div class="code-example">
  <code>
    >/* Em Unit - Relative to Parent Font Size */<br />
    .parent { font-size: 16px; }<br />
    .small { font-size: 0.8em; } /* = 12.8px */<br />
    .large { font-size: 1.5em; } /* = 24px */<br />
    .margin-demo { margin: 2em; } /* = 32px */
  </code>
</div>

<div class="use-case">
  <strong> ใช้em เมื่อ:</strong>
  <ul>
    <li>ต้องการ scalable components</li>
    <li>ต้องการให้padding/margin ขึ้นกับ font size</li>
    <li>ออกแบบ modular elements</li>
  </ul>
</div>

<!-- Rem Unit -->
<h3>Rem (rem) - เทียบกับ Root Font Size</h3>
<p>1rem = root element font-size (html { font-size: 16px; })</p>
<p>เหมือน em แต่อ้างอิงกับ root เท่านั้น ไม่ว่าจะ nesting ลึกแค่ไหน</p>

<div class="rem-demo">
  <div class="rem-demo-h">Heading: font-size: 2rem = 32px</div>
  <p class="rem-demo-p">
    Paragraph: font-size: 1rem = 16px<br />
    line-height: 1.5rem = 24px
  </p>
  <p style="font-size: 0.875rem; color: #666">
    Small text: font-size: 0.875rem = 14px
  </p>
</div>

<div class="code-example">
  <code>
    >/* Rem Unit - Relative to Root Font Size */<br />
    html { font-size: 16px; } /* root */<br />
    h1 { font-size: 2rem; } /* = 32px */<br />
    p { font-size: 1rem; } /* = 16px */<br />
    small { font-size: 0.875rem; } /* = 14px */
  </code>
</div>

<div class="use-case">
  <strong> ใช้rem เมื่อ:</strong>
  <ul>
    <li>ต้องการ consistent sizing ทั่วโครงการ</li>
    <li>ทำ responsive typography</li>
  </ul>

```

```

        <li>ง่ายต่อการ manage spacing</li>
    </ul>
</div>
</section>

<!-- Viewport Units -->
<section class="units-section">
  <h2>3. Viewport Units (หน่วยที่ขึ้นกับหน้าจอ)</h2>
  <p>
    หน่วยที่ขึ้นกับขนาดของ viewport (หน้าจอ) เหมาะสำหรับ responsive design
  </p>

  <h3>Viewport Width (vw) & Viewport Height (vh)</h3>
  <p>1vw = 1% ของ viewport width; 1vh = 1% ของ viewport height</p>

  <div class="viewport-demo">50vh = ครึ่งหนึ่งของหน้าจอ</div>

  <div class="viewport-demo-small">25vh = ผลสี่ของหน้าจอ</div>

  <div class="code-example">
    <code>
      >/* Viewport Units */<br />
      .full-screen { height: 100vh; } /* เต็มหน้าจอ */<br />
      .half-screen { height: 50vh; } /* ครึ่งหน้าจอ */<br />
      .full-width { width: 100vw; } /* เต็มความกว้าง */
    </code>
  </div>

  <div class="use-case">
    <strong>&input checked="" type="checkbox" /> ใช้ vw/vh เมื่อ:</strong>
    <ul>
      <li>ต้องการ full-screen sections</li>
      <li>ต้องการให้ element พอดีกับหน้าจอ</li>
      <li>ทำ hero sections, landing pages</li>
    </ul>
  </div>
</section>

<!-- Comparison -->
<section class="units-section">
  <h2>Unit Comparison & Use Cases</h2>

  <table class="units-table">
    <thead>
      <tr>
        <th>Unit</th>
        <th>ตัวอักษร</th>
        <th>อ้างอิงกับ</th>
        <th>Responsive</th>
        <th>ใช้เมื่อ</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td><strong>Pixel</strong></td>
        <td>px</td>
        <td>ไม่มี (fixed)</td>

```

```

        <td> No</td>
        <td>Borders, fixed layouts, precision</td>
    </tr>
    <tr>
        <td><strong>Percentage</strong></td>
        <td>%</td>
        <td>Parent element</td>
        <td><input checked="" type="checkbox"/> Yes</td>
        <td>Responsive widths, flexible layouts</td>
    </tr>
    <tr>
        <td><strong>Em</strong></td>
        <td>em</td>
        <td>Parent font-size</td>
        <td><input checked="" type="checkbox"/> Yes</td>
        <td>Scalable components, spacing</td>
    </tr>
    <tr>
        <td><strong>Rem</strong></td>
        <td>rem</td>
        <td>Root font-size</td>
        <td><input checked="" type="checkbox"/> Yes</td>
        <td>Consistent sizing, typography</td>
    </tr>
    <tr>
        <td><strong>Viewport Width</strong></td>
        <td>vw</td>
        <td>Viewport width</td>
        <td><input checked="" type="checkbox"/> Yes</td>
        <td>Full-width sections</td>
    </tr>
    <tr>
        <td><strong>Viewport Height</strong></td>
        <td>vh</td>
        <td>Viewport height</td>
        <td><input checked="" type="checkbox"/> Yes</td>
        <td>Full-height sections</td>
    </tr>
</tbody>
</table>
</section>

```

