ชื่อ........**จักรพงษ์ คชเผือก**............................................. รหัสนิสิต......**65160013**......... กลุ่ม.........**2**.................

คณะ...........**วิทยาการสารสนเทศ**................................... อาจารย์ผู้สอน.......**ประจักษ์ จิตเงินมะดัน**.......................

**ปฏิบัติการที่ 14**

**Flexbox and Position**

**วัตถุประสงค์**

1. เพื่อศึกษาเนื้อหาความรู้เกี่ยวกับการออกแบบ Web Application ด้วย CSS
2. เพื่อฝึกทักษะการออกแบบ Web Application ด้วย Flexbox and Position

**1. แหล่งเรียนรู้เนื้อหาสำคัญ**

ให้นิสิตศึกษาข้อมูลและเนื้อหาของ HTML, CSS, และ JavaScript จากบทความข้างล่างนี้

1. Hyper Text Markup Language (HTML): <https://www.w3schools.com/html/default.asp>
2. Cascading Style Sheets (CSS): <https://www.w3schools.com/css/default.asp>
3. JavaScript: <https://www.w3schools.com/js/default.asp>
4. Reference: <https://supersimple.dev/courses/html-css-course>

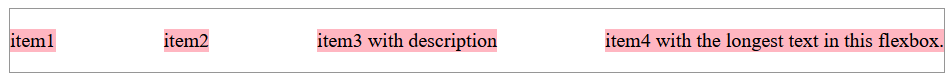
**2. Flexbox**

Flexbox เป็นเครื่องมือที่ใช้ในการจัดการ Layout ของเว็บไซต์เช่นเดียวกันกับ Grid แต่จะมีคุณสมบัติที่เหมือนและแตกต่างกันในบางเรื่อง เช่น ความยืดหยุ่นของการวาง element ในบรรทัดเดียวกัน หรือการสร้างเว็บไซต์ที่มีความ responsive ตามอุปกรณ์ที่แสดงผล โดยความกว้างของ element จะเป็นไปตามความกว้างของตัวมันเองที่ถูกตั้งค่าไว้แต่แรก ไม่ขึ้นอยู่กับตำแหน่งของมันใน Layout นั้น ๆ นั่นหมายความว่า ถ้ามีการสลับตำแหน่งของ element ใน layout flexbox ตัว element นั้น ๆ จะยังคงความกว้างเดิมเอาไว้ ไม่ว่าจะอยุ่ตำแหน่งไหนใน layout flexbox นั้น แต่ใน Grid นั้น ความกว้างของ element จะเป็นไปตาม layout ของ grid ที่ตัว element นั้นอยู่

หมายเหตุ การสร้าง flexbox จะใช้ 2 ขั้นตอนคือ การประกาศให้ display เป็น flex และการกำหนด flex-direction เป็น row

สามารถศึกษาเพิ่มเติมได้ที่ <https://www.w3schools.com/css/css3_flexbox.asp>

ตัวอย่าง การสร้าง Flexbox ที่มีความสูง 50px และ มีเส้นกรอบสีเทา โดยที่ element จะเรียงห่างเท่า ๆ กัน



ตัวอย่างโค้ด HTML

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="lab14\_flexbox.css">

</head>

<body>

<div class="flexbox">

<div class="lightpink">item1</div>

<div class="lightpink">item2</div>

<div class="lightpink">item3 with description</div>

<div class="lightpink">item4 with the longest text in this flexbox.</div>

</div>

</body>

</html>

ตัวอย่างโค้ด CSS

.flexbox {

display: flex;

flex-direction: row;

justify-content: space-between; /\*or space-around, center, flex-start, flex-end \*/

border-width: 1px;

border-style: solid;

border-color: rgb(150, 150, 150);

height: 50px;

align-items: center;

}

.lightpink {

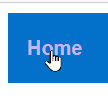
background-color: lightpink;

