

# HTML & CSS Revision Guide

## Chapter 1: HTML

### HTML Tables

**What it is:** Tables display data in rows and columns.

**When to use:** Use for structured data like marksheets, schedules, invoices.

**Quick tips & pitfalls:** Do not use for page layout; always include for headers.

```
<table border="1">
  <caption>Student Marks</caption>
  <tr><th>Name</th><th>Subject</th><th>Marks</th></tr>
  <tr><td>Amit</td><td>Math</td><td>90</td></tr>
</table>
```

### HTML Forms

**What it is:** Forms collect user input.

**When to use:** Use for login, registration, feedback, search boxes.

**Quick tips & pitfalls:** Use proper input types and labels; use POST for sensitive data.

```
<form action="submit.php" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" required><br>
  <label>Gender:</label>
  <input type="radio" name="gender" value="male"> Male
  <input type="radio" name="gender" value="female"> Female<br>
  <button type="submit">Submit</button>
</form>
```

## Chapter 2: CSS Fundamentals

### Background Properties

**What it is:** Controls backgrounds (color, image, gradient) of elements.  
**When to use:** Use to highlight sections, create hero banners, or decorative effects.  
**Quick tips & pitfalls:** Always provide fallback color; use cover/contain for images.

```
.hero {  
  background-image: url("banner.jpg");  
  background-size: cover;  
  background-position: center;  
  background-repeat: no-repeat;  
  background-color: lightgray;  
}
```

### Text & Font Properties

**What it is:** Styles text appearance (font, size, color, alignment).  
**When to use:** Use for headings, paragraphs, articles to improve readability.  
**Quick tips & pitfalls:** Use relative units (em/rem) for responsive design; line-height improves readability.

```
.article-title {  
  font-family: Georgia, serif;  
  font-size: 28px;  
  text-align: center;  
}  
.article-body {  
  font-size: 16px;  
  line-height: 1.6;  
  color: #333;  
}
```

## Chapter 3: CSS Effects

### Transform

**What it is:** Applies rotation, scaling, translation, or skewing to elements.

**When to use:** Use for hover effects, rotating logos, zooming images.

**Quick tips & pitfalls:** Use with transition for smooth effects; affects layout visually, not space.

```
.logo:hover {  
    transform: rotate(360deg) scale(1.2);  
}
```

### Transition

**What it is:** Smoothly animates property changes over time.

**When to use:** Use for hover states, dropdown menus, image zooms.

**Quick tips & pitfalls:** Specify property explicitly; transition does not work on 'display'.

```
.card {  
    transform: scale(1);  
    transition: transform 0.3s ease-in-out;  
}  
.card:hover {  
    transform: scale(1.1);  
}
```

### Animation

**What it is:** Creates continuous or timed animations using @keyframes.

**When to use:** Use for loaders, bouncing icons, text slides.

**Quick tips & pitfalls:** Do not overuse; can be distracting; combine with transform-opacity for performance.

```
@keyframes blink {  
    0% { opacity: 1; }  
    50% { opacity: 0; }  
    100% { opacity: 1; }  
}  
.dot {  
    animation: blink 1s infinite;  
}
```

# Chapter 4: CSS Layout

## Display

**What it is:** Controls how elements are rendered in layout (block, inline, flex, grid).

**When to use:** Use block for sections, inline for text, flex/grid for complex layouts.

**Quick tips & pitfalls:** Avoid using 'inline-block' for complex alignment; use flex/grid instead.

```
.navbar a {  
  display: inline-block;  
  padding: 10px;  
}  
.cards {  
  display: flex;  
  gap: 20px;  
}
```

## Position

**What it is:** Controls element placement relative to normal flow or parent.

**When to use:** Use relative for small shifts, absolute for floating elements, fixed for sticky widgets, sticky for headers.

**Quick tips & pitfalls:** Avoid absolute without parent; sticky needs top/bottom values.

```
.navbar {  
  position: sticky;  
  top: 0;  
  background: white;  
}
```

## Grid

**What it is:** 2D layout system for rows and columns.

**When to use:** Use for complex layouts, dashboards, galleries.

**Quick tips & pitfalls:** Combine with auto-fit/minmax for responsive design; always define gaps.

```
.layout {  
  display: grid;  
  grid-template-columns: 200px 1fr;  
  grid-template-rows: 60px 1fr 40px;  
  gap: 10px;  
}  
.header { grid-column: 1 / 3; }  
.footer { grid-column: 1 / 3; }
```

## Chapter 5: Responsive Design

### Media Queries

**What it is:** Apply different styles based on screen size or device.

**When to use:** Use for mobile, tablet, and desktop layout adjustments.

**Quick tips & pitfalls:** Test on multiple devices; combine with flexible units.

```
@media (max-width: 600px) {  
  body { background: lightblue; }  
  .box { font-size: 14px; }  
}
```

## Chapter 6: Quick Cheat Sheet

- **Tables:** , , ,
- **Forms:** , , ,
- **Background:** background-color, background-image, background-size, background-repeat
- **Text/Font:** font-family, font-size, color, text-align, line-height
- **Transform:** rotate(), scale(), translate(), skew()
- **Transition:** transition-property, transition-duration, transition-timing-function
- **Animation:** @keyframes, animation-name, animation-duration, animation-iteration-count
- **Display:** block, inline, inline-block, flex, grid
- **Position:** static, relative, absolute, fixed, sticky
- **Grid:** grid-template-columns, grid-template-rows, gap, grid-column, grid-row
- **Media Queries:** @media (max-width: ...), @media (min-width: ...)