

ใบงาน: สร้างแอปพลิเคชัน To-Do List

วิชา: การพัฒนาเว็บแอปพลิเคชันเบื้องต้น หัวข้อ: การใช้งาน TypeScript ร่วมกับ HTML และ CSS

วัตถุประสงค์การเรียนรู้

เมื่อทำใบงานนี้เสร็จสิ้น นักเรียนจะสามารถ:

1. กำหนดโครงสร้างข้อมูลด้วย interface ใน TypeScript
2. เข้าถึงและจัดการส่วนประกอบของหน้าเว็บ (DOM Manipulation)
3. สร้างและจัดการเหตุการณ์ (Event Handling) เช่น การคลิก, การส่งฟอร์ม
4. ประยุกต์ใช้ความรู้ HTML, CSS และ TypeScript เพื่อสร้างเว็บแอปพลิเคชันแบบโต้ตอบได้ (Interactive)

คำชี้แจง

ให้นักศึกษาเปิดไฟล์โค้ด todo-app-challenge ที่ให้มา และเขียนโค้ดภาษา TypeScript ในส่วน `<script type="module">` ให้สมบูรณ์ตามคำแนะนำใน comment ที่ขึ้นต้นด้วย `// TODO:` เพื่อสร้างแอปพลิเคชัน To-Do List ให้ทำงานได้ครบถ้วนตามที่กำหนด

ฟังก์ชันการทำงานที่ต้องมี:

- **เพิ่มรายการ (Add):** ผู้ใช้สามารถพิมพ์ข้อความในช่อง input และกดปุ่ม "เพิ่ม" เพื่อสร้างรายการใหม่ได้
- **ลบรายการ (Delete):** ผู้ใช้สามารถลบรายการที่ไม่ต้องการออกจากลิสต์ได้
- **เปลี่ยนสถานะ (Complete/Incomplete):** ผู้ใช้สามารถติ๊กที่ช่อง checkbox เพื่อบอกว่ารายการนั้น "ทำเสร็จแล้ว" (จะมีขีดฆ่าที่ข้อความ) และสามารถติ๊กออกเพื่อเปลี่ยนสถานะกลับได้

ขั้นตอนการทำ

ให้นักศึกษาเติมโค้ดในส่วนต่างๆ ตามลำดับดังนี้

ส่วนที่ 1: กำหนด Interface และ State

- สร้าง interface ชื่อ Task ให้มี property ตามที่ระบุใน comment

- แก้ไขการประกาศตัวแปร tasks ให้เป็น Array ของ Task

ส่วนที่ 2: อ้างอิง HTML Elements

- ส่วนนี้ได้เตรียมไว้ให้แล้ว ให้นักศึกษาทำความเข้าใจว่าแต่ละตัวแปรอ้างอิงถึงส่วนไหนของ HTML

ส่วนที่ 3: สร้างฟังก์ชันสำหรับการจัดการข้อมูล (Logic)

- **addTask(text: string):** เขียนโค้ดเพื่อสร้าง object Task ใหม่ และเพิ่มเข้าไปใน tasks array
- **toggleTaskCompleted(id: number):** เขียนโค้ดเพื่อค้นหา task ตาม id และสลับค่า isCompleted
- **deleteTask(id: number):** เขียนโค้ดเพื่อลบ task ที่มี id ที่ตรงกันออกจาก tasks array

ส่วนที่ 4: สร้างฟังก์ชันสำหรับแสดงผล (Render)

- **renderTasks():** เป็นส่วนที่สำคัญที่สุด ให้นักศึกษาเขียนโค้ดเพื่อ:
 1. ล้างข้อมูลเก่าใน `<ul id="task-list">`
 2. วนลูป tasks array เพื่อสร้าง `` element สำหรับแต่ละ task
 3. ใส่ checkbox และปุ่ม delete พร้อมกำหนด data-id ให้ถูกต้อง
 4. แสดงผลให้ถูกต้องตามสถานะ isCompleted (ติ๊ก checkbox, เพิ่ม class completed)
 5. นำ `` ที่สร้างเสร็จแล้วไปแสดงในหน้าเว็บ

ส่วนที่ 5: เพิ่ม Event Listeners

- **taskForm (submit event):** เขียนโค้ดเพื่อรับค่าจาก input, เรียกใช้ฟังก์ชัน addTask() และ renderTasks() จากนั้นล้างค่าใน input
- **taskList (click event):** ใช้เทคนิค Event Delegation เพื่อตรวจจับการคลิกที่ checkbox และปุ่ม delete จากนั้นเรียกใช้ฟังก์ชัน toggleTaskCompleted() หรือ deleteTask() ตามลำดับ และเรียก renderTasks() เพื่ออัปเดตหน้าจอ
-

โจทย์ท้าทายเพิ่มเติม (Bonus)

สำหรับนักศึกษาที่ทำส่วนหลักเสร็จแล้ว สามารถลองทำโจทย์เพิ่มเติมต่อไปนี้:

- **บันทึกข้อมูล:** ใช้ localStorage เพื่อบันทึกรายการ tasks ทำให้เมื่อปิดหรือรีเฟรชหน้าเว็บ ข้อมูลจะยังคงอยู่
- **ระบบกรอง:** เพิ่มปุ่มเพื่อกรองรายการ (เช่น "ทั้งหมด", "ที่ยังไม่เสร็จ", "ที่เสร็จแล้ว")

การส่งงาน

ให้นักศึกษาส่งไฟล์ index.html ที่เขียนโค้ด TypeScript สมบูรณ์แล้ว

ใบงาน: สร้างแอปสุ่มคำคม

วิชา: การพัฒนาเว็บแอปพลิเคชันเบื้องต้น หัวข้อ: การจัดการข้อมูลใน Array และ DOM Manipulation พื้นฐาน

วัตถุประสงค์การเรียนรู้

1. จัดเก็บและเข้าถึงข้อมูลใน Array of Objects
2. ใช้ฟังก์ชันทางคณิตศาสตร์ (Math.random) เพื่อสุ่มข้อมูล
3. อัปเดตเนื้อหาบนหน้าเว็บ (DOM) เพื่อตอบสนองต่อการกระทำของผู้ใช้
4. ฝึกฝนการสร้างแอปพลิเคชันขนาดเล็กให้เสร็จสมบูรณ์

คำชี้แจง

ให้นักศึกษาสร้างแอปพลิเคชันสำหรับแสดงคำคมแบบสุ่ม โดยจะมีปุ่ม "สุ่มคำคมใหม่" เมื่อผู้ใช้กดปุ่มนี้ โปรแกรมจะสุ่มเลือกคำคมจากชุดข้อมูลที่เราเตรียมไว้มาแสดงผลบนหน้าจอ

ฟังก์ชันการทำงานที่ต้องมี:

- แสดงคำคมเริ่มต้น: เมื่อเปิดหน้าเว็บครั้งแรก ให้แสดงคำคมขึ้นมา 1 คำคม
- สุ่มคำคมใหม่: เมื่อผู้ใช้กดปุ่ม ให้โปรแกรมสุ่มคำคมใหม่ที่ไม่ซ้ำกับคำคมเดิมที่แสดงอยู่มาแสดงผลแทนที่

ขั้นตอนการทำ

ส่วนที่ 1: กำหนดโครงสร้างข้อมูล

- สร้าง interface ชื่อ Quote ประกอบด้วย:
 - text: string (ตัวคำคม)
 - author: string (ชื่อผู้กล่าว)
- สร้าง Array ของ Quote ขึ้นมา โดยใส่คำคมที่นักศึกษาสนใจอย่างน้อย 5-7 คำคม

ส่วนที่ 2: อ้างอิง HTML Elements

- สร้าง HTML ที่มีพื้นที่สำหรับแสดงข้อความคำคม (<blockquote> หรือ <p>)

- สร้างพื้นที่สำหรับแสดงชื่อผู้กล่าว
- สร้างปุ่มสำหรับกดเพื่อสุ่มคำคมใหม่
- เขียนโค้ด TypeScript เพื่ออ้างอิงไปยัง Element เหล่านี้

ส่วนที่ 3: สร้างฟังก์ชันหลัก

- **getRandomQuote():** ฟังก์ชันสำหรับสุ่มคำคมจาก Array
 - ใช้ Math.random() ร่วมกับความยาวของ Array เพื่อสุ่ม index
 - คืนค่าเป็น Object Quote ที่สุ่มได้
- **displayQuote(quote: Quote):** ฟังก์ชันสำหรับนำข้อมูล Quote ที่ได้ไปแสดงผลบนหน้าเว็บ
 - นำ quote.text ไปใส่ใน Element ที่แสดงคำคม
 - นำ quote.author ไปใส่ใน Element ที่แสดงชื่อผู้กล่าว

ส่วนที่ 4: เพิ่ม Event Listener และการทำงานเริ่มต้น

- เขียน Event Listener ให้กับปุ่ม "สุ่มคำคมใหม่"
- เมื่อมีการคลิก ให้เรียกใช้ getRandomQuote() แล้วตามด้วย displayQuote()
 - **ข้อควรระวัง:** เขียน Logic เพิ่มเติมน้อยเพื่อป้องกันการสุ่มได้คำคมเดิมซ้ำกับที่แสดงอยู่
- เมื่อเปิดหน้าเว็บครั้งแรก ให้เรียกฟังก์ชันเพื่อสุ่มและแสดงคำคมเริ่มต้นทันที

โจทย์ท้าทายเพิ่มเติม (Bonus)

- เพิ่มปุ่ม "Tweet" หรือ "Copy" เพื่อให้ผู้ใช้สามารถแชร์หรือคัดลอกคำคมที่ชอบได้
- เพิ่ม Animation (เช่น Fade in/out) เวลาที่คำคมเปลี่ยน