}

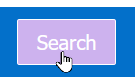
1. ให้นิสิตสร้างไฟล์ lab14\_flexbox\_excercise.html
2. เขียนโค้ดให้มีลักษณะดังนี้



หมายเหตุ

1. เมื่อชี้เมาส์ที่ Home จะเปลี่ยนสีและเมาส์เป็นมือชี้ 

2. เมื่อคลิกที่ Search คำว่า Search จะหายไป 

3. เมื่อชี้เมาส์ที่ปุ่ม Search จะเปลี่ยนสี และเป็นรูปมือชี้ 

**ผลลัพธ์ (โค้ด)**

**HTML:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <link rel="stylesheet" href="/lab14\_flexbox\_excercise.css">

    <link href="https://fonts.googleapis.com/css2?family=Sarabun:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800&display=swap" rel="stylesheet">

    <title> Flex Box </title>

</head>

<body>

<div class="flexbox">

    <div class="home">

        Home

    </div>

    <div class="search-bar">

        <input type="text" placeholder="Search">

    </div>

    <div class="search-btn">

        <button>Search</button>

    </div>

</div>

</body>

</html>

**CSS:**

\* {

    font-family: "Sarabun", sans-serif;

}

.flexbox {

display: flex;

background-color: rgba(3,112,202,255);

flex-direction: row;

align-items: center;

border-style: none;

height: 50px;

width: fit-content;

padding: 0px 15px;

}

.home {

    color: white;

    font-size: large;

    font-weight: bolder;

}

.home:hover {

    color: gray;

    cursor: pointer;

}

.search-bar {

    & input {

        width: 300px;

        height: 25px;

        border: 0px solid;

        border-radius: 50px;

        margin: 0px 18px;

        padding-left: 10px;

        outline: none;

    }

    & input:focus::placeholder {

        color: transparent;

    }

}

.search-btn {

    & button {

        color: white;

        background-color: rgba(3,112,202,255);

        border: 1px white solid;

        padding: 3px 15px;

    }

    & button:hover {

        cursor: pointer;

        background-color: mediumpurple;

    }

}



**3. Position**

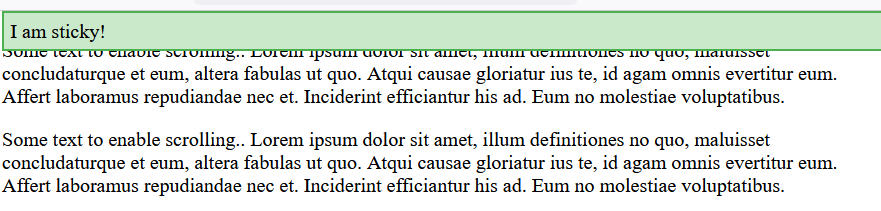
ใน CSS เราสามารถกำหนดตำแหน่งของ element ต่าง ๆ ได้โดยเป็นการเพิ่มมิติของการแสดงผลบนหน้าจอให้กับ element ที่ต้องการ เช่น มีการซ้อนทับกันของ elements ในการแสดงตัวแลขจำนวนสินค้าในตะกร้าสินค้า เป็นต้น

ตำแหน่งของ element ใน CSS จะมีอยู่ 5 ตัว คือ

1. static เป็นตำแหน่งปกติของ element
2. relative เป็นตำแหน่งแสดงผลที่อ้างอิงตำแหน่งเริ่มต้น
3. fixed เป็นตำแหน่งแสดงผลที่ตรึงอยู่ที่ตำแหน่งเดิมตลอด ตำแหน่งนี้จะคงตำแหน่งบนหน้าจอของอุปกรณ์ที่ใช้งาน
4. absolute เป็นตำแหน่งแสดงผลที่ตรึงอยู่ที่ตำแหน่งเดิม ขึ้นอยู่กับหมุดปักที่ใกล้ที่สุด เช่น ถ้าปักหมุดที่ page ตำแหน่งของ element ก็จะเป็นไปตาม page นั้น ๆ
5. sticky เป็นตำแหน่งที่เปลี่ยนไปตามการ scroll หน้าจอ

