เนื้อหาการสอนสัปดาห์ที่ 1: พื้นฐานการพัฒนาเว็บ และเครื่องมือที่ใช้

หัวข้อ: พื้นฐานการพัฒนาเว็บ และเครื่องมือที่ใช้
ระยะเวลา: 2 ชั่วโมง

1. แนะนำภาพรวมการพัฒนาเว็บไซต์ (Introduction to Web Development) (30 นาที)

- **เว็บไซต์คืออะไร?**
  - ความหมายของเว็บไซต์ และเว็บเพจ
  - ประเภทของเว็บไซต์ (เช่น เว็บไซต์ส่วนตัว, เว็บบล็อก, เว็บอีคอมเมิร์ซ, เว็บแอปพลิเคชัน)
  - ประโยชน์ของเว็บไซต์ในยุคปัจจุบัน
- **ส่วนประกอบหลักของเว็บไซต์:**
  - **Frontend (ส่วนหน้าบ้าน):** สิ่งที่ผู้ใช้มองเห็นและโต้ตอบด้วย (User Interface - UI, User Experience - UX)
    - เทคโนโลยีหลัก: HTML, CSS, JavaScript
  - **Backend (ส่วนหลังบ้าน):** ส่วนที่จัดการข้อมูล ตรรกะของระบบ และการติดต่อกับฐานข้อมูล
    - ภาษาโปรแกรมที่นิยม: Python (Django, Flask), JavaScript (Node.js), PHP (Laravel), Ruby (Ruby on Rails), Java (Spring)
    - ฐานข้อมูล (Database): SQL (MySQL)
  - **เซิร์ฟเวอร์ (Server) และโฮสติ้ง (Hosting):** ที่สำหรับเก็บไฟล์เว็บไซต์และทำให้เข้าถึงได้ผ่านอินเทอร์เน็ต
- **กระบวนการทำงานของเว็บไซต์:**
  - ผู้ใช้พิมพ์ URL ในเบราว์เซอร์
  - เบราว์เซอร์ส่งคำขอ (Request) ไปยังเซิร์ฟเวอร์
  - เซิร์ฟเวอร์ประมวลผลคำขอและส่งข้อมูล (Response) กลับมา (ไฟล์ HTML, CSS, JavaScript)
  - เบราว์เซอร์แสดงผลหน้าเว็บให้ผู้ใช้เห็น

2. เทคโนโลยีหลักในการพัฒนาเว็บไซต์ (Core Web Technologies) (60 นาที)

- **HTML (HyperText Markup Language): โครงสร้างของหน้าเว็บ**
  - ความสำคัญของ HTML: เป็นเหมือนโครงกระดูกของหน้าเว็บ
  - โครงสร้างพื้นฐานของเอกสาร HTML: <!DOCTYPE html>, <html>, <head>, <body>
  - แท็ก (Tags) ที่สำคัญเบื้องต้น:
    - Headings: <h1> ถึง <h6>
    - Paragraph: <p>
    - Links: <a> (attribute href)
    - Images: <img> (attributes src, alt)

        - Lists: <ul> (unordered list), <ol> (ordered list), <li> (list item)
        - Divisions and Spans: (block-level container), (inline container)
    - **ตัวอย่างโค้ด HTML ง่ายๆ:**

```
<!DOCTYPE html>
<html>
<head>
  <title>หน้าเว็บแรกของฉัน</title>
</head>
<body>
  <h1>สวัสดีชาวโลก!</h1>
  <p>นี่คือย่อหน้าแรกของฉันบนเว็บไซต์นี้</p>
  <a href="https://www.google.com">ไปที่ Google</a>
  <img src="https://placehold.co/600x400/EEE/31343C?text=รูปภาพตัวอย่าง" alt="รูปภาพตัวอย่าง">
</body>
</html>
```

- **CSS (Cascading Style Sheets): การตกแต่งหน้าเว็บ**
    - ความสำคัญของ CSS: ทำให้หน้าเว็บสวยงามและน่าสนใจ
    - วิธีการนำ CSS มาใช้กับ HTML:
        - Inline CSS (ในแท็ก HTML โดยตรง)
        - Internal CSS (ในแท็ก <style> ภายใน <head>)
        - External CSS (ไฟล์ .css ภายนอก แล้วเชื่อมโยงด้วยแท็ก <link>) (แนะนำวิธีนี้)
    - Selectors (ตัวเลือก) พื้นฐาน:
        - Element Selector (เลือกจากชื่อแท็ก): p { color: blue; }
        - Class Selector (เลือกจาก attribute class): .my-class { font-size: 16px; }
        - ID Selector (เลือกจาก attribute id): #my-id { background-color: yellow; }
    - Properties (คุณสมบัติ) ที่ใช้บ่อย: color, background-color, font-size, font-family, margin, padding, border, width, height
    - **ตัวอย่างโค้ด CSS ง่ายๆ (บันทึกเป็น style.css):**

```
body {
  font-family: sans-serif;
  margin: 20px;
  background-color: #f0f0f0;
}
h1 {
  color: navy;
  text-align: center;
```

```
}
p {
  color: #333;
  line-height: 1.6;
}
.highlight {
  background-color: yellow;
}
```

