Exercise 1

- Complete the following grid and flex tutorials: [CSS Grid Garden](#) [Flex Froggy](#)
- Complete all the levels
- Write a comparative table (pros and cons) between flexbox and grid
- Research about the best scenarios to use grid or flex

Exercise 2: CSS Methodologies & SCSS

Learning Goals:

- Understand and apply **BEM**, **OOCSS**, **SMACSS**, and **SCSS**

- Refactor existing spaghetti CSS into modular, maintainable code

- Practice writing reusable and scalable styles

Scenario

Your team has inherited a small project with unstructured HTML/CSS. You've been tasked to refactor the styles using modern methodologies and tools.

Project Description

A **Team Members Section** with cards for each profile.

HTML:

```
<section class="team">
  <h2>Meet the Team</h2>
  <div class="card">
    <img src="https://www.w3schools.com/howto/img_avatar2.png" alt="Alice"
class="avatar">
    <h3 class="name">Alice</h3>
    <p class="role">Designer</p>
    <button class="btn">Contact</button>
  </div>
  <div class="card">
    <img src="https://www.w3schools.com/howto/img_avatar.png" alt="Bob"
class="avatar">
    <h3 class="name">Bob</h3>
    <p class="role">Developer</p>
    <button class="btn">Contact</button>
  </div>
</section>
```

CSS:

```css
.card {
  border: 1px solid #ccc;
  padding: 15px;
  margin: 10px;
  text-align: center;
  width: 200px;
  display: inline-block;
}

.avatar {
  width: 80px;
  border-radius: 50%;
}

.name {
  font-size: 1.2em;
  margin: 10px 0;
}

.role {
  color: #777;
  font-style: italic;
}

.btn {
  background-color: #333;
  color: white;
  padding: 8px 12px;
  border: none;
  cursor: pointer;
}
```

Assignment Steps

1. **Refactor to BEM**
   a. Rename classes using BEM convention.
   b. Example: `.card__avatar`, `.card__role`, `.card__btn`
   c. Consider modifiers: `.card--featured`, `.btn--primary`

2. **Apply OOCSS Principles**
   a. Split structure and skin.
   b. Create reusable object classes like `.media`, `.btn`, `.box`
   c. Use classes like `.btn` + `.btn--dark` instead of `btn` with hardcoded styles

3. **Organize with SMACSS**
   a. Create logical file or comment sections: Base, Layout, Module, State, Theme
   b. Example: treat `.card` as a Module, `.btn--dark` as a State or Theme

4. **Convert to SCSS**
   a. Nest related styles
   b. Use variables (`$primary-color`, `$spacing`)
   c. Use mixins (`@mixin button-style`)
   d. Use `@import` or `@use` with logical structure (optional, if using CodePen with SCSS enabled)