**เฉลยแบบทดสอบย่อยวิชา ENGCE306 Web Programming**

**ส่วนที่ 1: ความเข้าใจพื้นฐาน**

**ข้อ 1. อธิบายส่วนประกอบหลักของโครงสร้างเอกสาร HTML5 พื้นฐาน มาอย่างน้อย 3 ส่วน พร้อมบอกหน้าที่ของแต่ละส่วนโดยสังเขป (5 คะแนน)**

* **<!DOCTYPE html>:** เป็นการประกาศ (Declaration) ที่ต้องอยู่บรรทัดแรกสุดของไฟล์ เพื่อบอกให้เว็บเบราว์เซอร์ทราบว่าเอกสารนี้เป็นไฟล์ HTML5
* **<html>:** เป็นแท็กราก (Root Element) ที่ห่อหุ้มเนื้อหาทั้งหมดของหน้าเว็บ
* **<head>:** เป็นส่วนที่เก็บข้อมูลเกี่ยวกับหน้าเว็บ (Metadata) ซึ่งจะไม่แสดงผลบนหน้าจอโดยตรง เช่น ชื่อเรื่องของเว็บ (<title>), การเชื่อมต่อไฟล์ CSS, หรือการกำหนดชุดอักขระ (<meta charset="UTF-8">)
* **<body>:** เป็นส่วนที่เก็บเนื้อหาทั้งหมดที่จะแสดงผลให้ผู้ใช้เห็นบนหน้าเว็บ เช่น ข้อความ, รูปภาพ, ลิงก์, ตาราง เป็นต้น

**ข้อ 2. จงบอกความแตกต่างที่สำคัญระหว่าง ID Selector และ Class Selector ใน CSS ในแง่ของการนำไปใช้งานกับองค์ประกอบ HTML (5 คะแนน)**

* **ID Selector (ใช้ #):** ถูกออกแบบมาเพื่อเลือก **Element เพียงตัวเดียวที่ไม่ซ้ำกัน** ในหน้าเว็บนั้นๆ (Unique) เหมาะสำหรับใช้กับองค์ประกอบหลักที่มีเพียงชิ้นเดียว เช่น #main-header, #footer
* **Class Selector (ใช้ .):** ถูกออกแบบมาเพื่อเลือก **กลุ่มของ Elements** ที่มีลักษณะหรือต้องการสไตล์เดียวกัน สามารถนำไปใช้ซ้ำกับหลายๆ Element ได้ และหนึ่ง Element สามารถมีได้หลาย Class

**ข้อ 3. ในภาษา JavaScript (มาตรฐาน ES6) การประกาศตัวแปรด้วยคีย์เวิร์ด let และ const มีความแตกต่างกันอย่างไรในการกำหนดและเปลี่ยนแปลงค่า? (5 คะแนน)**

* **let:** ใช้สำหรับประกาศตัวแปรที่สามารถ **เปลี่ยนแปลงค่า (Re-assign) ได้** ในภายหลัง
* **const (ย่อมาจาก constant):** ใช้สำหรับประกาศค่าคงที่ ซึ่งหมายความว่าเมื่อกำหนดค่าไปแล้ว จะไม่สามารถ **เปลี่ยนแปลงค่า (Re-assign) ได้อีก** และต้องกำหนดค่าเริ่มต้นให้ทันทีที่ประกาศ

**ข้อ 4. TypeScript คืออะไร และมีประโยชน์หลักอย่างไรเมื่อเทียบกับการใช้ JavaScript เพียงอย่างเดียวในการพัฒนาเว็บแอปพลิเคชัน? (5 คะแนน)**

TypeScript คือ Superset ของ JavaScript ที่เพิ่มฟีเจอร์ **Static Typing** เข้ามา ประโยชน์หลักคือ **การตรวจจับข้อผิดพลาดเรื่องชนิดของข้อมูล (Type Error) ได้ตั้งแต่ขั้นตอนการพัฒนา (Compile-time)** ก่อนที่โค้ดจะถูกรันจริง ทำให้โค้ดมีความน่าเชื่อถือสูง ลดโอกาสเกิด Runtime Error และช่วยให้การพัฒนาโปรเจกต์ขนาดใหญ่หรือการทำงานเป็นทีมทำได้ง่ายและปลอดภัยยิ่งขึ้น