เมื่อกำหนด position แล้ว ก็สามารถใช้คุณสมบัติ top, right, bottom, left ในการกำหนดตำแหน่งของ element ที่ต้องการได้ ยกเว้น static จะใช้ไม่ได้

ตัวอย่างการใช้งาน sticky



**Position: sticky**

<!DOCTYPE html>

<html>

<head>

<style>

div.sticky {

position: -webkit-sticky; /\* for Safari \*/

position: sticky;

top: 0;

padding: 5px;

background-color: #cae8ca;

border: 2px solid #4CAF50;

}

</style>

</head>

<body>

<p>Try to <b>scroll</b> inside this frame to understand how sticky positioning works.</p>

<div class="sticky">I am sticky!</div>

<div style="padding-bottom:2000px">

<p>In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.</p>

<p>Scroll back up to remove the stickyness.</p>

<p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>

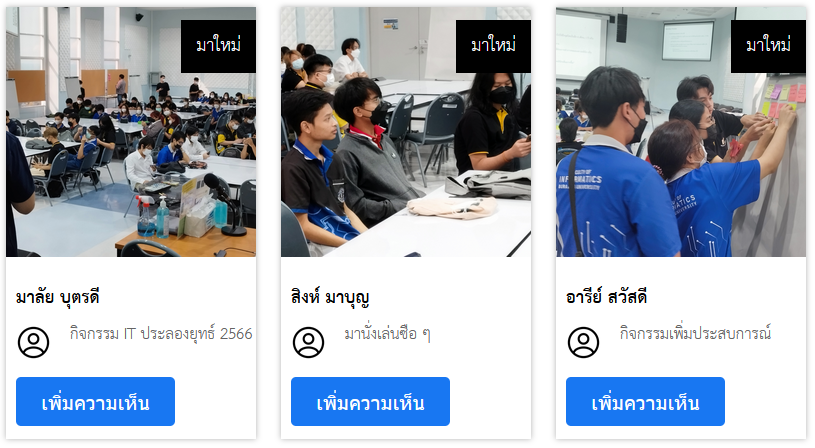
<p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>

