

ปฏิบัติการที่ 5: JavaScript - เกมทายตัวเลข (2 คน)

ปฏิบัติการนี้เป็นการศึกษาการเขียน JavaScript ร่วมกับ HTML ด้วย Git Workflow โดยทำงานแบบ 2 คน เริ่มจากโค้ดพื้นฐาน (v1) แล้วปรับปรุงเป็น v2 ด้วยการใช้ branch และ commit อย่างเป็นระบบ

ความต้องการเบื้องต้น

- สุ่มตัวเลข 1-100
- รับ input
- เปรียบเทียบ
- แสดงผล ถูกต้อง/สูงไป/ต่ำไป

เริ่มต้น v1

- ไม่มี CSS เลย - ใช้ HTML พื้นๆ ง่ายๆ
- ไม่มี validation - ไม่เช็คตัวเลขหรือเปล่า
- ไม่นับจำนวนครั้ง - ไม่สนใจ
- ไม่มีปุ่มเริ่มใหม่ - ต้อง refresh หน้าเอง
- ไม่ล้าง input - พิมพ์ทับไปเรื่อยๆ
- ชื่อตัวแปรสั้นๆ - ans, g, txt (ไม่สนว่าจะเข้าใจหรือเปล่า)
- ไม่มี comment - เขียนไปเรื่อยๆ
- โครงสร้างแบบง่ายสุด - ไม่มี div, ไม่จัด layout

ปรับเพิ่ม v2

- เพิ่ม CSS - ใช้ bootstrap 5 โดยไม่ต้องเขียน css เอง
- มี validation - เช็คตัวเลขหรือเปล่า
- นับจำนวนครั้ง - แสดงให้เห็นว่าทายมากี่ครั้งแล้ว
- มีปุ่มเริ่มใหม่ - ไม่ต้อง refresh หน้าเอง
- ไม่ล้าง input - เมื่อเอาเมาส์มาคลิกในช่องทายตัวเลข ให้ select text ไปได้เลยเพื่อให้สามารถพิมพ์ทับได้ทันที
- ตั้งชื่อตัวแปรที่มีความหมาย - เพื่อความเข้าใจ
- มี comment - เขียน comment ไปด้วย
- รองรับ Responsive

ตอนที่ 0: สร้าง GitHub Repository (Pre-work)

วัตถุประสงค์

- สร้าง repository บน GitHub
- ตั้งค่า SSH key (optional แต่แนะนำ)
- Clone repository ลงเครื่อง
- ให้ นิสิต#1 ทำเป็นคนหลัก

ขั้นตอน

1. นิสิต#1: ไปที่ **GitHub** และสร้าง **Repository**
 - ไปที่ <https://github.com> (สมัครสมาชิกถ้ายังไม่มี)
 - คลิก “+” แล้ว “New repository”
 - ตั้งชื่อ: number-guessing-game หรือชื่ออื่น
 - เลือก visibility: **Public** หรือ **Private**
 - ✓ Initialize this repository with:
 - Add a README file
 - Add .gitignore (เลือก Python หรือ Node.js)

- Choose a license (MIT แนะนำ)

- คลิก **Create repository**

2. นิสิต#1: Clone Repository ลงเครื่อง

```
# ไปยังตำแหน่งที่จะเก็บโปรเจกต์
```

```
cd lab05\
```

```
# Clone repository (ใช้ HTTPS หรือ SSH)
```

```
git clone https://github.com/yourusername/number-guessing-game.git
```

```
# เข้าโฟลเดอร์ที่ clone มา
```

```
cd number-guessing-game
```

3. นิสิต#2: ตรวจสอบ Remote

```
git remote -v
```

ผลลัพธ์ที่ต้องเห็น:

```
origin https://github.com/yourusername/number-guessing-game.git (fetch)
```

```
origin https://github.com/yourusername/number-guessing-game.git (push)
```

4. นิสิต#1: ตั้งค่า Git User (ถ้ายังไม่เคย)

```
git config --global user.name "Your Full Name"
```

```
git config --global user.email "your.email@example.com"
```

เสร็จสิ้น

- ✓ GitHub Repository สร้างเรียบร้อยแล้ว
- ✓ Clone ลงเครื่องเรียบร้อยแล้ว
- ✓ Git remote ตั้งค่าถูกต้อง