```
<!-- Visual Comparison -->
```

```
<section class="units-section">
```

```
  <h2>Visual Comparison</h2>
```

```
  <h3>Fixed vs Responsive Units</h3>
```

```
  <div class="units-comparison">
```

```
    <div class="unit-box px">Fixed: 200px</div>
```

```
    <div class="unit-box percent">Relative: 80%</div>
```

```
    <div class="unit-box vw">Viewport: 60vw</div>
```

```
  </div>
```

```
  <h3>Flexible Layouts</h3>
```

```
  <div class="responsive-demo">
```

```
    <div class="responsive-item">
```

```
      <p>Flex Item</p>
```

```
      <p style="font-size: 0.8em; color: #666">width: auto</p>
```

```

</div>
<div class="responsive-item">
  <p>Flex Item</p>
  <p style="font-size: 0.8em; color: #666">width: auto</p>
</div>
<div class="responsive-item">
  <p>Flex Item</p>
  <p style="font-size: 0.8em; color: #666">width: auto</p>
</div>
<div class="responsive-item-fixed">
  <p>Fixed: 200px</p>
  <p style="font-size: 0.8em; color: #666">width: 200px</p>
</div>
</div>
</section>

<!-- Best Practices -->
<section class="units-section">
  <h2>Best Practices</h2>

  <div
    class="use-case"
    style="background-color: #d1ecf1; border-left-color: #0c5460"
  >
    <strong style="color: #0c5460">1. ใช้rem สำหรับ Typography</strong>
    <p>
      ใช้rem สำหรับ font-size, line-height, spacing
      ทำให้ง่ายต่อการปรับแต่ง
    </p>
  </div>

  <div
    class="use-case"
    style="background-color: #d5f4e6; border-left-color: #27ae60"
  >
    <strong style="color: #27ae60">2. ใช้% สำหรับ Layouts</strong>
    <p>ใช้% สำหรับ width ของ containers เพื่อให้ responsive</p>
  </div>

  <div
    class="use-case"
    style="background-color: #fef5e7; border-left-color: #f39c12"
  >
    <strong style="color: #f39c12">3. ใช้px สำหรับ Small Details</strong>
    <p>ใช้px สำหรับ borders, shadows, small UI elements</p>
  </div>

  <div
    class="use-case"
    style="background-color: #fadbd8; border-left-color: #e74c3c"
  >
    <strong style="color: #e74c3c">
      >4. ใช้em สำหรับ Component Scaling</strong>
    >
    <p>
      ใช้em สำหรับ padding, margin ใน components ที่มี font-size
      เป็นตัวแปร
    </p>
  </div>

```

```

</div>

<div
  class="use-case"
  style="background-color: #f4ecf7; border-left-color: #8e44ad"
>
  <strong style="color: #8e44ad"
    >5. ใช้vw/vh สำหรับ Full-Screen</strong>
  </div>
  <p>ใช้vw/vh เมื่อต้องการให้element ครอบคลุมทั้งviewport</p>
</div>
</section>

<!-- Summary -->
<section class="units-section">
  <h2>สรุป</h2>
  <ul>
    <li><strong>px</strong> - สำหรับความแม่นยำสูง(borders, shadows)</li>
    <li>
      <strong>%</strong> - สำหรับ responsive widths และ flexible layouts
    </li>
    <li><strong>em</strong> - สำหรับ scalable components</li>
    <li>
      <strong>rem</strong> - สำหรับ consistent sizing ทั่วโครงการ (แนะนำ)
    </li>
    <li><strong>vw/vh</strong> - สำหรับ full-screen sections</li>
  </ul>
  <p style="margin-top: 20px; color: #666">
    💡 <strong>ทิป:</strong> ใช้rem และ % เป็นหลัก
    เพราะช่วยให้เว็บไซต์ตอบสนองต่อขนาดจอต่างๆ ได้ดีขึ้น
  </p>
</section>
</div>
</body>
</html>

```

2. ดูผลลัพธ์หน้าเว็บผ่าน Live Server

CSS Units - หน่วยวัด ใน CSS

เรียนรู้หน่วยวัดต่าง ๆ ที่ใช้ใน CSS และเมื่อไหร่ที่ควรใช้

บทนำ CSS Units

CSS มีหน่วยวัดมากมาย แต่ละหน่วยมีข้อดีและข้อเสียที่แตกต่างกัน การเลือกหน่วยที่ถูกต้องจะช่วยให้เว็บไซต์ของคุณสะดวกใช้งานและตอบสนองต่อขนาดจอได้ดี

1. Absolute Units (หน่วยคงที่)

หน่วยที่มีขนาดคงที่ ไม่ขึ้นกับค่าอื่น

Pixel (px) - หน่วยพิกเซล

หน่วยที่นิยมใช้มากที่สุด 1px = 1/96 inch

ตัวอย่าง: **width: 300px, padding: 20px**
Element นี้มีความกว้างคงที่ 300px
ขนาดจะไม่เปลี่ยนแปลง ไม่ว่าหน้าจอมีขนาดเท่าไร

Best Practices

1. ใช้ rem สำหรับ Typography

ใช้ rem สำหรับ font-size, line-height, spacing ทำให้ง่ายต่อการปรับแต่ง

2. ใช้ % สำหรับ Layouts

ใช้ % สำหรับ width ของ containers เพื่อให้ responsive

3. ใช้ px สำหรับ Small Details

ใช้ px สำหรับ borders, shadows, small UI elements

4. ใช้ em สำหรับ Component Scaling

ใช้ em สำหรับ padding, margin ใน components ที่มี font-size เป็นตัวแปร

5. ใช้ vw/vh สำหรับ Full-Screen

ใช้ vw/vh เมื่อต้องการให้ element ครอบคลุมทั้ง viewport

สรุป

- **px** - สำหรับความแม่นยำสูง (borders, shadows)
- **%** - สำหรับ responsive widths และ flexible layouts
- **em** - สำหรับ scalable components
- **rem** - สำหรับ consistent sizing ทั่วโครงการ (แนะนำ)
- **vw/vh** - สำหรับ full-screen sections

💡 ทิป: ใช้ rem และ % เป็นหลัก เพราะช่วยให้เว็บไซต์ตอบสนองต่อขนาดจอต่างๆ ได้ดีขึ้น

3. commit "add: Add CSS Units examples"

ตอนที่ 4: Box Model & Color & Text

1. สร้างไฟล์ box-model.html

