นายโยธิน โฆษิตภัณฑบดี 6421600247
นายธนภัทร เอี่ยมอาจ 6421600093

Midterm ข้อที่ 1

นำข้อมูลมาแสดงใน Table html ด้วย Javascript

1. สร้างไฟล์ index.html และไฟล์ script.js

ไฟล์ index.html

```
<table id="leagueTable">
    <thead>
        <tr>
            <th>ทีม</th>
            <th>แข่ง</th>
            <th>ชนะ</th>
            <th>เสมอ</th>
            <th>แพ้</th>
            <th>ได้ประตู</th>
            <th>เสียประตู</th>
            <th>ส่วนต่าง</th>
            <th>คะแนน</th>
        </tr>
    </thead>
    <tbody></tbody>
</table>
```

2. สร้างไฟล์ script.js และอ้างอิงไฟล์ script.js ในไฟล์ index.html

ไฟล์ index.html

```
...
<script src="script.js"></script>
```

ไฟล์ script.js

```
var table = document.getElementById("leagueTable");
```

3. สร้างตัวแปรข้อมูลของตาราง

ตัวอย่างข้อมูลในตัวแปล tableData

```
const tableData = [
    {
        team: "Liverpool",
        played: 20,
        win: 13,
        draw: 6,
        loss: 1,
        goals_for: 43,
        goals_against: 18,
        goal_difference: 25,
        points: 45,
    },
    {
        team: "Aston Villa",
        played: 20,
        win: 13,
        draw: 3,
        loss: 4,
        goals_for: 43,
        goals_against: 27,
        goal_difference: 16,
        points: 42,
    },
];
```

4. เพิ่มข้อมูลในตาราง

วิธีที่ 1: ใช้ innerHTML และ for loop

ในไฟล์ script.js:

```
document.addEventListener("DOMContentLoaded", () => {
    const table = document.querySelector("#leagueTable tbody");
    tableData.forEach((data) => {
        const row = document.createElement("tr");
        row.innerHTML = `<td>${data.team}</td><td>${data.played}</td>
<td>${data.win}</td><td>${data.draw}</td><td>${data.loss}</td>
<td>${data.goals_for}</td><td>${data.goals_against}</td>
<td>${data.goal_difference}</td><td>${data.points}</td>`;
        table.appendChild(row);
    });
});
```

วิธีที่ 2: ใช้ Object และ for loop

ในไฟล์ script.js:

```
document.addEventListener("DOMContentLoaded", () => {
    const table = document.querySelector("#leagueTable tbody");
    tableData.forEach((data) => {
        const row = document.createElement("tr");
        const cell = {
            team: document.createElement("td"),
            played: document.createElement("td"),
            win: document.createElement("td"),
            draw: document.createElement("td"),
            loss: document.createElement("td"),
            goals_for: document.createElement("td"),
            goals_against: document.createElement("td"),
            goal_difference: document.createElement("td"),
            points: document.createElement("td"),
        };

        // เพิ่มข้อมูลลงในแต่ละ cell
        cell.team.textContent = data.team;
        cell.played.textContent = data.played;
        cell.win.textContent = data.win;
        cell.draw.textContent = data.draw;
        cell.loss.textContent = data.loss;
        cell.goals_for.textContent = data.goals_for;
        cell.goals_against.textContent = data.goals_against;
        cell.goal_difference.textContent = data.goal_difference;
        cell.points.textContent = data.points;

        // เพิ่ม cell ลงใน row
        Object.values(cell).forEach((value) => {
            row.appendChild(value);
        });

        // เพิ่ม row ลงในตาราง
        table.appendChild(row);
    });
});
```

Midterm ข้อที่ 2

เพิ่มข้อมูลจากฟอร์ม และแสดงใน Table ด้วย Javascript

จากข้อที่ 1 ได้มีการดึงข้อมูลจาก Object JSON มาแสดงใน Table แล้ว ในข้อนี้จะเพิ่มข้อมูลจากฟอร์ม และแสดงใน Table