ตอนที่ 1: สร้าง Git Repository และ Commit แรก

วัตถุประสงค์

- ตรวจสอบไฟล์ที่สร้างจาก GitHub
- สร้าง README.md ที่ดีขึ้น และ commit เป็นครั้งแรก
- Push ขึ้น GitHub
- ให้ นิสิต#1 ทำเป็นคนหลัก

ขั้นตอน

1. นิสิต#1: เปิด VS Code Terminal (Ctrl + `)

```
# ย้ายไปยังโฟลเดอร์ lab05
```

```
cd lab05\number-guessing-game
```

2. นิสิต#1: ตรวจสอบไฟล์ที่มี

```
ls -la
```

ผลลัพธ์ที่ต้องเห็น:

```
.git/  
.gitignore  
README.md  
LICENSE
```

3. นิสิต#1: ปรับปรุง **README.md**

```
# เกมสํหายตัวเลข (Number Guessing Game)
```

```
## ความเป็นมา
```

โปรแกรมเกมสํหายตัวเลข 1-100 เขียนด้วย HTML, CSS และ Vanilla JavaScript

```
## เวอร์ชัน
```

- v1: เวอร์ชันพื้นฐาน
- v2: เพิ่ม Bootstrap, validation, counter

```
## วิธีการใช้
```

1. เปิด `index.html` ในเบราว์เซอร์
2. พิมพ์ตัวเลข แล้วคลิก "ทาย"
3. โปรแกรมจะบอกว่าสูงไป/ต่ำไป/ถูกต้อง

```
## ข้อกำหนด
```

- HTML5
- CSS3 (Bootstrap 5)
- JavaScript ES6+

4. นิสิต#2: สังเกตการแก้ไข

- ดูไฟล์ README.md ที่ถูกแก้ไข

5. นิสิต#1: Commit และ Push ขึ้น GitHub

```
git add README.md  
git commit -m "docs: update README with project description"  
git push origin main
```

6. นิสิต#2: ตรวจสอบบน GitHub

- ไปที่ <https://github.com/yourusername/number-guessing-game>
- ดูว่า README.md ถูก update แล้ว

เสร็จสิ้น

- ✓ Repository สำเร็จ
- ✓ Commit แรกอยู่บน GitHub แล้ว

ตอนที่ 2: สร้าง Branch Develop

วัตถุประสงค์

- สร้าง branch develop สำหรับ integration

- สังเกตการแยกตัว branch บน GitHub
- ให้ **นิสิต#2** ทำเป็นคนหลัก

ขั้นตอน

1. นิสิต#2: สร้าง **Branch Develop**

```
git branch develop
git switch develop
```

หรือสั้นลง:

```
git switch -c develop
```

2. นิสิต#1: ตรวจสอบ **Branch**

```
git branch -a
```

ผลลัพธ์ที่ต้องเห็น:

```
* develop
  main
```

(* บ่งชี้ branch ปัจจุบันที่เราอยู่นอยู่)

3. นิสิต#2: **Push Branch Develop** ขึ้น **GitHub**

```
git push -u origin develop
```

-u = set upstream (บันทึกว่า local develop เชื่อมกับ remote develop)

4. นิสิต#1: ตรวจสอบบน **GitHub**

- ไปที่ <https://github.com/yourusername/number-guessing-game>
- ดูส่วน “Branch” ควรเห็น main และ develop

5. นิสิต#2: สังเกต **Branch Graph**

```
git log --graph --oneline --all
```

สิ่งที่ต้องสังเกต

- Branch develop แยกออกจาก main บน GitHub
- ทั้ง local และ remote มี develop branch แล้ว
- สามารถเปลี่ยน branch ได้จาก GitHub UI

เสร็จสิ้น

- ✓ Branch develop สร้างเรียบร้อยแล้ว
- ✓ Push ขึ้น GitHub แล้ว

ตอนที่ 3: Feature/Minimal (v1 - 4 Commits)

วัตถุประสงค์

- สร้าง feature/minimal branch
- เขียนโค้ด v1 ตามความต้องการเบื้องต้น

- ทำ 4 commits
- Merge เข้า develop และสังเกตผลลัพธ์
- ให้ นิสิต#1 ทำโดยส่วนใหญ่

ขั้นตอน

Step 1: นิสิต#1 สร้าง Feature Branch

```
git switch develop
git switch -c feature/minimal
```

