

## Note:

This is **one** method of completing the following exercises, not the only method. How you choose to answer will depend on how you interpret the question, but the logic should remain roughly the same.

---

## Exercise 1:

Create a simple profile card component and apply CSS to style it.

```
// ProfileCard.css

.card {
  max-width: 300px;
  margin: 40px auto;
  border: 1px solid #ddd;
  padding: 20px;
  text-align: center;
}

.profileImg {
  width: 100px;
  height: 100px;
  border-radius: 50%;
  margin: 0 auto;
}

.profileName {
  font-size: 1.5em;
  margin: 10px 0;
}

.profileBio {
  font-size: 1em;
```

```
color: #666;  
}
```

## Exercise 2:

Build and style a responsive navigation bar with CSS that adapts to different screen sizes.

```
// NavBar.css  
  
.navBar {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  padding: 10px 20px;  
}  
  
.logo {  
  font-size: 1.5em;  
  font-weight: bold;  
}  
  
.navLinks {  
  list-style: none;  
  display: flex;  
}  
  
.navLinks li {  
  margin-left: 20px;  
}  
  
.navLinks li a {  
  text-decoration: none;  
  color: #333;
```

```

}

/* Responsive design */
@media (max-width: 600px) {
  .navBar {
    flex-direction: column;
  }

  .navLinks {
    flex-direction: column;
    align-items: center;
    margin-top: 10px;
  }

  .navLinks li {
    margin: 5px 0;
  }
}

```

## Exercise 3:

Implement a responsive grid layout for a series of content cards.

```

// ContentCard.css

.content-card {
  background: #fff;
  border-radius: 8px;
  box-shadow: 0px 2px 4px rgba(0, 0, 0, 0.1);
  margin: 10px;
  padding: 20px;
  text-align: center;
  transition: transform 0.2s;
}

.content-card:hover {
  transform: scale(1.05);
}

```

```
.card-image {  
  max-width: 100%;  
  height: auto;  
  border-radius: 4px;  
}  
  
// Grid.css  
  
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  gap: 20px;  
  padding: 20px;  
}  
  
/* Responsive design */  
@media (max-width: 768px) {  
  .grid-container {  
    grid-template-columns: repeat(2, 1fr);  
  }  
}  
  
@media (max-width: 480px) {  
  .grid-container {  
    grid-template-columns: 1fr;  
  }  
}
```