1. สร้างฟอร์ม และปุ่มสำหรับเพิ่มข้อมูล

สร้างฟอร์มสำหรับเพิ่มข้อมูล โดยมีปุ่มสำหรับเพิ่มข้อมูล โดยมี id ไว้สำหรับใช้เรียกใช้งาน id addResultForm

```
<h1>เพิ่มผลการแข่งขัน</h1>
<form id="addResultForm">
    <input
        type="text"
        id="date"
        placeholder="Date (e.g., Sunday 14 January 2024)"
        required
    /><br />
    <input type="text" id="homeTeam" placeholder="Home Team" required />
<br />
    <input type="text" id="awayTeam" placeholder="Away Team" required />
<br />
    <input
        type="text"
        id="score"
        placeholder="Score (e.g., 2-1)"
        required
    /><br />
    <input type="text" id="location" placeholder="Location" required />
<br />
    <button type="submit" class="btn">Add Result</button>
</form>
```

2. สร้าง Event สำหรับเพิ่มข้อมูล

2.1 สร้าง Event จาก Form โดยใช้ addEventListener และใช้ preventDefault() เพื่อทำให้แน่ใจว่าจะเรียกใช้งานฟังก์ชันที่เราต้องการเท่านั้น

```
const addResultForm = document.getElementById("addResultForm");
const resultsTable = document.getElementById("resultsTable");

addResultForm.addEventListener("submit", (event) => {
    event.preventDefault();
});
```

2.2 นำข้อมูลจากฟอร์มมาเก็บไว้ในตัวแปร โดยใช้ value ของ input และเก็บไว้ในตัวแปรที่มีชื่อเดียวกับ id ของ input date, homeTeam, awayTeam, score, location ตามลำดับ

```
addResultForm.addEventListener("submit", (event) => {
    event.preventDefault();
    ...

    const date = document.getElementById("date");
    const homeTeam = document.getElementById("homeTeam");
    const awayTeam = document.getElementById("awayTeam");
    const score = document.getElementById("score");
    const location = document.getElementById("location");
  ...
});
```

2.3 สร้าง Object ของข้อมูลที่จะเพิ่ม โดยใช้ชื่อ key ตามชื่อของ column ในตาราง และใช้ value จากตัวแปรที่เก็บข้อมูลจากฟอร์ม มาเพิ่มเข้าไปใน Element ของ Tag tr โดยใช้ innerHTML และเก็บไว้ในตัวแปร newRow ก่อนจะเคลียร์ค่าในฟอร์ม และเพิ่ม newRow เข้าไปใน tbody ของตาราง

```
addResultForm.addEventListener("submit", (event) => {
    event.preventDefault();

    const date = document.getElementById("date");
    const homeTeam = document.getElementById("homeTeam");
    const awayTeam = document.getElementById("awayTeam");
    const score = document.getElementById("score");
    const location = document.getElementById("location");

    const newRow = document.createElement("tr");
    newRow.innerHTML = `
      <td>${date.value}</td>
      <td>${homeTeam.value} vs ${awayTeam.value}</td>
      <td>${score.value}</td>
      <td>${location.value}</td>
    `;

    // clear input
    date.value = "";
    homeTeam.value = "";
    awayTeam.value = "";
    score.value = "";
    location.value = "";

    resultsTable.querySelector("tbody").appendChild(newRow);
});
```

Midterm ข้อที่ 3

เกมทายตัวอักษร

ระบบจะสุ่ม word มา 1 คำ และแสดงให้ผู้ใช้เห็น _ ตามจำนวนตัวอักษรของคำนั้น และให้ผู้ใช้ทายตัวอักษร 1 ตัว โดยระบบจะตรวจสอบว่าตัวอักษรที่ผู้ใช้ทาย มีอยู่ในคำนั้นหรือไม่ ถ้ามีจะแสดงตำแหน่งของตัวอักษรนั้น และให้ผู้ใช้ทายต่อไปจนกว่าจะทายถูก หรือทายผิดหากผิดเกิน คูณ 2 จำนวนคำ ที่สุ่มได้