```
<!DOCTYPE html>
<html lang="th">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CSS Box Model</title>
    <link rel="stylesheet" href="styles.css" />
    <style>
      /* =====
        BOX MODEL DEMONSTRATION
        ===== */

      .box-model-section {
        margin: 40px 0;
      }

      /* Basic box */
      .box {
        width: 300px;
        padding: 20px; /* ระยะห่างด้านใน */
        border: 3px solid #264de4; /* กรอบ */
        margin: 20px; /* ระยะห่างด้านนอก */
        background-color: #e8f4f8;
      }

      /* Box with border radius */
      .box-rounded {
        width: 300px;
        padding: 20px;
        border: 2px dashed #27ae60;
        border-radius: 10px;
        margin: 20px 0;
        background-color: #d5f4e6;
      }

      /* Box with different padding sides */
      .box-padding-demo {
        width: 280px;
        padding-top: 10px;
        padding-right: 20px;
        padding-bottom: 30px;
        padding-left: 40px;
        border: 2px solid #e74c3c;
        margin: 20px;
        background-color: #fadbd8;
      }

      /* Centered box using margin */
      .box-centered {
        width: 400px;
        padding: 20px;
        margin: 20px auto; /* center horizontally */
        border: 2px solid #f39c12;
        background-color: #fef5e7;
      }
    </style>
  </head>
  <body>
  </body>
</html>
```

```
/* Box sizing comparison */
.box-sizing-demo {
  display: flex;
  gap: 20px;
  margin: 20px 0;
  flex-wrap: wrap;
}

.box-content {
  width: 300px;
  padding: 20px;
  border: 2px solid #8e44ad;
  box-sizing: content-box; /* width = content only */
  background-color: #f4ecf7;
}

.box-border {
  width: 300px;
  padding: 20px;
  border: 2px solid #8e44ad;
  box-sizing: border-box; /* width = content + padding + border */
  background-color: #f4ecf7;
}

/* Box with shadow */
.box-shadow-demo {
  width: 300px;
  padding: 20px;
  border: 1px solid #ddd;
  margin: 20px;
  background-color: white;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  transition: box-shadow 0.3s ease;
}

.box-shadow-demo:hover {
  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
}

/* Card layout */
.card {
  width: 300px;
  border: 1px solid #ddd;
  border-radius: 8px;
  overflow: hidden;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
  margin: 20px;
}

.card-header {
  background-color: #264de4;
  color: white;
  padding: 15px;
}

.card-body {
  padding: 20px;
}
```

```

    .card-footer {
      background-color: #ecf0f1;
      padding: 15px;
      border-top: 1px solid #ddd;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>CSS Box Model & Colors & Text</h1>

    <section class="box-model-section">
      <h2>Box Model Components</h2>

      <!-- Basic Box -->
      <h3>1 Basic Box (Content + Border)</h3>
      <div class="box">
        <h4>Basic Box</h4>
        <p>
          นี่คือ box ธรรมดา มีเนื้อหา(content) padding border และ margin
        </p>
      </div>

      <!-- Box with Border Radius -->
      <h3>2 Box with Border Radius</h3>
      <div class="box-rounded">
        <h4>Rounded Box</h4>
        <p>Box นี้มีมุมโค้ง ด้วย border-radius property</p>
      </div>

      <!-- Padding Demo -->
      <h3>3 Different Padding Sides</h3>
      <p>
        <span class="highlight">padding-top: 10px</span>
        <span class="highlight">padding-right: 20px</span>
        <span class="highlight">padding-bottom: 30px</span>
        <span class="highlight">padding-left: 40px</span>
      </p>
      <div class="box-padding-demo">
        <h4>Padding Demo</h4>
        <p>Box นี้มี padding ต่างกันในแต่ละด้าน</p>
      </div>

      <!-- Centered Box -->
      <h3>4 Centered Box (margin: 0 auto)</h3>
      <div class="box-centered">
        <h4>Centered Box</h4>
        <p>Box นี้อยู่ตรงกลาง ด้วยการใส่ <code>margin: 0 auto</code></p>
      </div>

      <!-- Box Sizing -->
      <h3>5 Box Sizing Comparison</h3>
      <p>ลองเปรียบเทียบ content-box vs border-box (ความกว้าง 300px):</p>
      <div class="box-sizing-demo">
        <div>
          <p>
            <strong>content-box</strong> (ถ้าเริ่มต้น) <br /> width = content
            only
          </p>
        </div>
      </div>
    </div>
  </body>
</html>

```

```

    </p>
    <div class="box-content">
      <p>Width = 300px (content only)</p>
    </div>
  </div>
  <div>
    <p>
      <strong>border-box</strong> (แนะนำ<br />width = content + padding
      + border
    </p>
    <div class="box-border">
      <p>Width = 300px (รวมทั้งหมด)</p>
    </div>
  </div>
</div>

<!-- Box Shadow -->
<h3>6 Box with Shadow (Hover me)</h3>
<div class="box-shadow-demo">
  <h4>Shadow Box</h4>
  <p>Hover ตรงนี้เพื่อดูการเปลี่ยนแปลง shadow</p>
</div>

<!-- Card Component -->
<h3>7 Card Component (Real Example)</h3>
<div class="card">
  <div class="card-header">
    <h4>Card Title</h4>
  </div>
  <div class="card-body">
    <p>นี่คือเนื้อหาของ card ที่มีการจัดรูปแบบด้วย Box Model</p>
  </div>
  <div class="card-footer">
    <small>Footer content</small>
  </div>
</div>
</section>

<!-- Colors & Text -->
<section class="box-model-section">
  <h2>Colors & Text Styling</h2>