</div>

</body>

</html>

จงสร้างหน้าเว็บที่มีรูปแบบต่อไปนี้



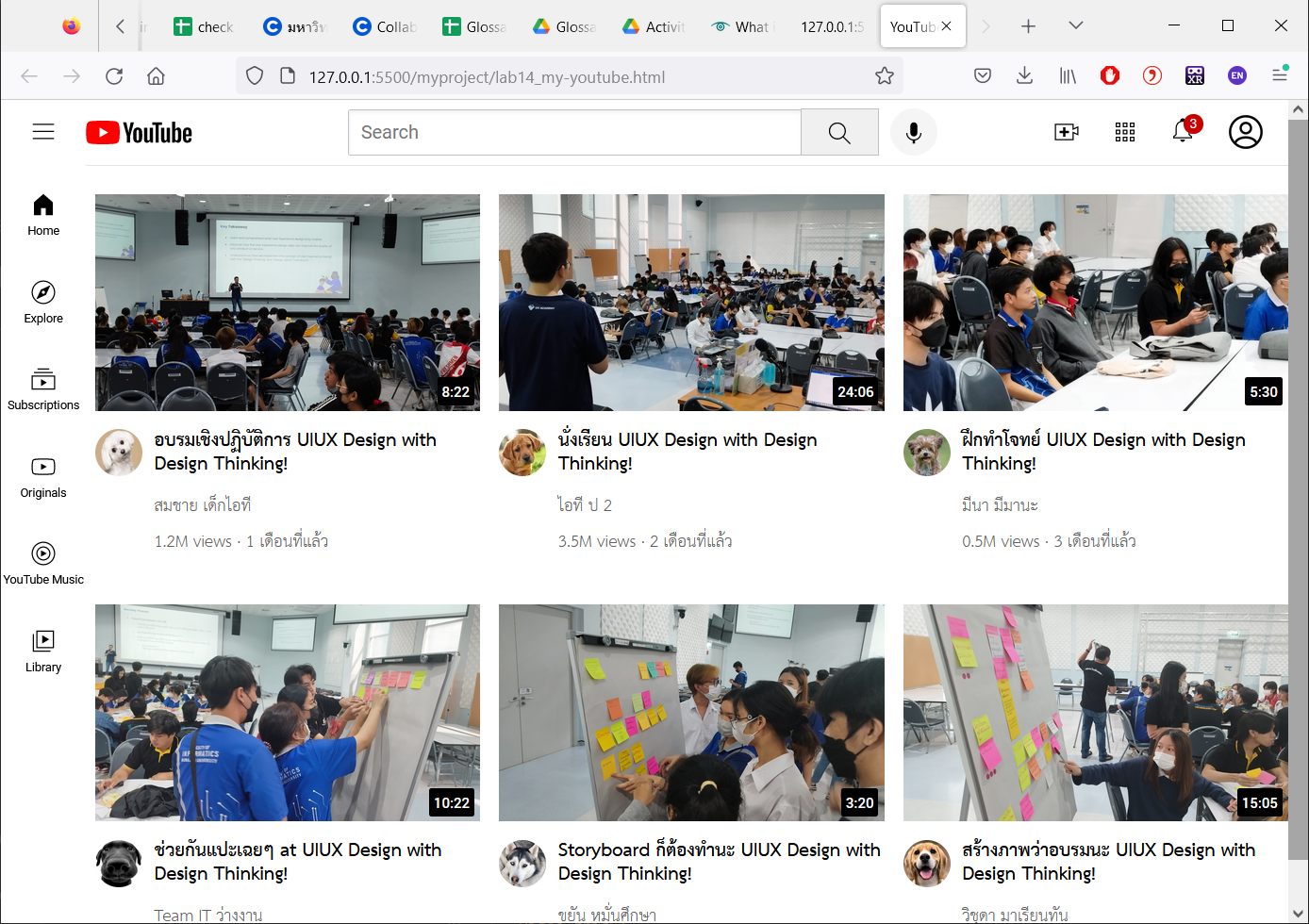
หมายเหตุ

1. นิสิตสามารถใช้การซ้อนกันของ position ได้ โดยที่ให้กรอบนอกเป็น relative และกรอบด้านในเป็น absolute เพื่อให้กล่องสีดำนั้นอยู่ในตำแหน่งที่ติดกับรูปตลอดเวลา
2. นิสิตสามารถสร้าง div ที่มี position: relative อยู่ด้านนอก
3. แล้วทำการสร้าง div ที่มี position: absolute อยู่ด้านใน พร้อมทั้งกำหนดค่า top, right ให้เหมาะสม

**ผลลัพธ์ (โค้ด)**

**5. Try it yourself!**

ให้นิสิตเพิ่มเติมหน้าเว็บ my-youtube.html ให้มีลักษณะ ดังนี้



หมายเหตุ

1. สามารถสร้างแต่ละส่วนแยกกันได้ เช่น header, sidebar, content โดยให้มีไฟล์ CSS ที่แยกกันได้ เพื่อความสะดวกในการจัดการในอนาคต

2. สามารถดูตัวอย่าง CSS ได้ที่ด้านล่าง

**ผลลัพธ์ (โค้ด) HTML**

Reference: <https://supersimple.dev/courses/html-css-course>

**ตัวอย่าง CSS ที่ใช้ในการสร้างเว็บไซต์ Youtube**

General.css

p {

font-family: 'TH Sarabun New', Sarabun, Tahoma;

font-size: 28px;

margin-top: 0;

margin-bottom: 0;

}

body {

margin: 0;

padding-top: 80px;

padding-left: 80px;