Step 2: Commit ที่ 1 - นิสิต#1 สร้างไฟล์ HTML พื้นฐาน

สร้างไฟล์ index.html

```
<!DOCTYPE html>
<html lang="th">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>เกมส่ายตัวเลข</title>
  </head>
  <body>
    <h1>เกมส่ายตัวเลข 1-100</h1>

    <p>ส่ายตัวเลขตั้งแต่ 1 ถึง 100</p>

    <input type="text" id="txt" placeholder="พิมพ์ตัวเลข" />
    <button onclick="checkGuess()">ส่าย</button>

    <p id="result"></p>

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

```
git add index.html
git commit -m "feat: add basic HTML structure for number guessing game"
git add index.html
git commit -m "feat: add basic HTML structure for number guessing game"
```

Step 3: Commit ที่ 2 - นิสิต#2 เขียน JavaScript พื้นฐาน

สร้างไฟล์ script.js:

```
// filepath: script.js
// ตัวแปรเก็บตัวเลขลับ
let ans = Math.floor(Math.random() * 100) + 1;

// ฟังก์ชันตรวจสอบการส่าย
function checkGuess() {
  let g = parseInt(document.getElementById("txt").value);
  let result = document.getElementById("result");

  if (g === ans) {
    result.textContent = "✓ ถูกต้อง!";
  } else if (g > ans) {
    result.textContent = "↓ ตัวเลขสูงไป";
  } else {
```

```
    result.textContent = "↑ ตัวเลขต่ำไป";
  }
}
```

```
git add script.js
```

```
git commit -m "feat: add basic JavaScript logic for game comparison"
```

Step 4: Commit ที่ 3 - นิสิต#1 เพิ่ม CSS พื้นฐาน

สร้างไฟล์ style.css:

```
/* filepath: style.css */
body {
  font-family: Arial, sans-serif;
  padding: 20px;
}
```

```
input {
  padding: 5px;
  font-size: 16px;
}
```

```
button {
  padding: 5px 15px;
  font-size: 16px;
  cursor: pointer;
}
```

```
#result {
  margin-top: 20px;
  font-size: 18px;
  font-weight: bold;
}
```

```
git add style.css
```

```
git commit -m "feat: add basic CSS styling"
```

Step 5: Commit ที่ 4 - นิสิต#2 อัปเดต HTML ให้เชื่อมต่อกับ CSS

อัปเดต index.html:

```
<!-- filepath: index.html -->
<!DOCTYPE html>
<html lang="th">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>เกมส่ายตัวเลข</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h1>เกมส่ายตัวเลข 1-100</h1>

    <p>ส่ายตัวเลขตั้งแต่ 1 ถึง 100</p>

    <input type="text" id="txt" placeholder="พิมพ์ตัวเลข" />
    <button onclick="checkGuess()">ส่าย</button>

    <p id="result"></p>
```

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

```
git add index.html
git commit -m "docs: link CSS file to HTML"
```

Step 6: นิสิต#1 ตรวจสอบ Commits

```
git log --oneline
```

ผลลัพธ์ที่ต้องเห็น:

```
def5678 docs: link CSS file to HTML
abc4567 feat: add basic CSS styling
abc3456 feat: add basic JavaScript logic for game comparison
abc2345 feat: add basic HTML structure for number guessing game
abc1234 docs: initial commit with project description
```

Step 7: นิสิต#1 Merge เข้า Develop และ Push ขึ้น GitHub

```
git switch develop
git merge feature/minimal --no-ff -m "merge: feature/minimal (v1) into develop"
git push origin develop
```

Step 8: นิสิต#2 ตรวจสอบบน GitHub

```
git log --graph --oneline --all
```

- ไปที่ GitHub ดูการเปลี่ยนแปลงบน develop branch
- ควรเห็น feature/minimal merge เข้า develop
- **Merge commit** แสดงในประวัติ

ผลลัพธ์ที่ต้องเห็น:

```
* abc7890 merge: feature/minimal (v1) into develop
|\
| * def5678 docs: link CSS file to HTML
| * abc4567 feat: add basic CSS styling
| * abc3456 feat: add basic JavaScript logic for game comparison
| * abc2345 feat: add basic HTML structure for number guessing game
|/
* abc1234 docs: initial commit with project description
```

สิ่งที่ต้องสังเกต

- Feature branch ถูก merge กลับเข้า develop
- Git history แสดงการแยกตัว branch อย่างชัดเจน
- ทั้ง 4 commits ของ feature/minimal อยู่บน GitHub
- **GitHub branch graph** แสดงการ merge อย่างชัดเจน

เสร็จสิ้น

- ✓ สร้าง v1 ครบถ้วนตามความต้องการเบื้องต้น
- ✓ 4 commits ในประวัติ
- ✓ Merge เข้า develop สำเร็จ
- ✓ Push ขึ้น GitHub เรียบร้อย

ตอนที่ 4: Feature/Full (v2 - 8 Commits)

วัตถุประสงค์

- สร้าง feature/full branch
- ปรับปรุงโค้ด v1 เป็น v2 ตามข้อกำหนด
- ทำ 8 commits (ครั้งละ 1 ข้อ)
- Merge เข้า develop และสังเกตผลลัพธ์
- ให้ นิสิต#2 ทำโดยส่วนใหญ่

ขั้นตอน

Step 1: นิสิต#2 สร้าง Feature Branch

```
git switch develop
git switch -c feature/full
```

Step 2: Commit ที่ 1 - นิสิต#2 เปลี่ยนเป็น Bootstrap 5

อัปเดต index.html:

```
<!-- filepath: index.html -->
<!DOCTYPE html>
<html lang="th">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>เกมส่ายตัวเลข</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
      rel="stylesheet"
    />
  </head>
  <body class="bg-light">
    <div class="container mt-5">
      <div class="row justify-content-center">
        <div class="col-md-6">
          <div class="card shadow">
            <div class="card-body">
              <h1 class="card-title text-center">เกมส่ายตัวเลข</h1>
              <p class="text-center text-muted">ทายตัวเลขตั้งแต่ 1 ถึง 100</p>

              <div class="mb-3">
                <label for="guessInput" class="form-label">พิมพ์ตัวเลข:</label>
                <input
                  type="text"
                  id="guessInput"
                  class="form-control"
                  placeholder="เช่น 50"
                />
              </div>
            </div>
          </div>

          <div class="d-grid gap-2">
            <button onclick="checkGuess()" class="btn btn-primary">
              ทาย
            </button>
            <button onclick="resetGame()" class="btn btn-secondary">
              เริ่มใหม่
            </button>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        </div>

        <div id="resultContainer" class="mt-4"></div>
        <div id="attemptsContainer" class="mt-3 text-center"></div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>

<script src="script.js"></script>
</body>
</html>

```

```

git add index.html
git commit -m "style: replace CSS with Bootstrap 5 framework"

```

Step 3: Commit ที่ 2 - นิสิต#1 ตั้งชื่อตัวแปรใหม่และเพิ่ม comments

อัปเดต script.js:

```

// filepath: script.js

// ตัวแปรเก็บตัวเลขลับ
let secretNumber = 0;

// ตัวแปรนับจำนวนครั้งที่ทาย
let attemptCount = 0;

// ฟังก์ชันเริ่มเกมใหม่
function initializeGame() {
    secretNumber = Math.floor(Math.random() * 100) + 1;
    attemptCount = 0;
    updateDisplay();
}

// ฟังก์ชันตรวจสอบการทาย
function checkGuess() {
    const guessInput = document.getElementById("guessInput");
    const guessValue = parseInt(guessInput.value);
    const resultContainer = document.getElementById("resultContainer");

    attemptCount++;

    if (guessValue === secretNumber) {
        resultContainer.innerHTML = `
            <div class="alert alert-success" role="alert">
                <h5>✓ ถูกต้อง!</h5>
                <p>คุณทายถูกในครั้งที่ ${attemptCount}</p>
            </div>
        `;
    } else if (guessValue > secretNumber) {
        resultContainer.innerHTML = `
            <div class="alert alert-warning" role="alert">
                ↓ ตัวเลขสูงไป
            </div>
        `;
    } else {

```

```

    resultContainer.innerHTML = `
        <div class="alert alert-info" role="alert">
            ↑ ตัวเลขต่ำไป
        </div>
    `;
}

updateDisplay();
guessInput.value = "";
guessInput.focus();
}

// ฟังก์ชันอัปเดตจำนวนครั้ง
function updateDisplay() {
    const attemptsContainer = document.getElementById("attemptsContainer");
    attemptsContainer.textContent = `ท้ายแล้ว: ${attemptCount} ครั้ง`;
}