**ส่วนที่ 2: การอ่านโค้ดและคิดวิเคราะห์**

**ข้อ 5. การใช้ Semantic HTML มีข้อดีกว่าการใช้แท็ก <div> เพียงอย่างเดียวอย่างไร? จงอธิบายในมุมมองของ SEO และ Accessibility (15 คะแนน)**

การใช้ Semantic HTML เช่น <nav>, <main>, <article> ดีกว่าการใช้ <div> ล้วนๆ เพราะแท็กเหล่านี้มีความหมายในตัวเอง ซึ่งส่งผลดีดังนี้:

* **ในมุมมองของ SEO:** Search Engines (เช่น Google) สามารถเข้าใจโครงสร้างและความสำคัญของเนื้อหาในหน้าเว็บได้ดีขึ้น เช่น รู้ว่าส่วนไหนคือเมนูนำทาง (<nav>) ส่วนไหนคือเนื้อหาหลัก (<main>) ซึ่งช่วยให้การจัดอันดับผลการค้นหา (Ranking) มีประสิทธิภาพมากขึ้น
* **ในมุมมองของ Accessibility:** เครื่องมือช่วยอ่านสำหรับผู้พิการทางสายตา (Screen Reader) สามารถตีความและสื่อสารโครงสร้างของหน้าเว็บให้ผู้ใช้ฟังได้เข้าใจง่าย เช่น สามารถบอกผู้ใช้ได้ว่า "นี่คือส่วนนำทาง" และผู้ใช้สามารถสั่งให้ข้ามไปอ่านเนื้อหาหลักได้ทันที ทำให้ผู้พิการเข้าถึงข้อมูลได้สะดวกยิ่งขึ้น

**ข้อ 6. จากโค้ด HTML และ CSS ที่กำหนดให้ จงอธิบายว่าแท็ก <p> ที่มีข้อความว่า "This paragraph is special." จะถูกแสดงผลบนหน้าจออย่างไรบ้าง? (15 คะแนน)**

แท็ก <p> ดังกล่าวจะถูกแสดงผลโดยได้รับสไตล์จาก **3 rules** มาผสมกัน ดังนี้:

1. **จาก p.special:** เนื่องจาก Selector นี้มีความเฉพาะเจาะจงสูง จะทำให้ข้อความมี **สีแดง (color: red)** และเป็น **ตัวหนา (font-weight: bold)**
2. **จาก .highlight:** จะทำให้พื้นหลังของย่อหน้าเป็น **สีเหลือง (background-color: yellow)**
3. **จาก #main-content:** จะได้รับการสืบทอด (Inherit) คุณสมบัติฟอนต์ ทำให้ฟอนต์ของข้อความเป็น **Arial, sans-serif**

**สรุป:** ย่อหน้า "This paragraph is special." จะแสดงผลเป็น **ตัวอักษรหนาสีแดง บนพื้นหลังสีเหลือง และใช้ฟอนต์ Arial**

**ข้อ 7. จากโค้ด JavaScript จงอธิบายการทำงานทีละขั้นตอน และอธิบายว่า "Closure" มีบทบาทสำคัญอย่างไร? (20 คะแนน)**

**การทำงานทีละขั้นตอน:**

1. ฟังก์ชัน setupCounter ถูกเรียกใช้งาน
2. มีการประกาศตัวแปร count และกำหนดค่าเริ่มต้นเป็น 0 ภายใน Scope ของ setupCounter
3. โค้ดทำการเลือกปุ่ม (<button>) ที่มี id="myButton" มาเก็บไว้ในตัวแปร counterButton
4. มีการเพิ่ม Event Listener ให้กับปุ่ม เพื่อดักจับเหตุการณ์ click โดยเมื่อเกิดการคลิก จะไปเรียกใช้ฟังก์ชันที่อยู่ข้างใน (Anonymous Function)
5. ฟังก์ชัน setupCounter ทำงานเสร็จสิ้น

