

# HTML & CSS Quick Reference

Elements, selectors, flexbox, grid, responsive

## HTML Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, initial-scale=1.0">
  <title>Page Title</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Hello World</h1>
  <script src="app.js"></script>
</body>
</html>
```

## Common Elements

### Text & Headings

```
<h1>Heading 1</h1> ... <h6>Heading 6</h6>
<p>Paragraph text</p>
<strong>Bold</strong> <em>Italic</em>
<br> <!-- line break -->
<hr> <!-- horizontal rule -->
```

### Links & Images

```
<a href="https://example.com">Link text</a>
<a href="#section">Anchor link</a>

```

### Lists

```
<ul>
  <li>Item</li>
</ul>
<ol>
  <li>First</li>
</ol>
```

## Forms

### Input Types

```
<form action="/submit" method="POST">
  <input type="text" name="user"
    placeholder="Name" required>
  <input type="email" name="email">
  <input type="password" name="pw">
  <input type="number" min="0" max="100">
  <input type="checkbox" name="agree">
  <input type="radio" name="size" value="M">
  <button type="submit">Send</button>
</form>
```

### Select & Textarea

```
<select name="color">
  <option value="r">Red</option>
  <option value="g" selected>Green</option>
</select>
<textarea name="msg" rows="4"
  cols="30"></textarea>
```

## Semantic HTML

<b>&lt;header&gt;</b>	Intro content or nav container
<b>&lt;nav&gt;</b>	Navigation links
<b>&lt;main&gt;</b>	Primary page content (one per page)
<b>&lt;section&gt;</b>	Thematic grouping of content
<b>&lt;article&gt;</b>	Self-contained content
<b>&lt;aside&gt;</b>	Sidebar or tangential content
<b>&lt;footer&gt;</b>	Footer for section or page
<b>&lt;figure&gt;</b>	Image/diagram with caption
<b>&lt;figcaption&gt;</b>	Caption for <figure>

## CSS Selectors

### Basic Selectors

<b>element</b>	<b>p { }</b> — all <p> elements
<b>.class</b>	<b>.card { }</b> — class selector
<b>#id</b>	<b>#main { }</b> — id selector
<b>*</b>	Universal selector (all elements)

### Combinators

<b>A B</b>	Descendant (B inside A)
<b>A &gt; B</b>	Direct child only
<b>A + B</b>	Adjacent sibling (B right after A)
<b>A ~ B</b>	General sibling (B after A)

### Pseudo-classes

<b>:hover</b>	Mouse over element
<b>:focus</b>	Element has focus
<b>:first-child</b>	First child of parent
<b>:nth-child(n)</b>	nth child (2n = even, odd)

## Box Model

```
.box {
  margin: 10px;      /* outside */
  border: 1px solid #333; /* edge */
  padding: 15px;    /* inside */
  width: 200px;
  box-sizing: border-box; /* include pad+border */
}
```

## Shorthand

<b>margin: 10px</b>	All sides
<b>margin: 10px 20px</b>	Top/bottom   left/right
<b>margin: 10px 20px 15px</b>	Top   left/right   bottom
<b>margin: 10px 20px 15px 5px</b>	Top   right   bottom   left

## Flexbox

### Container

```
.flex-container {
  display: flex;
  flex-direction: row; /* row | column */
  justify-content: center; /* main axis */
  align-items: center; /* cross axis */
  gap: 10px;
  flex-wrap: wrap;
}
```

### Item Properties

<b>flex: 1</b>	Grow to fill available space
<b>flex-grow: 2</b>	Grow ratio relative to siblings
<b>flex-shrink: 0</b>	Don't shrink below basis
<b>flex-basis: 200px</b>	Initial size before grow/shrink
<b>align-self: flex-end</b>	Override container alignment
<b>order: -1</b>	Visual order (default 0)

## Grid

### Container

```
.grid {
  display: grid;
  grid-template-columns: 1fr 2fr 1fr;
  grid-template-rows: auto;
  gap: 10px;
}
```

### Item Placement

```
.item {
  grid-column: 1 / 3; /* span col 1-2 */
  grid-row: 1 / 2;
}
/* shorthand: repeat, minmax */
grid-template-columns: repeat(3, 1fr);
grid-template-rows:
  repeat(auto-fit, minmax(200px, 1fr));
```

## Positioning

<b>static</b>	Default flow (no offset properties)
<b>relative</b>	Offset from normal position
<b>absolute</b>	Offset from nearest positioned ancestor
<b>fixed</b>	Offset from viewport (stays on scroll)
<b>sticky</b>	Relative until scroll threshold, then fixed

### Example

```
.parent { position: relative; }
.child {
  position: absolute;
  top: 0; right: 0; /* top-right corner */
}
```

## Media Queries

```
/* Mobile-first: base styles for small screens */
.container { padding: 10px; }

@media (min-width: 768px) {
  .container { max-width: 720px; }
}

@media (min-width: 1024px) {
  .container { max-width: 960px; }
}
```

## Common Breakpoints

<b>480px</b>	Small phones
<b>768px</b>	Tablets
<b>1024px</b>	Small desktops / landscape tablets
<b>1280px</b>	Desktops