// ฟังก์ชันเริ่มเกมใหม่
function resetGame() {
    initializeGame();
    document.getElementById("resultContainer").innerHTML = "";
    document.getElementById("guessInput").value = "";
    document.getElementById("guessInput").focus();
}

// เริ่มเกมเมื่อโหลดหน้า
window.addEventListener("load", initializeGame);

```

```
git add script.js
```

```
git commit -m "refactor: rename variables and add comprehensive comments"
```

Step 4: Commit ที่ 3 - นิสิต#2 เพิ่ม Validation

อัปเดต script.js (แก้ฟังก์ชัน checkGuess):

```

// filepath: script.js
// ...existing code...

// ฟังก์ชันตรวจสอบการทาย
function checkGuess() {
    const guessInput = document.getElementById("guessInput");
    const guessValue = parseInt(guessInput.value);
    const resultContainer = document.getElementById("resultContainer");

    // Validation: ตรวจสอบว่าใส่ตัวเลขหรือไม่
    if (isNaN(guessValue) || guessInput.value === "") {
        resultContainer.innerHTML = `
            <div class="alert alert-danger" role="alert">
                กรุณาใส่ตัวเลข!
            </div>
        `;
        return;
    }

    // Validation: ตรวจสอบว่าอยู่ในช่วง 1-100 หรือไม่
    if (guessValue < 1 || guessValue > 100) {
        resultContainer.innerHTML = `

```

```

        <div class="alert alert-danger" role="alert">
            กรุณาใส่ตัวเลขระหว่าง 1 ถึง 100!
        </div>
    `;
    return;
}

attemptCount++;

if (guessValue === secretNumber) {
    resultContainer.innerHTML = `
        <div class="alert alert-success" role="alert">
            <h5>✓ ถูกต้อง!</h5>
            <p>คุณทายถูกในครั้งที่ ${attemptCount}</p>
        </div>
    `;
} else if (guessValue > secretNumber) {
    resultContainer.innerHTML = `
        <div class="alert alert-warning" role="alert">
            ↓ ตัวเลขสูงไป
        </div>
    `;
} else {
    resultContainer.innerHTML = `
        <div class="alert alert-info" role="alert">
            ↑ ตัวเลขต่ำไป
        </div>
    `;
}

updateDisplay();
guessInput.value = "";
guessInput.focus();
}

// ...existing code...

git add script.js
git commit -m "feat: add input validation for number range"

```

Step 5: Commit ที่ 4 - นิสิต#1 เพิ่มฟังก์ชัน resetGame และปุ่มในไฟล์

อัปเดต script.js เพิ่มฟังก์ชัน:

```

// filepath: script.js
// ...existing code...

// ฟังก์ชันเริ่มเกมใหม่
function resetGame() {
    initializeGame();
    document.getElementById("resultContainer").innerHTML = "";
    document.getElementById("guessInput").value = "";
    document.getElementById("guessInput").focus();
}

// ...existing code...

```

อัปเดต index.html เพิ่ม button:

```

<!-- filepath: index.html -->
<!-- ...existing code... -->
<div class="d-grid gap-2">
  <button onclick="checkGuess()" class="btn btn-primary">ทาย</button>
  <button onclick="resetGame()" class="btn btn-secondary">เริ่มใหม่</button>
</div>
<!-- ...existing code... -->

```

```

git add script.js index.html
git commit -m "feat: add reset game button functionality"

```

Step 6: Commit ที่ 5 - นิสิต#2 นับจำนวนครั้งที่ทาย

อัปเดต script.js โดยเพิ่มตัวแปร attemptCount และอัปเดต updateDisplay:

```

// filepath: script.js
// ตัวแปรนับจำนวนครั้งที่ทาย
let attemptCount = 0;

// ฟังก์ชันอัปเดตจำนวนครั้ง
function updateDisplay() {
  const attemptsContainer = document.getElementById("attemptsContainer");
  attemptsContainer.textContent = `ทายแล้ว: ${attemptCount} ครั้ง`;
}

```

และแก้ไข checkGuess() ให้เพิ่ม attemptCount:

```

function checkGuess() {
  const guessInput = document.getElementById("guessInput");
  const guessValue = parseInt(guessInput.value);
  const resultContainer = document.getElementById("resultContainer");