**บทบาทของ Closure:** โดยปกติแล้ว เมื่อฟังก์ชัน setupCounter ทำงานเสร็จ ตัวแปร count ที่อยู่ข้างในควรจะถูกลบไปจากหน่วยความจำ แต่ **Closure** คือกลไกที่ทำให้ฟังก์ชันลูก (Anonymous Function ที่รอการคลิก) ยังคง **"จดจำ"** และ **"เข้าถึง"** ตัวแปรและสภาพแวดล้อม (Scope) ของฟังก์ชันแม่ที่สร้างมันขึ้นมาได้

ดังนั้น แม้ setupCounter จะจบไปแล้ว แต่ตัวแปร count ยังไม่ถูกทำลาย และทุกครั้งที่ผู้ใช้คลิกปุ่ม ฟังก์ชันลูกจะเข้าไปอัปเดตค่า count ตัวเดิมที่ถูกเก็บไว้ใน Closure ทำให้สามารถนับเลขต่อเนื่องได้อย่างถูกต้อง

**ข้อ 8. จากโค้ด TypeScript จงตอบคำถาม (20 คะแนน)**

1. **อธิบายความสัมพันธ์:**
   * **class Car extends Vehicle:** Car เป็นคลาสลูกที่ **สืบทอด (Inherits)** คุณสมบัติและเมธอดทั้งหมดมาจากคลาสแม่ Vehicle (เช่น brand และ move() เวอร์ชั่นดั้งเดิม)
   * **class Car implements Movable:** Car ได้ทำ **"สัญญา" (Implements)** ว่าจะมีโครงสร้างตามที่ interface Movable กำหนดไว้ ซึ่งหมายความว่าคลาส Car จะต้องมี property speed และเมธอด move() เป็นของตัวเอง
2. **ผลลัพธ์ที่แสดงใน Console คือ:** The Toyota car is moving at 60 km/h.

**เพราะเหตุใด:** เพราะคลาส Car ได้ทำการ **Override (เขียนทับ)** เมธอด move() ที่สืบทอดมาจาก Vehicle ด้วยเมธอด move() เวอร์ชันใหม่ของตัวเอง ดังนั้นเมื่อมีการเรียกใช้ myCar.move() โปรแกรมจะเรียกใช้เมธอดที่ถูกเขียนทับในคลาส Car เสมอ

**ข้อ 9. จากโค้ด JavaScript ด้านล่าง จงอธิบายการทำงานทีละขั้นตอนว่าทำไมผลลัพธ์สุดท้ายของ activeUserNames จึงเป็น ['ALICE', 'CHARLIE']? (15 คะแนน)**

การทำงานแบ่งเป็น 2 ขั้นตอนหลักดังนี้:

1. **การกรองข้อมูลด้วย for...of และ if:**
   * โค้ดสร้าง Array ว่างชื่อ activeUsers ขึ้นมาก่อน
   * for...of loop จะวนเข้าไปดูข้อมูลผู้ใช้ทีละคนใน userData
   * เงื่อนไข if (user.age >= 18 && user.isActive) จะทำการตรวจสอบว่าผู้ใช้คนนั้น **มีอายุ 18 ปีขึ้นไป และ สถานะต้องเป็น true** พร้อมกันทั้งสองเงื่อนไข
   * **Alice** และ **Charlie** เป็นสองคนเดียวที่ผ่านเงื่อนไขนี้ จึงถูก push เข้าไปเก็บใน Array activeUsers
2. **การแปลงข้อมูลด้วย .map():**
   * เมธอด .map() ถูกเรียกใช้กับ Array activeUsers (ซึ่งตอนนี้มีข้อมูลของ Alice และ Charlie)
   * .map() จะสร้าง Array ใหม่ขึ้นมา โดยนำข้อมูลแต่ละชิ้นใน activeUsers มาแปลงค่าตามฟังก์ชันที่กำหนด คือ user => user.name.toUpperCase() ซึ่งหมายถึงการดึงเอาเฉพาะ name ของผู้ใช้ออกมาแล้วเปลี่ยนให้เป็นตัวพิมพ์ใหญ่ทั้งหมด
   * ดังนั้น activeUsers จะถูกแปลงเป็น ['ALICE', 'CHARLIE']

ผลลัพธ์สุดท้ายที่เก็บใน activeUserNames จึงเป็น ['ALICE', 'CHARLIE']