

Advanced CSS Documentation

A comprehensive guide to advanced CSS properties and layout techniques.

Display Property

The display property defines how an element is displayed in the document flow.

Display Values

CSS

display: block;

/ Takes full width, starts on new line */*

display: inline;

/ Takes only necessary width, no line break */*

display: inline-block;

/ Inline but accepts width/height */*

display: none;

/ Element not displayed (removed from flow) */*

display: flex;

/ Flexbox container */*

display: grid;

/ Grid container */*

display: inline-flex;

/ Inline flexbox container */*

display: inline-grid;

/ Inline grid container */*

Block vs Inline vs Inline-Block

CSS

```
/* Block elements: div, h1-h6, p, section */
.block {
  display: block;
  width: 100%;      /* Takes full width available */
  margin: 10px 0;   /* Vertical margins work */
}

/* Inline elements: span, a, strong, em */
.inline {
  display: inline;
  /* width and height are ignored */
  /* only horizontal margin/padding work */
}

/* Inline-block: best of both */
.inline-block {
  display: inline-block;
  width: 200px;      /* Width and height work */
  margin: 10px;      /* All margins work */
}
```

Height and Width

Control the size of elements.

Basic Sizing

```
CSS

width: 300px;      /* Fixed width */
height: 200px;     /* Fixed height */
width: 50%;        /* Percentage of parent */
height: 100vh;     /* Viewport height (100% of screen) */
width: 100vw;      /* Viewport width */
```

Min and Max

```
CSS
```

```
min-width: 200px;    /* Minimum width */
max-width: 1200px;   /* Maximum width */
min-height: 300px;   /* Minimum height */
max-height: 500px;   /* Maximum height */
```

Box Sizing

CSS

```
box-sizing: content-box; /* Default: padding/border added to width */
box-sizing: border-box;  /* Recommended: padding/border included in width */

/* Best practice - apply to all elements */
* {
  box-sizing: border-box;
}
```

Auto and Fit Content

CSS

```
width: auto;          /* Default: fits content */
height: auto;          /* Default: fits content */
width: fit-content;    /* Shrinks to fit content */
```

Positions

Position property controls how elements are positioned in the document.

Position Values

CSS

```
position: static;    /* Default: normal flow */
position: relative;   /* Relative to its normal position */
position: absolute;   /* Relative to nearest positioned ancestor */
position: fixed;      /* Relative to viewport */
position: sticky;     /* Hybrid of relative and fixed */
```

Positioning with Offsets

CSS

/ Relative: moves from its original position */*

```
.relative {  
  position: relative;  
  top: 20px;      /* Moves 20px down */  
  left: 30px;     /* Moves 30px right */  
}
```

/ Absolute: positioned relative to parent */*

```
.absolute {  
  position: absolute;  
  top: 0;         /* Distance from top of parent */  
  right: 0;       /* Distance from right of parent */  
  bottom: 0;      /* Distance from bottom of parent */  
  left: 0;        /* Distance from left of parent */  
}
```

/ Fixed: stays in place when scrolling */*

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

/ Sticky: sticks when scrolling past threshold */*

```
.sticky {  
  position: sticky;  
  top: 0;        /* Sticks when reaching top */  
}
```

Z-Index

CSS

z-index: 10; */* Stacking order (higher = on top) */*

z-index: -1; */* Behind other elements */*

/ Only works with positioned elements (not static) */*

Background Images

Apply and control background images.