  // ... validation code ...

  attemptCount++; // เพิ่มตรงนี้

  if (guessValue === secretNumber) {
    resultContainer.innerHTML = `
      <div class="alert alert-success" role="alert">
        <h5>✓ ถูกต้อง!</h5>
        <p>คุณทายถูกในครั้งที่ ${attemptCount}</p>
      </div>
    `;
  }
  // ... rest of code ...
}

```

```

git add script.js
git commit -m "feat: add attempt counter display"

```

Step 7: Commit ที่ 6 - นิสิต#1 เพิ่ม auto-select text เมื่อคลิก input

อัปเดต script.js เพิ่ม event listener ที่เมื่อคลิก input ให้ select text อัตโนมัติ:

```

// filepath: script.js
// ...existing code...

// เพิ่มการ select text เมื่อคลิก input
document.addEventListener("DOMContentLoaded", function () {
  const guessInput = document.getElementById("guessInput");

```

```
guessInput.addEventListener("focus", function () {
  this.select();
});
});

// ...existing code...

git add script.js
git commit -m "feat: auto-select input text for better UX"
```

Step 8: Commit ที่ 7 - นิลิต#2 เพิ่ม CSS Custom

สร้างไฟล์ custom.css:

```
/* filepath: custom.css */
/* Custom styling for the number guessing game */

body {
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  min-height: 100vh;
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
}

.card {
  border: none;
  border-radius: 10px;
}

.card-title {
  font-weight: bold;
  color: #667eea;
  font-size: 2rem;
}

.btn-primary {
  background-color: #667eea;
  border: none;
}

.btn-primary:hover {
  background-color: #764ba2;
}

.alert {
  border-radius: 8px;
  font-size: 1.1rem;
}

#attemptsContainer {
  font-size: 1.2rem;
  font-weight: bold;
  color: #667eea;
}

@media (max-width: 576px) {
  .card-title {
    font-size: 1.5rem;
  }
}
```

```

.container {
  margin-top: 2rem;
}
}

```

อัปเดต index.html:

```

<!-- filepath: index.html -->
<!DOCTYPE html>
<html lang="th">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>เกมส่ายตัวเลข</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
      rel="stylesheet"
    />
    <link rel="stylesheet" href="custom.css" />
  </head>
  <body class="bg-light">
    <div class="container mt-5">
      <div class="row justify-content-center">
        <div class="col-md-6">
          <div class="card shadow">
            <div class="card-body">
              <h1 class="card-title text-center">เกมส่ายตัวเลข</h1>
              <p class="text-center text-muted">ทายตัวเลขตั้งแต่ 1 ถึง 100</p>

              <div class="mb-3">
                <label for="guessInput" class="form-label">พิมพ์ตัวเลข:</label>
                <input
                  type="text"
                  id="guessInput"
                  class="form-control"
                  placeholder="เช่น 50"
                />
              </div>

              <div class="d-grid gap-2">
                <button onclick="checkGuess()" class="btn btn-primary">
                  ทาย
                </button>
                <button onclick="resetGame()" class="btn btn-secondary">
                  เริ่มใหม่
                </button>
              </div>

              <div id="resultContainer" class="mt-4"></div>
              <div id="attemptsContainer" class="mt-3 text-center"></div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

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

```

```
</body>
</html>
```

```
git add custom.css index.html
git commit -m "style: add custom CSS for better responsive design"
```

Step 9: Commit ที่ 8 - นิสิต#1 เพิ่ม Keyboard Support (Enter key)

อัปเดต script.js:

```
// filepath: script.js
// ...existing code...

// เพิ่มการรองรับ Enter key
document.addEventListener("DOMContentLoaded", function () {
  document
    .getElementById("guessInput")
    .addEventListener("keypress", function (event) {
      if (event.key === "Enter") {
        checkGuess();
      }
    });
});

// ...existing code...
```

```
git add script.js
git commit -m "feat: add keyboard support for Enter key"
```

Step 10: นิสิต#2 ตรวจสอบ Commits

```
git log --oneline
```

ผลลัพธ์ที่ต้องเห็น:

```
xyz1234 feat: add keyboard support for Enter key
xyz1233 style: add custom CSS for better responsive design
xyz1232 feat: auto-select input text for better UX
xyz1231 feat: add attempt counter display
xyz1230 feat: add reset game button functionality
xyz1229 feat: add input validation for number range
xyz1228 refactor: rename variables and add comprehensive comments
xyz1227 style: replace CSS with Bootstrap 5 framework
def5678 merge: feature/minimal (v1) into develop
...
```

Step 11: นิสิต#2 Merge เข้า Develop และ Push ขึ้น GitHub

```
git switch develop
git merge feature/full --no-ff -m "merge: feature/full (v2) into develop"
git push origin develop
```

Step 12: นิสิต#1 ตรวจสอบบน GitHub

- ไปที่ GitHub ดูการเปลี่ยนแปลงบน develop branch
- ควรเห็น feature/full merge เข้า develop
- **Branch graph** แสดง v1 และ v2 อย่างชัดเจน

```
git log --graph --oneline --all
```

สิ่งที่ต้องสังเกต

- Feature/full branch มี 8 commits

- แต่ละ commit เป็นการปรับปรุง 1 ข้อ
- Merge สำเร็จและ develop มี v2 แล้ว
- **GitHub** แสดง **branch graph** ที่ยุ่งกว่า v1 (มี feature/minimal และ feature/full)
- ทั้ง local และ remote develop sync กัน

เสร็จสิ้น

- ✓ สร้าง v2 ครบถ้วนตามข้อกำหนด
- ✓ 8 commits ในประวัติ (ครั้งละ 1 ข้อ)
- ✓ Merge เข้า develop สำเร็จ
- ✓ Push ขึ้น GitHub เรียบร้อย

ตอนที่ 5: Merge Branch Develop เข้า Main

วัตถุประสงค์

- Merge branch develop เข้า main
- สังเกตโค้ดใน main ที่ GitHub
- สร้าง Release tag
- ให้ นิสิต#1 และ นิสิต#2 ทำร่วมกัน

ขั้นตอน

Step 1: นิสิต#1 ย้ายไปยัง Main

```
git switch main
git pull origin main
```

Step 2: นิสิต#2 Merge Develop เข้า Main

```
git merge develop --no-ff -m "merge: develop into main (v2 release)"
```

Step 3: นิสิต#1 สร้าง Tag สำหรับ Release

```
git tag -a v2.0 -m "Version 2.0 - Number Guessing Game with Bootstrap and Validation"
```

Step 4: นิสิต#2 Push ขึ้น GitHub

```
git push origin main
git push --tags
```

Step 5: นิสิต#1 ตรวจสอบ Branch Graph

```
git log --graph --oneline --all
```

ผลลัพธ์ที่ต้องเห็น:

```
* xyz9999 merge: develop into main (v2 release)
| \
| * xyz1234 feat: add keyboard support for Enter key
| * xyz1233 style: add custom CSS for better responsive design
| * ...
| /
* def5678 merge: feature/minimal (v1) into develop
* ...
```

Step 6: นิสิต#2 ตรวจสอบบน GitHub

- ไปที่ GitHub repository
- ดูส่วน **Releases** ควรเห็น v2.0 tag
- ดู **main branch** ควรมี v2 code เสร็จสมบูรณ์
- ดู **branch graph** แสดงการ merge ทั้งหมด

สิ่งที่ต้องสังเกต

- Main branch มี v2 ที่สมบูรณ์
- Tag v2.0 ที่ไปยัง release ล่าสุด
- GitHub แสดง **release page** พร้อม download option
- History แสดงการพัฒนาจาก v1 → v2 อย่างชัดเจน
- ทั้ง main, develop, และ tags ถูก push ขึ้น GitHub

เสร็จสิ้น

- ✓ Merge develop เข้า main สำเร็จ
- ✓ Tag v2.0 สร้างเรียบร้อย
- ✓ Push ขึ้น GitHub สำเร็จ
- ✓ Release page บน GitHub พร้อมใช้งาน

ตอนที่ 6: Challenge เพิ่มเติม

ตัวเลือกการปรับปรุง

Option 1: เพิ่ม Difficulty Level

- สร้าง branch feature/difficulty
- เพิ่ม dropdown เลือกระดับความยาก (1-10, 1-50, 1-100, 1-1000)
- อัปเดต logic สำหรับแต่ละระดับ
- Commit และ merge กลับ develop

Option 2: เพิ่ม High Score

- สร้าง branch feature/highscore
- ใช้ localStorage เก็บ high score
- แสดง high score บนหน้าเว็บ
- Commit และ merge กลับ develop

Option 3: เพิ่ม Timer

- สร้าง branch feature/timer

- เพิ่มตัวจับเวลา (เช่น 60 วินาที)
- จบเกมเมื่อหมดเวลา
- Commit และ merge กลับ develop

ขั้นตอนสำหรับ Challenge