- **การเชื่อมโยงไฟล์ CSS ใน HTML:**

```
<head>
  <title>หน้าเว็บแรกของฉัน</title>
  <link rel="stylesheet" href="style.css">
</head>
```

- **JavaScript (JS): การเพิ่มความสามารถในการโต้ตอบ (Interactivity)**
  - ความสำคัญของ JavaScript: ทำให้เว็บไซต์มีการตอบสนองต่อผู้ใช้ เช่น การคลิกปุ่ม, การแสดงผลข้อมูลแบบไดนามิก
  - วิธีการนำ JavaScript มาใช้กับ HTML:
    - Internal JavaScript (ในแท็ก <script> ภายใน <body> หรือ <head>)
    - External JavaScript (ไฟล์ .js ภายนอก แล้วเชื่อมโยงด้วยแท็ก <script src="..."></script>) (แนะนำวิธีนี้)
  - แนวคิดเบื้องต้น:
    - Variables (ตัวแปร)
    - Functions (ฟังก์ชัน)
    - Events (เหตุการณ์ เช่น onclick, onmouseover)
    - DOM Manipulation (การเปลี่ยนแปลงเนื้อหาและโครงสร้างของ HTML)
  - **ตัวอย่างโค้ด JavaScript ง่ายๆ (บันทึกเป็น script.js):**

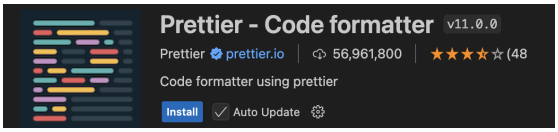
```
function showAlert() {
  alert('คุณคลิกปุ่มแล้ว!');
}
// สามารถเพิ่มโค้ดเพื่อผูกฟังก์ชันกับปุ่มได้เมื่อ HTML โหลดเสร็จ
// window.onload = function() {
//     const myButton = document.getElementById('myButton');
//     if (myButton) {
//         myButton.onclick = showAlert;
//     }
// };
```

- **การเชื่อมโยงไฟล์ JavaScript และการสร้างปุ่มใน HTML:**

```
<body>
  <button onclick="showAlert()">คลิกฉันสิ!</button>
  <script src="script.js"></script>
</body>
```

3. เครื่องมือที่จำเป็นสำหรับการพัฒนาเว็บไซต์ (Essential Tools) (30 นาที)

- **เว็บเบราว์เซอร์ (Web Browser):**
  - หน้าที่: แสดงผลหน้าเว็บ, มีเครื่องมือสำหรับนักพัฒนา (Developer Tools)
  - ตัวอย่าง: Google Chrome, Mozilla Firefox, Microsoft Edge
  - แนะนำการใช้ Developer Tools (Inspect Element, Console, Network)
- **โปรแกรมแก้ไขโค้ด (Code Editor):**
  - หน้าที่: ช่วยในการเขียนโค้ดได้สะดวกขึ้น มี syntax highlighting, code completion
  - แนะนำ: **Visual Studio Code (VS Code)**
    - วิธีดาวน์โหลดและติดตั้ง: https://code.visualstudio.com/
    - Extensions ที่มีประโยชน์เบื้องต้น: Live Server, Prettier - Code formatter



- **Node.js และ npm (Node Package Manager):**
  - **Node.js:** สภาพแวดล้อมสำหรับการรัน JavaScript นอกเบราว์เซอร์ (จำเป็นสำหรับ Next.js)
  - **npm:** เครื่องมือจัดการแพ็กเกจสำหรับ JavaScript (มาพร้อมกับ Node.js)
  - วิธีดาวน์โหลดและติดตั้ง Node.js (LTS version): https://nodejs.org/
  - วิธีตรวจสอบการติดตั้ง (เปิด Terminal หรือ Command Prompt):
    - node -v (ตรวจสอบเวอร์ชัน Node.js)
    - npm -v (ตรวจสอบเวอร์ชัน npm)

4. กิจกรรม/ปฏิบัติการ (Activity/Lab) (ในห้องเรียนหรือเป็นการบ้าน)

1. **ติดตั้งโปรแกรม:**
   - ติดตั้ง Visual Studio Code
   - ติดตั้ง Node.js (LTS)
2. **สร้างโปรเจกต์แรก:**
   - สร้างโฟลเดอร์ใหม่สำหรับโปรเจกต์ (เช่น my-first-website)
   - เปิดโฟลเดอร์นี้ด้วย VS Code
   - สร้างไฟล์ index.html และใส่โค้ด HTML พื้นฐาน (ตามตัวอย่างด้านบน)
   - สร้างไฟล์ style.css และใส่โค้ด CSS พื้นฐาน (ตามตัวอย่างด้านบน) และเชื่อมโยงกับ

index.html
    - (ถ้ามีเวลา) สร้างไฟล์ script.js และใส่โค้ด JavaScript ง่ายๆ (เช่น alert เมื่อคลิกปุ่ม) และเชื่อมโยงกับ index.html
3. **ทดลองเปิดไฟล์ HTML ด้วยเบราว์เซอร์:**
    - เปิดไฟล์ index.html โดยตรงในเบราว์เซอร์
    - ทดลองแก้ไขโค้ด HTML, CSS และดูการเปลี่ยนแปลง