

# Design Fundamentals (Implementation)

≡ Week 2

Adib Sakhawat  
IUT CSE '21

## Spacing

Proper spacing ensures that a website is easy to read and navigate. It involves the strategic use of margins, padding, and whitespace to create a harmonious layout.

- **Margins and Padding:** Use CSS `margin` to control the space outside elements and `padding` for space inside elements. Consistent use of these properties helps in aligning content neatly.

```
.container {  
  margin: 20px;  
  padding: 15px;  
}
```

- **Whitespace:** Embrace whitespace to prevent clutter. It guides users' eyes and emphasizes important content.

```
.text-block {  
  margin-bottom: 30px;  
}
```

## Line Height

Line height affects the readability of text by controlling the vertical space between lines.

- **CSS `line-height`:** Set an appropriate `line-height` to enhance legibility, especially for longer passages of text.

```
p {
  line-height: 1.6;
}
```

- **Relative Units:** Using relative units like unitless numbers or percentages allows the line height to scale with the font size.

```
p {
  font-size: 16px;
  line-height: 1.5; /* 24px line-height */
}
```

## Pagination

Pagination improves user experience by dividing content into digestible sections.

- **HTML Structure:** Utilize semantic HTML elements for pagination controls.

```
<nav aria-label="Page navigation">
  <ul class="pagination">
    <li><a href="#">&laquo; Previous</a></li>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li></li>
    <li><a href="#">Next &raquo;</a></li>
  </ul>
</nav>
```

- **Accessible Design:** Ensure pagination is navigable via keyboard and screen readers.

```
.pagination a:focus {
  outline: 2px solid blue;
}
```

## Roundness

Rounded corners can make interfaces feel more friendly and modern.

- **CSS `border-radius`**: Apply `border-radius` to elements to achieve roundness.

```
.button {  
  border-radius: 5px;  
}
```

- **Circular Shapes**: Create circles by setting equal width and height and a `border-radius` of 50%.

```
.avatar {  
  width: 100px;  
  height: 100px;  
  border-radius: 50%;  
}
```

## Tone

The tone of your website communicates your brand's personality.

- **Color Palette**: Choose colors that align with the desired emotional response.

```
body {  
  background-color: #f5f5f5;  
  color: #333;  
}
```

- **Typography**: Select fonts that reflect your brand's voice.

```
h1 {  
  font-family: 'Roboto', sans-serif;  
}
```

## Content Density

Balancing content density ensures information is presented effectively without overwhelming users.

- **Hierarchical Layout:** Use headings, subheadings, and paragraphs to structure content.

```
<h1>Main Title</h1>
<h2>Subheading</h2>
<p>Paragraph text...</p>
```

- **Responsive Design:** Adjust content density based on screen size using media queries.

```
@media (max-width: 600px) {
  .sidebar {
    display: none;
  }
}
```

## Shadows

Shadows add depth, making flat elements appear layered and interactive.

- **CSS `box-shadow`:** Apply shadows to elements to enhance visual hierarchy.

```
.card {
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}
```

- **Shadow Parameters:** Customize shadows using offsets, blur radius, and color.

```
.button {
  box-shadow: inset 0 0 5px #000;
}
```

## Text and Component Size

Appropriate sizing ensures readability and usability across devices.

- **Relative Units for Text:** Use `em` or `rem` units for font sizes to accommodate user settings.

```
body {  
  font-size: 1rem; /* Equivalent to the root font size  
  */  
}
```

- **Scalable Components:** Design buttons and inputs that scale with text size.

```
.button {  
  padding: 0.5em 1em;  
}
```

## Buttons and Links

Buttons and links are primary interaction points and should be designed for clarity and responsiveness.

- **Visual Distinction:** Differentiate buttons from links through styling.

```
a {  
  color: blue;  
  text-decoration: underline;  
}  
  
.button {  
  background-color: blue;  
  color: white;  
  text-decoration: none;  
}
```

- **Interactive States:** Implement hover, focus, and active states for feedback.

```
.button:hover {  
  background-color: darkblue;  
}  
  
.button:active {
```

```
background-color: navy;
}
```

## Positioning

Effective positioning arranges elements in a coherent and intuitive layout.

- **CSS Position Properties:** Use `static`, `relative`, `absolute`, `fixed`, and `sticky` to control element positioning.

```
.fixed-header {
  position: fixed;
  top: 0;
  width: 100%;
}
```

- **Flexbox and Grid Layouts:** Utilize modern layout modules for flexible and responsive designs.

```
.container {
  display: flex;
  flex-direction: column;
}
```

## Hero Section or Call-Out Sections

Hero sections capture attention and communicate key messages immediately.

- **Hero Section Design:** Incorporate compelling visuals and clear calls to action.

```
<section class="hero">
  <h1>Welcome to Our Website</h1>
  <p>Your success starts here.</p>
  <a href="#" class="button">Get Started</a>
</section>
```

```
.hero {  
  background-image: url('hero.jpg');  
  background-size: cover;  
  color: white;  
  text-align: center;  
  padding: 100px 20px;  
}
```

- **Call-Out Sections:** Highlight important information or promotions throughout the site.

```
<div class="call-out">  
  <p>Don't miss our special offer!</p>  
</div>
```

```
.call-out {  
  background-color: #ffffae6;  
  border-left: 5px solid #ffd700;  
  padding: 15px;  
  margin: 20px 0;  
}
```

## Conclusion

Mastering these design fundamentals enables you to create websites that are not only visually appealing but also user-friendly and effective in communication. By thoughtfully applying principles of spacing, typography, color, and layout, you can enhance user experience and achieve your design objectives using pure HTML and CSS.

## Further Exploration

- Experiment with different fonts and color schemes to find the best fit for your project.
- Practice building responsive layouts that adapt to various screen sizes.
- Stay updated with CSS advancements to continually improve your design capabilities.