}

Header.css

.header {

height: 55px;

display: flex;

flex-direction: row;

justify-content: space-between;

position: fixed;

top: 0;

left: 0;

right: 0;

z-index: 100;

background-color: white;

border-bottom-width: 1px;

border-bottom-style: solid;

border-bottom-color: rgb(228, 228, 228);

}

.left-section {

display: flex;

align-items: center;

}

.hamburger-menu {

height: 24px;

margin-left: 24px;

margin-right: 24px;

}

.youtube-logo {

height: 20px;

}

.middle-section {

flex: 1;

margin-left: 70px;

margin-right: 35px;

max-width: 500px;

display: flex;

align-items: center;

}

.search-bar {

flex: 1;

height: 36px;

padding-left: 10px;

font-size: 22px;

border-width: 1px;

border-style: solid;

border-color: rgb(192, 192, 192);

border-radius: 2px;

box-shadow: inset 1px 2px 3px rgba(0, 0, 0, 0.05);

width: 0;

}

.search-bar::placeholder {

font-family: Roboto, Arial;

font-size: 16px;

}

.search-bar:focus::placeholder{ /\*delete placeholder text \*/

color: transparent;

}

.search-button {

height: 40px;

width: 66px;

background-color: rgb(240, 240, 240);

border-width: 1px;

border-style: solid;

border-color: rgb(192, 192, 192);

margin-left: -1px;

margin-right: 10px;

}

.search-icon {

height: 25px;

margin-top: 4px;

}

.voice-search-button {

height: 40px;

width: 40px;

border-radius: 20px;

border: none;

background-color: rgb(245, 245, 245);

}

.voice-search-icon {

height: 24px;

margin-top: 4px;

}

.right-section {

width: 180px;

margin-right: 20px;

display: flex;

align-items: center;

justify-content: space-between;

flex-shrink: 0;

}

.upload-icon {

height: 24px;

}

.youtube-apps-icon {

height: 24px;

}

.notifications-icon {

height: 24px;

}

.current-user-picture {

height: 32px;

border-radius: 16px;

}

.notifications-icon-container {

position: relative;

}

.notifications-count {

position: absolute;

top: -2px;

right: -5px;

background-color: rgb(200, 0, 0);

color: white;

font-family: Roboto, Arial;

font-size: 11px;

padding-left: 5px;

padding-right: 5px;

padding-top: 2px;

padding-bottom: 2px;

border-radius: 10px;

}

Sidebar.css

.sidebar {

position: fixed;

left: 0;

bottom: 0;

top: 55px;

background-color: white;

width: 72px;

z-index: 200;

padding-top: 5px;

}

.sidebar-link {

height: 74px;

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

cursor: pointer;

}

.sidebar-link:hover {

background-color: rgb(235, 235, 235);

}

.sidebar-link img {

height: 24px;

margin-bottom: 4px;

}

.sidebar-link div {

font-family: Roboto, Arial;

font-size: 10px;

}

Video.css

.thumbnail {

width: 100%;

}

.video-title {

margin-top: 0;

font-size: 22px;

font-weight: bold;

line-height: 20px;

margin-bottom: 12px;

}

.video-info-grid {

display: grid;

grid-template-columns: 50px 1fr;

}

.profile-picture {

width: 40px;

border-radius: 50%;

}

.thumbnail-row {

margin-bottom: 12px;

position: relative;

}

.video-author,

.video-stats {

font-size: 20px;

color: rgb(96, 96, 96);

}

.video-author {

margin-bottom: 4px;

}

.video-grid {

display: grid;

grid-template-columns: 1fr 1fr 1fr;

column-gap: 16px;

row-gap: 40px;

}

.video-time {

font-family: Roboto, Arial;

font-size: 12px;

font-weight: 500;

padding-left: 4px;

padding-right: 4px;

padding-top: 4px;

padding-bottom: 4px;

border-radius: 2px;

background-color: black;

color: white;

position: absolute;

bottom: 8px;

right: 5px;

}