

HTML & CSS Cheat Sheet

General HTML5 & CSS concepts · AI-generated summary — verify against MDN & caniuse.com before production use · see the full Web reference for details

HTML5 STRUCTURE

Semantic elements

header top area · nav links block · main unique content (one per page) · article self-contained unit · section themed group · aside tangential content · footer bottom area · div generic fallback

Skeleton

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport"
      content="width=device-width,
      initial-scale=1.0">
    <title>Page</title>
  </head>
  <body>
    <header><nav>...</nav></header>
    <main>
      <article><section>...</section></article>
      <aside>...</aside>
    </main>
    <footer>...</footer>
  </body>
</html>
```

Content types

metadata · flow · sectioning · phrasing · heading · embedded · interactive — describe *where* an element may appear

BOX MODEL

Layers (inside out)

content (sized by width/height) → padding → border → margin (transparent, pushes neighbors)

```
.box {
  width: 200px; height: 100px;
  padding: 10px 15px 20px 25px; /* T R B L */
  border: 5px solid #333;
  margin: 20px 0;                /* V H */
}
```

box-sizing

```
html { box-sizing: border-box; }
*, *::before, *::after {
  box-sizing: inherit;
}
```

content-box (default): width/height = content only, padding/border added on top. border-box: width/height = final rendered size.

Adjacent block margins (top/bottom) can *collapse* to the larger value — never in flex/grid.

FLEXBOX & GRID

Flexbox

```
.row {
  display: flex;
  flex-direction: row;        /* or column */
  justify-content: space-between; /* main axis
*/
  align-items: center;         /* cross
axis */
  gap: 8px;
}
.item {
  flex: 1 1 200px; /* grow shrink basis */
  order: 2;
}
```

Grid

```
.grid {
  display: grid;
  grid-template-columns: 1fr 2fr 1fr;
  grid-template-rows: auto 1fr;
  grid-template-areas:
    "nav head head"
    "nav main side";
  gap: 12px;
}
.item { grid-area: main; }
```

Responsive

```
@media (min-width: 768px) { }
@container (min-width: 400px) { }
```

SELECTORS & SPECIFICITY

Selector types

```
div          /* element */
.card        /* class */
#id          /* id */
a:hover      /* pseudo-class */
p::first-line/* pseudo-element */
a[target]    /* attribute */
*           /* universal */
```

Combinators

A B descendant · A > B child · A + B adjacent sibling · A ~ B general sibling

Specificity (id, class, elem)

inline style > #id (1,0,0) > .class/[attr]:hover (0,1,0) > element/:before (0,0,1). !important overrides all.
Modern: :has(selector), native nesting with &.

TRANSITIONS & ANIMATION

transition shorthand

```
.btn {
  transition: background-color 0.3s
    ease-in-out 0s;
  /* property duration timing-fn delay */
}
.btn:hover { background-color: #333; }
```

@keyframes + animation

```
@keyframes spin {
  from { transform: rotate(0deg); }
  to   { transform: rotate(360deg); }
}
.spinner {
  animation: spin 1.2s linear infinite;
  /* name duration timing-fn delay
  iteration-count direction
  fill-mode play-state */
}
```

Timing functions: ease, linear, ease-in-out, cubic-bezier(). Animate transform/opacity for best performance.

FORMS & ACCESSIBILITY

Labels & grouping

```
<label for="email">Email</label>
<input id="email" type="email" required>

<fieldset>
  <legend>Contact</legend>
  ...
</fieldset>
```

Constraint validation

```
<input type="text" required
  pattern="[A-Za-z]{3,}"
  aria-describedby="hint">
```

```
input:invalid { border-color: red; }
input:valid   { border-color: green; }
```

JS API: el.checkValidity(), el.validity.valid, el.setCustomValidity("msg").