จากข้อที่ 1

1. สร้างโครงสร้าง html

อ้างอิงข้อมูลด้วย id

- word ใช้สำหรับแสดงคำที่สุ่มมา
- point ใช้สำหรับแสดงจำนวนครั้งที่ทายถูก
- maxpoint ใช้สำหรับแสดงจำนวนครั้งที่ทายได้สูงสุด
- letters ใช้สำหรับแสดงตัวอักษรที่ผู้ใช้ทายไปแล้ว
- message ใช้สำหรับแสดงข้อความว่าผู้ใช้ทายถูกหรือผิด
- playAgain ใช้สำหรับแสดงปุ่มเพื่อเริ่มเกมใหม่

ไฟล์ index.html

```
<h1>เกมทายอะไรเอ่ยย ENG</h1>
<div id="word">_ _ _ _ _</div>
<div>เดาได้ <span id="point"></span>/<span id="maxpoint"></span></div>
<div id="letters"></div>
<div id="message"></div>
<button id="playAgain" style="display: none" onclick="resetGame()">
    Play Again
</button>
```

2. สร้าง Object สำหรับคำที่จะนำมาสุ่ม

ไฟล์ script.js

```
const words = [
    "apple",
    "banana",
    "orange",
    "mango",
    ...
];
```

3. สร้างฟังก์ชันสำหรับเกม

3.1 กำหนดค่าเริ่มต้น

กำหนดค่าเริ่มต้นให้ตัวแปรต่างๆ

```
// ดึงข้อมูลจาก id html มาเก็บไว้ในตัวแปร
const letters = document.getElementById("letters");
const word_show = document.getElementById("word");
const show_point = document.getElementById("point");
const max_point_show = document.getElementById("maxpoint");
```

3.2 config point

ทำการกำหนดค่าเริ่มต้นของ point และ max_point โดยให้ point = 0 และ max_point = ความยาวของคำที่สุ่มมา * 2 เพื่อให้ผู้เล่นทาย

```
let point = 0; // ค่าเริ่มต้นของ point
let max_point = 0; // ค่าเริ่มต้นของ ค่าสูงสุดของ
// random word
const word_rand = words[Math.floor(Math.random() * words.length)]; // สุ่ม
คำจาก words
answer = "_".repeat(word_rand.length); // แทนตัวอักษรที่ยังไม่ถูกทายด้วย _

max_point = answer.length * 2; // คำนวณค่าสูงสุดของ point
show_point.innerHTML = point; // แสดงค่า point ที่เป็น 0
max_point_show.innerHTML = max_point;
```

3.3 สร้างฟังก์ชันสำหรับการเพิ่ม ปุ่ม ตัวอักษร

สร้าง function สำหรับการสุ่มตัวอักษร โดยจะนำ A-Z มาสร้างเป็นปุ่ม และเมื่อกดปุ่มจะเรียกใช้ฟังก์ชัน ให้เพิ่ม point ไปเรื่อย ละในแต่ละรอบจะตรวจสอบว่าตัวอักษรที่ผู้ใช้กด มีอยู่ในคำที่สุ่มมาหรือไม่ ถ้ามีจะแสดงตำแหน่งใน element ของตัวอักษรนั้น และให้ผู้ใช้ทายต่อไปจนกว่าจะทายถูก หรือทายผิดหากผิดเกิน คูณ 2 จำนวนคำ ที่สุ่มได้

```
const alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
for (let i = 0; i < alphabet.length; i++) {
    const btn = document.createElement("button"); // สร้างปุ่ม
    btn.classList.add("btn"); // เพิ่ม class ให้ปุ่ม
    btn.innerHTML = alphabet[i]; // ใส่ตัวอักษรลงในปุ่ม
    // เพิ่ม event ให้กับปุ่ม
    btn.addEventListener("click", () => {
        point += 1;
        show_point.innerHTML = point;
        checkWord(alphabet[i]); // เรียกใช้ฟังก์ชัน checkWord
```

```
        if (point == max_point) {
            document.getElementById("message").innerHTML =
                "Game over! The word was " + word_rand; // แสดงข้อความว่า
เกมจบแล้ว
            document.getElementById("playAgain").style.display =
"block";
            disableAllButton(); // เรียกใช้ฟังก์ชัน disableAllButton
        }
    });
    letters.appendChild(btn); // เพิ่มปุ่มลงใน div ที่มี id = letters
}
```