  <h3>Color Formats</h3>
  <div class="code-box">
    <code>
      > /* Named color */ color: red;<br />
      /* HEX */ color: #ff0000;<br />
      /* RGB */ color: rgb(255, 0, 0);<br />
      /* RGBA */ color: rgba(255, 0, 0, 0.5);</code>
    </div>

  <h3>Text Color Examples</h3>
  <p class="text-primary">📄 Primary Color (Blue)</p>
  <p class="text-success">✅ Success Color (Green)</p>
  <p class="text-warning">⚠️ Warning Color (Orange)</p>
  <p class="text-danger">❌ Danger Color (Red)</p>

```

```

<h3>Background Color Examples</h3>
<div class="bg-light">
  <p>Light Background</p>
</div>
<div class="bg-info">
  <p>Info Background with left border</p>
</div>

<h3>Font Styling Examples</h3>
<p class="font-bold">This text is BOLD</p>
<p class="font-italic">This text is ITALIC</p>
<p class="text-uppercase">this is uppercase</p>
<p class="text-center">This text is centered</p>
<p class="text-justify">
  <em>This is justified text.</em> Lorem ipsum dolor sit amet,
  consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut
  labore et dolore magna aliqua.
</p>

<h3>Font Family & Size</h3>
<p style="font-size: 12px">Small text (12px)</p>
<p style="font-size: 16px">Normal text (16px)</p>
<p style="font-size: 20px">Large text (20px)</p>
<p style="font-size: 28px">Very large text (28px)</p>

<p style="font-family: 'Courier New', monospace">
  Monospace font (code-like)
</p>
<p style="font-family: Georgia, serif">Serif font (traditional)</p>
<p style="font-family: Arial, sans-serif">Sans-serif font (modern)</p>

<h3>Line Height Example</h3>
<p style="line-height: 1">Line height 1.0 (tight)</p>
<p style="line-height: 1.6">
  Line height 1.6 (comfortable) - This is more readable
</p>
<p style="line-height: 2">
  Line height 2.0 (spacious) - This has more breathing room
</p>
</section>
</div>
</body>
</html>

```

4. ดูผลลัพธ์หน้าเว็บผ่าน Live Server

CSS Box Model & Colors & Text

Box Model Components

1 Basic Box (Content + Border)

Basic Box
นี่คือ box ธรรมดา มีเนื้อหา (content)
padding border และ margin

2 Box with Border Radius

Rounded Box
Box นี้มีมุมโค้ง ด้วย border-radius
property

3 Different Padding Sides

padding-top: 10px padding-right: 20px padding-bottom: 30px padding-left: 40px

Colors & Text Styling

Color Formats

```
/* Named color */ color: red;  
/* HEX */ color: #ff0000;  
/* RGB */ color: rgb(255, 0, 0);  
/* RGBA */ color: rgba(255, 0, 0, 0.5);
```

Text Color Examples

- Primary Color (Blue)
- ✓ Success Color (Green)
- ⚠ Warning Color (Orange)
- ✗ Danger Color (Red)

Background Color Examples

Light Background

Info Background with left border

Font Styling Examples

5. commit "add: Add box model and color & text demonstration"

ตอนที่ 5: Positioning

1. สร้างไฟล์ positioning.html

```
<!DOCTYPE html>  
<html lang="th">  
<head>
```



```
padding: 10px;
border-radius: 5px;
display: flex;
align-items: center;
justify-content: center;
font-size: 12px;
font-weight: bold;
}

.absolute-child.top-left {
  top: 10px;
  left: 10px;
}

.absolute-child.top-right {
  top: 10px;
  right: 10px;
}

.absolute-child.bottom-left {
  bottom: 10px;
  left: 10px;
}

.absolute-child.bottom-right {
  bottom: 10px;
  right: 10px;
}

.absolute-child.centered {
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  background-color: #f39c12;
}

/* Fixed Position */
.fixed-demo {
  position: fixed;
  bottom: 20px;
  right: 20px;
  width: 60px;
  height: 60px;
  background-color: #264de4;
  color: white;
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 30px;
  cursor: pointer;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
  transition: all 0.3s ease;
  z-index: 999;
}

.fixed-demo:hover {
  background-color: #1a3a8f;
  box-shadow: 0 6px 12px rgba(0, 0, 0, 0.3);
}
```

```
}

/* Sticky Position */
.sticky-header {
  position: sticky;
  top: 0;
  background-color: #3498db;
  color: white;
  padding: 15px;
  margin-top: 20px;
  border-radius: 4px 4px 0 0;
  z-index: 100;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.sticky-content {
  background-color: #d6eaf8;
  padding: 20px;
  border-radius: 0 0 4px 4px;
  margin-bottom: 20px;
}

/* Stacking order (z-index) */
.stacking-demo {
  position: relative;
  height: 200px;
  margin: 20px 0;
  background-color: #ecf0f1;
  border: 2px solid #95a5a6;
}

.box-z1 {
  position: absolute;
  width: 150px;
  height: 100px;
  background-color: #e74c3c;
  top: 20px;
  left: 20px;
  z-index: 1;
  display: flex;
  align-items: center;
  justify-content: center;
  color: white;
  font-weight: bold;
}

.box-z2 {
  position: absolute;
  width: 150px;
  height: 100px;
  background-color: #f39c12;
  top: 40px;
  left: 100px;
  z-index: 2;
  display: flex;
  align-items: center;
  justify-content: center;
  color: white;
  font-weight: bold;
}
```

```

}

.box-z3 {
  position: absolute;
  width: 150px;
  height: 100px;
  background-color: #27ae60;
  top: 80px;
  left: 180px;
  z-index: 3;
  display: flex;
  align-items: center;
  justify-content: center;
  color: white;
  font-weight: bold;
}

