Exercice 1 : root() et var()

Créer une page avec un thème défini dans :root, et appliquer des variables CSS.

1. **Instructions** :
    - Dans votre fichier CSS, définissez les couleurs principales et secondaires dans :root.
    - Utilisez ces variables pour styliser un titre (<h1>), un bouton (<button>), et un paragraphe (<p>).
2. **Code de départ :**

Html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exercice :root et var()</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Bienvenue sur mon site</h1>
  <p>Ceci est un paragraphe stylisé avec des variables CSS.</p>
  <button>Cliquez-moi</button>
</body>
</html>
```

Css

```
:root {
  --primary-color: #3498db;
  --secondary-color: #2ecc71;
  --font-size: 18px;
}
body {
  font-family: Arial, sans-serif;
  background-color: var(--primary-color);
  color: #fff;
}
h1 {
  font-size: calc(var(--font-size) * 2);
}
p {
  font-size: var(--font-size);
}
button {
  background-color: var(--secondary-color);
  color: #fff;
  border: none;
  padding: 10px 20px;
  cursor: pointer;
}
```

Exercice 2 : Utilisation de calc() pour des mises en page flexibles

Créer une boîte centrée qui ajuste sa taille en fonction de la largeur de l'écran.

1. **Instructions** :
    - Créez un conteneur avec une largeur basée sur un calcul entre une taille fixe et un pourcentage.
    - Ajoutez un bouton qui ajuste sa taille en fonction du conteneur.

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exercice calc()</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <button>Cliquez-moi</button>
  </div>
</body>
</html>
```

CSS

```
body {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  background-color: #f3f3f3;
}
.container {
  width: calc(50% + 100px);
  height: 200px;
  background-color: #8e44ad;
  display: flex;
  justify-content: center;
  align-items: center;
}
button {
  padding: calc(10px + 2%);
  background-color: #fff;
  border: none;
  cursor: pointer;
}
```

Exercice 3 : Utilisation avancée de attr

Créer une carte de profil qui affiche des informations dynamiquement grâce à `attr()`.

- Ajoutez une carte contenant une image, un nom, et un rôle.
- Utilisez `attr()` pour afficher une infobulle (tooltip) sur le rôle.

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exercice attr()</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="profile-card">
    <img src="https://via.placeholder.com/150" alt="Profile Image">
    <h2>Jane Doe</h2>
    <p class="role" data-role="Développeuse Web">Développeuse</p>
  </div>
</body>
</html>
```

CSS

```
body {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  font-family: Arial, sans-serif;
  background-color: #ecf0f1;
}
.profile-card {
  text-align: center;
  padding: 20px;
  background-color: #fff;
  border: 1px solid #ddd;
  border-radius: 8px;
  width: 300px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}
.profile-card .role::after {
  content: attr(data-role);
  display: block;
  margin-top: 10px;
```

```
    font-size: 14px;
    color: #7f8c8d;
}
```