# เปรียบเทียบ Utility Class กับ CSS

|  |  |  |
| --- | --- | --- |
| ลักษณะ | Utility class | CSS |
| แนวคิด | ใช้คลาสเล็ก ๆ ที่มีสไตล์เฉพาะเจาะจง (e.g., text-center, mt-4) | สร้างคลาสที่มีสไตล์ที่กำหนดเองโดยเฉพาะ เช่น .header { ... } |
| ความยืดหยุ่น | ยืดหยุ่นสูง ใช้หลายคลาสรวมกันเพื่อกำหนดสไตล์ได้หลากหลาย | มีข้อจำกัดขึ้นกับโครงสร้างของคลาสที่ออกแบบไว้ |
| ความง่ายในการใช้งาน | ใช้ได้ทันที ไม่ต้องเขียน CSS ใหม่ | ต้องเขียน CSS ใหม่หรือเพิ่มในไฟล์ CSS |
| ความเร็วในการพัฒนา | รวดเร็ว เพราะมีชุดคลาสพร้อมใช้งาน เช่น Tailwind CSS | ช้ากว่า เพราะต้องสร้างและกำหนดสไตล์เฉพาะเอง |
| ความสามารถในการบำรุงรักษา | อาจซับซ้อนถ้ามีคลาสจำนวนมากใน HTML | จัดการง่ายขึ้นในบางกรณี โดยเฉพาะถ้าใช้ชื่อคลาสเชิงความหมาย |
| การปรับแต่ง | จำกัดการปรับแต่ง (ขึ้นกับ framework ที่ใช้) | ปรับแต่งได้อย่างอิสระ |
| ไฟล์ CSS | ไฟล์ CSS มักมีขนาดเล็กกว่า เพราะใช้ utility class ซ้ำ | ไฟล์ CSS ใหญ่กว่าในโปรเจกต์ที่มีขนาดใหญ่ |

# คำอธิบาย

## Utility Class

- เป็นคลาสที่มุ่งเน้นไปที่ฟังก์ชันเดียว เช่น การจัดตำแหน่งข้อความ (text-left, text-right), การเพิ่มระยะห่าง (mt-4, p-2), หรือการเปลี่ยนสี (bg-blue-500).

- มีประโยชน์มากในงานที่ต้องการความรวดเร็วในการออกแบบ และเหมาะกับโครงการขนาดเล็กหรือทีมงานที่ต้องการมาตรฐานเดียวกัน.

**คัวอย่าง**

<div class="text-center bg-gray-200 p-4">

Utility Class Example

</div>

## CSS ปกติ (Custom CSS)

ช่วยให้ออกแบบสไตล์ได้อย่างเจาะจง และให้ความสำคัญกับการตั้งชื่อคลาสที่มีความหมาย ทำให้โค้ดอ่านง่ายขึ้น.

เหมาะสำหรับโครงการขนาดใหญ่หรือโปรเจกต์ที่ต้องการความแตกต่างและความยืดหยุ่นสูง.

**คัวอย่าง**

**CSS**

.custom-header {

text-align: center; background-color: #f3f3f3; padding: 20px;

}

**HTML**

<div class="custom-header">

Custom CSS Example

</div>