/* Badge with absolute position */
.badge-container {
  position: relative;
  width: 200px;
  height: 200px;
  background-color: #d6eaf8;
  border: 2px solid #3498db;
  border-radius: 8px;
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 20px;
}

.badge {
  position: absolute;
  top: -10px;
  right: -10px;
  width: 40px;
  height: 40px;
  background-color: #e74c3c;
  color: white;
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  font-weight: bold;
  font-size: 18px;
}
</style>
</head>
<body>
<div class="container">
  <h1>CSS Positioning</h1>

  <!-- Static Position -->
  <section class="positioning-section">
    <h2>1. Static Position (ค่าเริ่มต้น)</h2>
    <p>Static คือค่าเริ่มต้น element อยู่ในตำแหน่งปกติใน document flow</p>
    <div class="static-demo">
      <p>
        This is static positioned element<br />

```

→ อยู่ในลำดับปกติของ HTML

```
</p>  
</div>  
</section>
```

```
<!-- Relative Position -->  
<section class="positioning-section">  
  <h2>2. Relative Position (สัมพันธ์กับตำแหน่งเดิม)</h2>  
  <p>Relative ข้ายตำแหน่งเทียบกับตำแหน่งเดิม แต่ยังคงใช้พื้นที่เดิม</p>  
  <div class="relative-demo">  
    <p>ไม่มีมีการเลื่อน</p>  
  </div>  
  <div class="relative-demo example">  
    <p>เลื่อนด้วย relative positioning</p>  
  </div>  
</section>
```

```
<!-- Absolute Position -->  
<section class="positioning-section">  
  <h2>3. Absolute Position (สัมพันธ์กับ parent)</h2>  
  <p>  
    Absolute นำ element ออกจาก flow และวางตำแหน่งเทียบกับ parent ที่มี  
    position  
  </p>  
  
  <h3>ตำแหน่งต่างๆ ใน Container:</h3>  
  <div class="absolute-container">  
    <div class="absolute-child top-left">Top Left</div>  
    <div class="absolute-child top-right">Top Right</div>  
    <div class="absolute-child bottom-left">Bottom Left</div>  
    <div class="absolute-child bottom-right">Bottom Right</div>  
    <div class="absolute-child centered">Center</div>  
  </div>  
  
  <h3>ตัวอย่าง: Badge on Card</h3>  
  <div class="badge-container">  
    <p>Product Card</p>  
    <div class="badge">🔥</div>  
  </div>  
</section>
```

```
<!-- Fixed Position -->  
<section class="positioning-section">  
  <h2>4. Fixed Position (สัมพันธ์กับ viewport)</h2>  
  <p>  
    Fixed ติดกับหน้าจอ ไม่เลื่อนไปกับ scroll content (ดูปุ่มด้านล่างขวา)  
  </p>  
  <p>ปุ่ม + ด้านล่างขวาของหน้าจอคือ fixed element</p>  
</section>
```

```
<!-- Sticky Position -->  
<section class="positioning-section">  
  <h2>5. Sticky Position (Hybrid)</h2>  
  <p>  
    Sticky รวม relative และ fixed - ปกติ relative แต่ sticky เมื่อ scroll  
  </p>
```

```

<div class="sticky-header">Sticky Header - Scroll ลงมาดู!</div>
<div class="sticky-content">
  <p>Sticky position ทำให้header ติดด้านบนเมื่อ scroll</p>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua.
  </p>
  <p>
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.
  </p>
  <p>
    Duis aute irure dolor in reprehenderit in voluptate velit esse
    cillum dolore eu fugiat nulla pariatur.
  </p>
  <p>
    Excepteur sint occaecat cupidatat non proident, sunt in culpa qui
    officia deserunt mollit anim id est laborum.
  </p>
</div>
</section>

<!-- Stacking Order -->
<section class="positioning-section">
  <h2>6. Stacking Order (z-index)</h2>
  <p>z-index ควบคุมลำดับการ stack ซึ่งสูง ยิ่งอยู่บนสุด</p>
  <div class="stacking-demo">
    <div class="box-z1">z-index: 1</div>
    <div class="box-z2">z-index: 2</div>
    <div class="box-z3">z-index: 3</div>
  </div>
</section>

<!-- Position Comparison Table -->
<section class="positioning-section">
  <h2>Position Values Comparison</h2>
  <table style="width: 100%; border-collapse: collapse; margin-top: 20px">
    <thead>
      <tr
        style="background-color: #264de4; color: white; text-align: left"
      >
        <th style="padding: 10px; border: 1px solid #ddd">Position</th>
        <th style="padding: 10px; border: 1px solid #ddd">อ้างอิงกับ</th>
        <th style="padding: 10px; border: 1px solid #ddd">ใช้เมื่อ</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td style="padding: 10px; border: 1px solid #ddd">
          <strong>static</strong>
        </td>
        <td style="padding: 10px; border: 1px solid #ddd">
          Document flow
        </td>
        <td style="padding: 10px; border: 1px solid #ddd">
          ไม่ต้องย้ายตำแหน่ง
        </td>
      </tr>
    </tbody>
  </table>

```

```

<tr>
  <td style="padding: 10px; border: 1px solid #ddd">
    <strong>relative</strong>
  </td>
  <td style="padding: 10px; border: 1px solid #ddd">ตำแหน่งเดิม</td>
  <td style="padding: 10px; border: 1px solid #ddd">
    ปรับแต่งเล็กน้อย
  </td>
</tr>
<tr>
  <td style="padding: 10px; border: 1px solid #ddd">
    <strong>absolute</strong>
  </td>
  <td style="padding: 10px; border: 1px solid #ddd">
    Parent ที่มี position
  </td>
  <td style="padding: 10px; border: 1px solid #ddd">
    วาง badge, tooltip
  </td>
</tr>
<tr>
  <td style="padding: 10px; border: 1px solid #ddd">
    <strong>fixed</strong>
  </td>
  <td style="padding: 10px; border: 1px solid #ddd">Viewport</td>
  <td style="padding: 10px; border: 1px solid #ddd">
    Navbar, floating button
  </td>
</tr>
<tr>
  <td style="padding: 10px; border: 1px solid #ddd">
    <strong>sticky</strong>
  </td>
  <td style="padding: 10px; border: 1px solid #ddd">
    Parent + Viewport
  </td>
  <td style="padding: 10px; border: 1px solid #ddd">
    Table header, sticky nav
  </td>
</tr>
</tbody>
</table>
</section>
</div>