```
# นิสิต#1 หรือ นิสิต#2 สร้าง feature branch
git switch develop
git pull origin develop
git switch -c feature/[your-choice]

# ทำการปรับปรุง
# (แก้ไขไฟล์)

# Commit (อาจหลายครั้ง)
git add .
git commit -m "feat: add [feature description]"

# Push ขึ้น GitHub
git push -u origin feature/[your-choice]

# สร้าง Pull Request บน GitHub
# 1. ไปที่ GitHub repository
# 2. คลิก "Pull requests" tab
# 3. คลิก "New pull request"
# 4. ตั้งค่า: feature/[your-choice] → develop
# 5. เขียนอธิบาย แล้วคลิก "Create pull request"

# รอให้อีกคนตรวจสอบและ approve

# Merge ผ่าน GitHub UI (ดีกว่า merge จาก command line)
# เลือก "Squash and merge" หรือ "Create a merge commit"

# หลังจาก merge เสร็จ ตั้ง develop ล่าสุด
git switch develop
git pull origin develop
```

บน GitHub

- นิสิต#1 และ นิสิต#2 จะเห็น **Pull Request** ระหว่าง feature branch และ develop
 - สามารถ comment, review code, และ approve ได้
 - หลังจาก merge ใน GitHub สามารถลบ feature branch ได้
-

สรุปคำสั่ง Git ที่สำคัญ

คำสั่ง	ความหมาย
git init	สร้าง repository ใหม่
git clone URL	Clone repository จาก GitHub
git switch -c branch-name	สร้างและย้ายไป branch ใหม่
git switch branch-name	ย้ายไป branch อื่น
git branch	ดูรายการ branch ทั้งหมด
git add .	เพิ่มไฟล์ทั้งหมดไปยัง staging
git commit -m "message"	บันทึก commit พร้อมข้อความ
git push origin branch-name	Push branch ขึ้น GitHub
git pull origin branch-name	ดึง branch ล่าสุดจาก GitHub
git merge branch-name	รวม branch เข้าในปัจจุบัน
git log --graph --oneline --all	ดู history ทั้งหมดแบบ graph
git tag -a v1.0 -m "message"	สร้าง tag เพื่อทำเครื่องหมาย release

สรุปบทบาทของ นิสิต#1 และ นิสิต#2

นิสิต#1 ทำหน้าที่:

- ตอนที่ 0: Setup GitHub repository
- ตอนที่ 1: ปรับปรุง README.md และ push
- ตอนที่ 3: สร้าง HTML และ CSS ใน v1
- ตอนที่ 4: ตั้งชื่อตัวแปร, ปุ่มเริ่มใหม่, auto-select, Enter key
- ตอนที่ 5: ตรวจสอบ branch graph, tag release, push tags

นิสิต#2 ทำหน้าที่:

- ตอนที่ 0: Clone repository จาก GitHub
- ตอนที่ 2: สร้าง develop branch และ push ขึ้น GitHub
- ตอนที่ 3: เขียน JavaScript พื้นฐาน, merge เข้า develop, push
- ตอนที่ 4: Bootstrap, validation, นับครั้ง, CSS custom, merge
- ตอนที่ 5: Merge develop → main, push

เป้าหมายการเรียนรู้

หลังจากทำปฏิบัติการนี้เสร็จ ทั้ง นิสิต#1 และ นิสิต#2 จะสามารถ:

- ใช้ Git workflow แบบ feature branch อย่างเป็นระบบ
- ทำงานเป็นทีม โดยหลีกเลี่ยง conflict
- สร้าง commit ที่มีความหมายและอธิบายได้ชัดเจน
- Merge branch อย่างถูกต้อง
- ออกแบบและพัฒนา Web Application จากเบื้องต้นไปจนสมบูรณ์
- ใช้ Bootstrap และ Vanilla JavaScript อย่างมีประสิทธิภาพ
- เข้าใจ Git history และ branch graph