3.4 สร้างฟังก์ชันสำหรับการเช็คคำ

เป็น function ที่หาตำแหน่งของตัวอักษรที่ผู้ใช้ทาย และแทนตัวอักษรที่ถูกทายแล้วลงในคำตอบ และเช็คว่าชนะหรือยัง

```
function checkWord(alpha) {
    const word_split_list = word_rand.split("");
    console.log(word_split_list);
    console.log(alpha.toLowerCase());

    if (word_split_list.includes(alpha.toLowerCase())) {
        // get index of alpha
        indexList = [];
        for (let i = 0; i < word_split_list.length; i++) {
            if (word_split_list[i] == alpha.toLowerCase()) {
                indexList.push(i);
            }
        }

        // replace answer
        for (let i = 0; i < indexList.length; i++) {
            answer = answer.split("");
            answer[indexList[i]] = alpha.toLowerCase();
            answer = answer.join("");
        }

        // check win
        checkWin();

        word_show.innerHTML = answer;
    } else {
        console.log("false");
    }
}
```

3.5 สร้างฟังก์ชันสำหรับการเช็คชนะ

เป็น function ที่เช็คว่าชนะหรือยัง ถ้าชนะจะแสดงข้อความว่าชนะ และปุ่มเพื่อเริ่มเกมใหม่ แต่ถ้าเกมจบแล้วจะแสดงข้อความว่าเกมจบแล้ว และปุ่มเพื่อเริ่มเกมใหม่

```
function checkWin() {
    if (point == max_point) {
        document.getElementById("message").innerHTML =
            "Game over! The word was " + word_rand;
        document.getElementById("playAgain").style.display = "block";
        disableAllButton();
    } else if (answer == word_rand) {
        document.getElementById("message").innerHTML = "Congratulations!
You Win!";
        document.getElementById("playAgain").style.display = "block";
    }
}
```

3.6 สร้างฟังก์ชันสำหรับปุ่มเริ่มเกมใหม่

เป็น function ที่ใช้สำหรับทำให้ปุ่มตัวอักษรทั้งหมดทำงานไม่ได้ เพื่อไม่ให้ผู้ใช้กดตัวอักษรได้หลังจากเกมจบแล้ว

```
function disableAllButton() {
    const btn = document.getElementsByClassName("btn");
    for (let i = 0; i < btn.length; i++) {
        btn[i].disabled = true;
    }
}
```

3.7 เพิ่ม event ให้กับปุ่มเพื่อเริ่มเกมใหม่

เมื่อกดปุ่มเพื่อเริ่มเกมใหม่ จะทำการเรียกใช้ฟังก์ชัน reload window เพื่อเริ่มเกมใหม่

```
document.getElementById("playAgain").addEventListener("click", () => {
    window.location.reload();
});
```

Midterm ข้อที่ 4

1. สร้างโครงสร้าง html

สร้างโครงสร้าง html ดังนี้
สร้าง tag ต่างๆ ดังนี้ โดยจะมี id ใช้สำหรับกาอ้างอิงค์เพื่อใช้ในการแสดงผล

```
<h1>ณ ร้านขายผัก - ผลไม้ร้านหนึ่ง</h1>
<div id="result">ใส่คำตอบตรงนี้</div>
<p>Name: <span id="names"></span></p>
<p>Fruits: <span id="fruits"></span></p>
<p>Totalcost: <span id="totalcost"></span></p>
<p>Sorted Price: <span id="sortedprice"></span></p>
<p>Cucumber Details: <span id="cucumberdetails"></span></p>
<hr />

<p>Sum Price Fruit: <span id="sumf"></span></p>
<p>Low Price Fruit: <span id="lowpricefruit"></span></p>
<p>Sort Fruit Name: <span id="list_sort_name"></span></p>
<p>Average Price Fruit: <span id="average"></span></p>
<p>price more than 1: <span id="list_price_1"></span></p>
```