<!-- Fixed Button -->
<div class="fixed-demo">+</div>
</body>
</html>

```

2. ดูผลลัพธ์หน้าเว็บผ่าน Live Server

CSS Positioning

1. Static Position (ค่าเริ่มต้น)

Static คือค่าเริ่มต้น element อยู่ในตำแหน่งปกติใน document flow

This is static positioned element
→ อยู่ในลำดับปกติของ HTML

2. Relative Position (สัมพันธ์กับตำแหน่งเดิม)

Relative ย้ายตำแหน่งเทียบกับตำแหน่งเดิม แต่ยังคงใช้พื้นที่เดิม

ไม่มีการเลื่อน
↘ เลื่อน ขึ้น 20px ไป 30px

เลื่อนด้วย relative positioning
↘ เลื่อน ขึ้น 20px ไป 30px

3. Absolute Position (สัมพันธ์กับ parent)

Absolute นำ element ออกจาก flow และวางตำแหน่งเทียบกับ parent ที่มี position

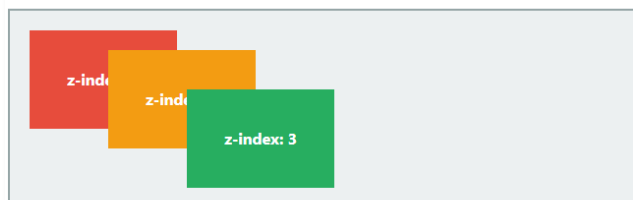
Sticky Header - Scroll ลงมาดู!

nona piamur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

6. Stacking Order (z-index)

z-index ควบคุมลำดับการ stack ยิ่งสูง ยิ่งอยู่บนสุด



Position Values Comparison

Position	อ้างอิงกับ	ใช้เมื่อ
static	Document flow	ไม่ต้องย้ายตำแหน่ง
relative	ตำแหน่งเดิม	ปรับแต่งเล็กน้อย
absolute	Parent ที่มี position	วาง badge, tooltip
fixed	Viewport	Navbar, floating button
sticky	Parent + Viewport	Table header, sticky nav



3. commit "add: Add positioning examples (static, relative, absolute, fixed, sticky)"

ตอนที่ 6: Display & Layout

1. สร้างไฟล์ display-layout.html

```
<!DOCTYPE html>
<html lang="th">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CSS Display & Layout</title>
    <link rel="stylesheet" href="styles.css" />
    <style>
      /* =====
        DISPLAY PROPERTY DEMONSTRATION
        ===== */

      .display-section {
        margin: 40px 0;
      }

      /* Display: Block */
      .block-demo {
        display: block;
        width: 300px;
        background-color: #d5f4e6;
        border: 2px solid #27ae60;
        padding: 15px;
        margin: 10px 0;
      }

      /* Display: Inline */
      .inline-demo {
        display: inline;
        background-color: #fef5e7;
        border: 2px solid #f39c12;
        padding: 5px 10px;
        margin: 0 5px;
      }

      /* Display: Inline-Block */
      .inline-block-demo {
        display: inline-block;
        width: 150px;
        background-color: #fadbd8;
        border: 2px solid #e74c3c;
        padding: 15px;
        margin: 10px 5px;
        vertical-align: top;
      }

      /* Display: None */
      .hidden {
        display: none;
      }

      .visible {
        display: block;
      }

      /* Flexbox Layout */
```

```
.flex-container {
  display: flex;
  gap: 20px;
  flex-wrap: wrap;
  margin: 20px 0;
}

.flex-item {
  flex: 1;
  min-width: 200px;
  background-color: #d1ecf1;
  border: 2px solid #0c5460;
  padding: 20px;
  border-radius: 4px;
}

/* Grid Layout */
.grid-container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 20px;
  margin: 20px 0;
}

.grid-item {
  background-color: #f0f3ff;
  border: 2px solid #264de4;
  padding: 20px;
  text-align: center;
  border-radius: 4px;
}

@media (max-width: 768px) {
  .grid-container {
    grid-template-columns: repeat(2, 1fr);
  }
}

@media (max-width: 480px) {
  .grid-container {
    grid-template-columns: 1fr;
  }
}

/* Two Column Layout */
.two-column {
  display: flex;
  gap: 20px;
  margin: 20px 0;
}

.main-content {
  flex: 2;
  background-color: #ecf0f1;
  padding: 20px;
  border-radius: 4px;
}
```

```

.sidebar-col {
  flex: 1;
  background-color: #d5dbdb;
  padding: 20px;
  border-radius: 4px;
}
</style>
</head>

<body>
  <div class="container">
    <h1>CSS Display & Layout</h1>

    <!-- Display Values -->
    <section class="display-section">
      <h2>Display Property Values</h2>

      <h3>1. Display: Block</h3>
      <p>element ทำให้เต็มความกว้างที่มีได้ และ line break:</p>
      <div class="block-demo">
        <p>ยินดีต้อนรับสู่ block element</p>
      </div>
      <div class="block-demo">
        <p>block element อยู่บนบรรทัดใหม่</p>
      </div>