2. List Item ใช้สำหรับการทดลอง

```
const items = [
    { name: "Apple", category: "Fruit", price: 1 },
    { name: "Carrot", category: "Vegetable", price: 0.5 },
    { name: "Banana", category: "Fruit", price: 0.8 },
    { name: "Cucumber", category: "Vegetable", price: 1.2 },
    { name: "Orange", category: "Fruit", price: 0.9 },
    { name: "Tomato", category: "Vegetable", price: 0.7 },
    { name: "Lettuce", category: "Vegetable", price: 0.3 },
    { name: "Grapes", category: "Fruit", price: 2 },
    { name: "Mushroom", category: "Vegetable", price: 1.5 },
    { name: "Strawberry", category: "Fruit", price: 1.8 },
    { name: "Blueberry", category: "Fruit", price: 2.5 },
    { name: "Potato", category: "Vegetable", price: 0.4 },
    { name: "Broccoli", category: "Vegetable", price: 1.1 },
    { name: "Mango", category: "Fruit", price: 1.7 },
    { name: "Spinach", category: "Vegetable", price: 0.6 },
    { name: "Cherry", category: "Fruit", price: 2.2 },
    { name: "Peas", category: "Vegetable", price: 0.9 },
    { name: "Peach", category: "Fruit", price: 1.3 },
    { name: "Pineapple", category: "Fruit", price: 1.5 },
    { name: "Celery", category: "Vegetable", price: 0.8 },
];
```

.

3. ส่วนที่ 1

3.1 อ้างอิงค่าจาก html

อ้างอิงค่าจาก html โดยใช้ id ที่กำหนดไว้ใน html ดังนี้

```
const names = document.querySelector("#names");
const fruits = document.querySelector("#fruits");
const totalcost = document.querySelector("#totalcost");
const sortedprice = document.querySelector("#sortedprice");
const cucumberdetails = document.querySelector("#cucumberdetails");
```

3.2 สร้างฟังก์ชันสำหรับการนำชื่อผลไม้มาแสดง

สร้างฟังก์ชันสำหรับการนำชื่อมาแสดงโดยใช้ฟังก์ชัน map และ join ดังนี้

map คือการนำค่าจาก array มาแปลงให้เป็นค่าใหม่ โดยจะทำการวนลูปแต่ละค่าใน array และนำค่า name ใหม่ที่ได้จากการวนลูปแต่ละค่ามาเก็บไว้ใน array ใหม่

```
const namesList = items.map((item) => item.name);
names.innerHTML = namesList.join(", ");
```

3.3 สร้างฟังก์ชันสำหรับการนำชื่อผลไม้มาแสดง

filter เป็นการ filter ค่าใน array โดยจะทำการวนลูปแต่ละค่าใน array และเช็คว่าค่านั้นๆ มี category เป็น Fruit หรือไม่ ถ้าใช่ก็จะเก็บค่า name ลงใน array ใหม่ และนำ array name ใหม่ที่ได้มา join ด้วยเครื่องหมาย , ดังนี้

```
const fruitsList = items.filter((item) => item.category === "Fruit");
fruits.innerHTML = fruitsList.map((item) => item.name).join(", ");
```

3.4 สร้างฟังก์ชันสำหรับการนำเฉพาะผลไม้มารวมราคา

reduce เป็นการรวมค่าใน array โดยจะทำการวนลูปแต่ละค่าใน array และนำค่า price มาบวกกัน โดยค่าเริ่มต้นจะเป็น 0 ดังนี้

```
const SumTotalCost = items.reduce((total, item) => total + item.price,
0);
totalcost.innerHTML = "$" + SumTotalCost.toFixed(2);
```

3.5 สร้างฟังก์ชันสำหรับการนำเฉพาะผลไม้มาเรียงลำดับตามราคา

sort เป็นการเรียงลำดับค่าใน array โดยจะทำการวนลูปแต่ละค่าใน array และนำค่า price มาเปรียบเทียบกัน โดยจะเรียงจากน้อยไปมาก ดังนี้

```
const SortByPrice = items.sort((a, b) => a.price - b.price);
sortedprice.innerHTML = SortByPrice.map((item) => item.name).join(", ");
```

3.6 สร้างฟังก์ชันสำหรับการแสดงรายละเอียดของ Cucumber

find เป็นการหาค่าใน array โดยจะทำการวนลูปแต่ละค่าใน array และเช็คว่าค่านั้นๆ มี name เป็น Cucumber หรือไม่ ถ้าใช่ก็จะเก็บค่าที่เจอไว้ในตัวแปร CucumberDetails และแสดงผลออกมา ก่อนจะแสดงผลออกมาจะทำการแปลงเป็น string ด้วยฟังก์ชัน JSON.stringify ดังนี้

```
const CucumberDetails = items.find((item) => item.name === "Cucumber");
cucumberdetails.innerHTML = JSON.stringify(CucumberDetails);
```

ส่วนที่ 2

3.7 อ้างอิงค่าจาก html

นำ Id ที่อยู่ในช่วงที่ 2 มาเชื่อมโยงกับตัวแปรเพื่อมาใช้งานในช่วงที่ 2

```
const sumf = document.querySelector("#sumf");
const lowpricefruit = document.querySelector("#lowpricefruit");
const list_sort_namefruit = document.querySelector("#list_sort_name");
const averagefruit = document.querySelector("#average");
const list_price_1 = document.querySelector("#list_price_1");
```

3.8 รวมราคาเฉพาะ Category Fruit

รวมราคาของ Category Fruit โดยใช้ Filter ในการเลือกข้อมูลที่ต้องการก่อนจะมา บวกกันด้วย Reduce โดยให้คาเริ่มต้นเป็น 0

```
const SumPriceFruit = items.filter((item) => item.category === "Fruit");
const sumP = SumPriceFruit.reduce((total, item) => total + item.price,
0);
sumf.innerHTML = sumP;
```

3.9 หาผลไม้ที่มีราคถูกที่สุด

โดยในการหาราคาที่ถูกที่สุด จะเรียงราคาจากน้อยไปมาก โดยการนำ a และ b ที่อ้างอิงถึง item มาเปรียบเทียบกับ จากนั้นจะนำข้อมูลชื่อจาก index ที่ 0 หลังเรียงข้อมูลมาแล้วมาใช้ในการแสดง

```
const fruitlowprice = items.sort((a, b) => a.price - b.price);
lowpricefruit.innerHTML = fruitlowprice[0].name;
```

3.10 เรียงผลไม้ตามชื่อ

โดยในการเรียงผลไม้ตามชื่อ จะเรียงจากน้อยไปมาก โดยการนำ a < b คือ น้อยกว่า แล้วจะคือค่า -1 (ต้องเรียง a ก่อน b)
a > b คือ มากกว่า แล้วจะคือค่า 1 (ต้องเรียง b ก่อน a)
a = b คือ เท่ากัน แล้วจะคือค่า 0 (ไม่ต้องเปลี่ยนตำแหน่ง)

จากนั้นนำข้อมูลที่ถูกเรียงแล้วมาใช้ในการแสดงผลโดยใช้คำสั่ง map และ join ในการแสดงข้อมูลทั้งหมด

```
const sort_name = items.sort(function (a, b) {
    if (a.name < b.name) {
        return -1;
    }
    if (a.name > b.name) {
        return 1;
    }
    return 0;
});
list_sort_namefruit.innerHTML = sort_name.map((item) =>
item.name).join(", ");
```

3.11 หาค่าเฉลี่ยของผลไม้

โดยในการหาค่าเฉลี่ยของผลไม้ จะนำผลรวมของผลไม้ทั้งหมดมาหารด้วยจำนวนของผลไม้ทั้งหมด

```
const average = sumP / sort_word.length;
averagefruit.innerHTML = average;
```

3.12 แสดงผลไม้ที่มีราคามากกว่า 1 และเรียงจากมากไปน้อย

โดยในการแสดงผลไม้ที่มีราคามากกว่า 1 และเรียงจากมากไปน้อย จะใช้ filter ในการเลือกข้อมูลที่ต้องการก่อน แล้วจะนำข้อมูลที่เลือกมาเรียงจากมากไปน้อย โดยใช้ sort ในการเรียงข้อมูล และใช้ map และ join ในการแสดงผล

```
list_price_1.innerHTML = fruitsList
    .filter((item) => item.price > 1) // แสดงผลไม้ที่มีราคามากกว่า 1
    .sort((a, b) => b.price - a.price) // เรียงจากมากไปน้อย
    .map((item) => {
        return `${item.name} - $${item.price}`;
    })
    .join(", ");
```