      <h3>2. Display: Inline</h3>
      <p>element อยู่ในบรรทัดเดียวกัน ไม่สามารถกำหนด width/height ได้:</p>
      <p>
        <span class="inline-demo">inline element 1</span>
        <span class="inline-demo">inline element 2</span>
        <span class="inline-demo">inline element 3</span>
      </p>

      <h3>3. Display: Inline-Block</h3>
      <p>
        รวม inline และ block - อยู่ในบรรทัดเดียวแต่สามารถกำหนด width/height:
      </p>
      <div>
        <div class="inline-block-demo">
          <p>Inline-Block 1</p>
        </div>
        <div class="inline-block-demo">
          <p>Inline-Block 2</p>
        </div>
        <div class="inline-block-demo">
          <p>Inline-Block 3</p>
        </div>
      </div>

      <h3>4. Display: None</h3>
      <p>ซ่อน element โดยสิ้นเชิง (ไม่ใช่พื้นที่):</p>
      <p class="hidden">✘ สมาชิก องค์ประกอบนี้ถูกซ่อนด้วย display: none</p>
      <p class="visible">☑ สมาชิก องค์ประกอบนี้ถูกแสดงด้วย display: block</p>
    </section>
  </div>
</body>

```

```

<!-- Flexbox -->
<section class="display-section">
  <h2>Flexbox Layout</h2>
  <p>Flexbox เป็น layout model ที่ช่วยจัด element ในแนวนอนหรือแนวตั้ง</p>

  <h3>Flex Container (display: flex)</h3>
  <div class="flex-container">
    <div class="flex-item">Flex Item 1</div>
    <div class="flex-item">Flex Item 2</div>
    <div class="flex-item">Flex Item 3</div>
  </div>

  <h3>Flex Container with responsive</h3>
  <div class="flex-container">
    <div class="flex-item">Design</div>
    <div class="flex-item">Code</div>
    <div class="flex-item">Deploy</div>
    <div class="flex-item">Analyze</div>
  </div>
</section>

<!-- Grid Layout -->
<section class="display-section">
  <h2>CSS Grid Layout</h2>
  <p>
    CSS Grid เป็น layout system ที่ช่วยจัด element เป็นแบบ 2D (rows &
    columns)
  </p>

  <h3>Grid Container (display: grid)</h3>
  <div class="grid-container">
    <div class="grid-item">Grid Item 1</div>
    <div class="grid-item">Grid Item 2</div>
    <div class="grid-item">Grid Item 3</div>
    <div class="grid-item">Grid Item 4</div>
    <div class="grid-item">Grid Item 5</div>
    <div class="grid-item">Grid Item 6</div>
  </div>
</section>

<!-- Two Column Layout -->
<section class="display-section">
  <h2>Two Column Layout with Flexbox</h2>

  <div class="two-column">
    <div class="main-content">
      <h3>Main Content</h3>
      <p>นี่คือเนื้อหาหลัก ซึ่งใช้พื้นที่มากขึ้น (flex: 2)</p>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do
        eiusmod tempor incididunt ut labore et dolore magna aliqua.
      </p>
    </div>
    <aside class="sidebar-col">
      <h3>Sidebar</h3>
      <p>เนื้อหาเสริม (flex: 1)</p>
      <ul>
        <li>Link 1</li>
        <li>Link 2</li>
      </ul>
    </aside>
  </div>
</section>

```

```

    <li>Link 3</li>
  </ul>
</aside>
</div>
</section>

<!-- Display Comparison Table -->
<section class="display-section">
  <h2>Display Values Comparison</h2>
  <table style="width: 100%; border-collapse: collapse; margin-top: 20px">
    <thead>
      <tr
        style="background-color: #264de4; color: white; text-align: left"
      >
        <th style="padding: 10px; border: 1px solid #ddd">Display</th>
        <th style="padding: 10px; border: 1px solid #ddd">
          Width/Height
        </th>
        <th style="padding: 10px; border: 1px solid #ddd">Line Break</th>
        <th style="padding: 10px; border: 1px solid #ddd">👉👈</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td style="padding: 10px; border: 1px solid #ddd">
          <strong>block</strong>
        </td>
        <td style="padding: 10px; border: 1px solid #ddd">☑ Yes</td>
        <td style="padding: 10px; border: 1px solid #ddd">☑ Yes</td>
        <td style="padding: 10px; border: 1px solid #ddd">
          Paragraph, sections
        </td>
      </tr>
      <tr>
        <td style="padding: 10px; border: 1px solid #ddd">
          <strong>inline</strong>
        </td>
        <td style="padding: 10px; border: 1px solid #ddd">✗ No</td>
        <td style="padding: 10px; border: 1px solid #ddd">✗ No</td>
        <td style="padding: 10px; border: 1px solid #ddd">
          Links, spans
        </td>
      </tr>
      <tr>
        <td style="padding: 10px; border: 1px solid #ddd">
          <strong>inline-block</strong>
        </td>
        <td style="padding: 10px; border: 1px solid #ddd">☑ Yes</td>
        <td style="padding: 10px; border: 1px solid #ddd">✗ No</td>
        <td style="padding: 10px; border: 1px solid #ddd">
          Buttons, small boxes
        </td>
      </tr>
      <tr>
        <td style="padding: 10px; border: 1px solid #ddd">
          <strong>flex</strong>
        </td>
        <td style="padding: 10px; border: 1px solid #ddd">☑ Yes</td>

```

```

<td style="padding: 10px; border: 1px solid #ddd">Flexible</td>
<td style="padding: 10px; border: 1px solid #ddd">
  Flexible layouts
</td>
</tr>
<tr>
<td style="padding: 10px; border: 1px solid #ddd">
  <strong>grid</strong>
</td>
<td style="padding: 10px; border: 1px solid #ddd"><input checked="" type="checkbox"> Yes</td>
<td style="padding: 10px; border: 1px solid #ddd">Flexible</td>
<td style="padding: 10px; border: 1px solid #ddd">
  Complex layouts
</td>
</tr>
<tr>
<td style="padding: 10px; border: 1px solid #ddd">
  <strong>none</strong>
</td>
<td style="padding: 10px; border: 1px solid #ddd">N/A</td>
<td style="padding: 10px; border: 1px solid #ddd">N/A</td>
<td style="padding: 10px; border: 1px solid #ddd">
  Hide elements
</td>
</tr>
</tbody>
</table>
</section>
</div>
</body>
</html>

```

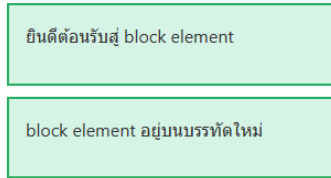
2. ดูผลลัพธ์หน้าเว็บผ่าน Live Server

CSS Display & Layout

Display Property Values

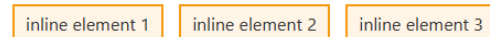
1. Display: Block

element ทำให้เต็มความกว้างที่มีได้ และ line break:



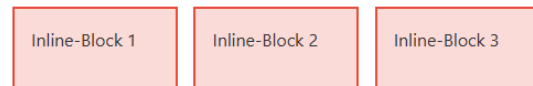
2. Display: Inline

element อยู่ในบรรทัดเดียวกัน ไม่สามารถกำหนด width/height ได้:



3. Display: Inline-Block

รวม inline และ block - อยู่ในบรรทัดเดียวแต่สามารถกำหนด width/height:



4. Display: None

ซ่อน element โดยสิ้นเชิง (ไม่ใช้พื้นที่):

สมาชิก องค์ประกอบนี้ถูกแสดงด้วย display: block

3. commit "add: Add display and layout examples (block, inline, flex, grid)"

ตอนที่ 7: Final Review & Summary

1. สร้างไฟล์ README.md ดังนี้

```
# Week 3 Lab: CSS Fundamentals

## ปฏิบัติการนี้ครอบคลุมเนื้อหา CSS ตั้งแต่พื้นฐาน ได้แก่

1. **CSS Syntax & Properties** - โครงสร้างและคุณสมบัติพื้นฐาน
2. ** CSS Units** - หน่วยวัดต่าง ๆ ของ CSS
2. **CSS Selectors** - การเลือก element ด้วย selector ต่างๆ
3. **Colors & Text** - การจัดรูปแบบสี และข้อความ
4. **Box Model** - การทำความเข้าใจ Content, Padding, Border, Margin
5. **Positioning** - Static, Relative, Absolute, Fixed, Sticky
6. **Display & Layout** - Block, Inline, Flexbox, Grid

---

## 📁 File Structure

...

wk03-lab/
├── index.html           # หน้าแรก
├── styles.css          # External CSS stylesheet
├── units.html          # CSS Units - หน่วยวัด
├── box-model.html     # Box Model & Colors & Text
├── positioning.html   # Positioning examples
├── display-layout.html # Display & Layout
└── README.md          # Documentation

...
```

🎯 Key Concepts

CSS Selectors Priority (Specificity)

...

Universal (*) < Element < Class < ID < Inline Style

...

ยิ่งเฉพาะเจาะจง priority ยิ่งสูง

Box Model

...

Content → Padding → Border → Margin

...

- **Content**: เนื้อหาจริง (text, images, etc.)
- **Padding**: ระยะห่างด้านใน (ระหว่าง content และ border)
- **Border**: กรอบรอบ element
- **Margin**: ระยะห่างด้านนอก (ระหว่าง element นี้กับ element อื่น)

Positioning Values

Position	อ้างอิงกับ	ใช้เมื่อ	ตัวอย่าง
static	Document flow	ไม่ต้องย้าย	ข้อความปกติ
relative	ตำแหน่งเดิม	ปรับแต่งเล็กน้อย	ย้ายข้อความขึ้น
absolute	Parent ที่มี position	วางตำแหน่งอิสระ	Badge, tooltip
fixed	Viewport	ติดกับหน้าจอ	Navbar, button
sticky	Parent + Viewport	ติดเมื่อ scroll ถึง	Table header

Display Values

Display	Width/Height	Line Break	ใช้เมื่อ
block	✔ Yes	✔ Yes	Paragraph, div
inline	✘ No	✘ No	span, link
inline-block	✔ Yes	✘ No	Button, small box
flex	✔ Yes	Flexible	Flexible layout
grid	✔ Yes	Flexible	Complex layout
none	N/A	N/A	Hide element

CSS Units (หน่วยวัด)

Unit	ตัวอักษร	อ้างอิงกับ	Responsive	ใช้เมื่อ
Pixel	px	ไม่มี (fixed)	✘ No	Borders, fixed layouts, precision
Percentage	%	Parent element	✔ Yes	Responsive widths, flexible layouts
Em	em	Parent font-size	✔ Yes	Scalable components, spacing
Rem	rem	Root font-size	✔ Yes	Consistent sizing, typography (แนะนำ)
Viewport Width	vw	Viewport width	✔ Yes	Full-width sections
Viewport Height	vh	Viewport height	✔ Yes	Full-height sections

💡 ทิป: ใช้ `rem` และ `%` เป็นหลัก เพราะช่วยให้เว็บไซต์ตอบสนองต่อขนาดจอต่างๆ ได้ดีขึ้น

การส่งปฏิบัติการ

1. สร้าง Repository ใหม่ชื่อ wk03-lab ที่ Github
2. push ไฟล์ใน Local repository ในเครื่อง ไปยัง remote repository wk03-lab ที่ Github
3. ส่ง repository url <https://github.com/.../wk03-lab> ที่ไฟล์นี้ [Assignment](#)

OJO