Basic Background Image

CSS

```
background-image: url('image.jpg');  
background-image: url('https://example.com/image.png');  
background-image: linear-gradient(to right, red, blue);
```

Background Properties

CSS

```
background-color: #f0f0f0;  
background-image: url('image.jpg');  
background-repeat: no-repeat; /* repeat, repeat-x, repeat-y, no-repeat */  
background-position: center; /* top, bottom, left, right, center */  
background-position: 50% 50%; /* x% y% */  
background-position: 10px 20px; /* x y */  
background-size: cover; /* cover, contain, auto */  
background-size: 100% 100%; /* width height */  
background-attachment: fixed; /* scroll, fixed, local */
```

Background Shorthand

CSS

```
background: #fff url('image.jpg') no-repeat center/cover fixed;  
/* color image repeat position/size attachment */
```

Multiple Backgrounds

CSS

```
background-image: url('overlay.png'), url('background.jpg');  
background-position: center, top left;  
background-repeat: no-repeat, repeat;
```

Combinators

Combinators define relationships between selectors.

Descendant Combinator (space)

```
CSS

/* Selects all <p> inside <div> (any level) */
div p {
  color: blue;
}
```

Child Combinator (>)

```
CSS

/* Selects <p> that are direct children of <div> */
div > p {
  color: red;
}
```

Adjacent Sibling (+)

```
CSS

/* Selects <p> immediately after <h1> */
h1 + p {
  font-weight: bold;
}
```

General Sibling (~)

```
CSS

/* Selects all <p> that are siblings after <h1> */
h1 ~ p {
  color: gray;
}
```

Pseudo-Classes

Select elements based on their state or position.

Common Pseudo-Classes

CSS

```
/* Link states */
a:link { color: blue; }      /* Unvisited link */
a:visited { color: purple; } /* Visited link */
a:hover { color: red; }     /* Mouse over */
a:active { color: orange; }  /* Being clicked */
a:focus { outline: 2px solid; } /* Keyboard focus */

/* Input states */
input:focus { border-color: blue; }
input:disabled { opacity: 0.5; }
input:checked { background: green; }

/* Structural pseudo-classes */
li:first-child { font-weight: bold; }
li:last-child { border: none; }
li:nth-child(odd) { background: #f0f0f0; }
li:nth-child(even) { background: white; }
li:nth-child(3) { color: red; }
li:nth-child(3n) { color: blue; } /* Every 3rd */

/* Other useful pseudo-classes */
p:first-of-type { margin-top: 0; }
p:last-of-type { margin-bottom: 0; }
div:not(.active) { display: none; }
input:valid { border-color: green; }
input:invalid { border-color: red; }
```

Pseudo-Elements

Style specific parts of an element.

Common Pseudo-Elements

CSS

```
/* Before and After */
.element::before {
  content: "→"; /* Required property */
  color: blue;
}

.element::after {
  content: " ✓";
  color: green;
}

/* First letter and line */
p::first-letter {
  font-size: 2em;
  font-weight: bold;
  float: left;
}

p::first-line {
  font-weight: bold;
  color: blue;
}

/* Selection */
::selection {
  background: yellow;
  color: black;
}

/* Placeholder text */
input::placeholder {
  color: #999;
  font-style: italic;
}
```

Practical Examples

CSS


```
/* Decorative elements */
```

```
.quote::before {  
  content: "";  
  font-size: 3em;  
  color: #ccc;  
}
```

```
/* Clearfix */
```

```
.clearfix::after {  
  content: "";  
  display: table;  
  clear: both;  
}
```

```
/* Custom bullets */
```

```
li::before {  
  content: " ► ";  
  color: blue;  
  font-weight: bold;  
}
```

Pagination

Create page navigation controls.

Basic Pagination

CSS

```
.pagination {  
  display: flex;  
  list-style: none;  
  gap: 5px;  
  padding: 0;  
}  
  
.pagination li {  
  display: inline-block;  
}  
  
.pagination a {  
  display: block;  
  padding: 8px 12px;  
  text-decoration: none;  
  border: 1px solid #ddd;  
  color: #333;  
  border-radius: 4px;  
  transition: all 0.3s;  
}  
  
.pagination a:hover {  
  background-color: #007bff;  
  color: white;  
  border-color: #007bff;  
}  
  
.pagination .active a {  
  background-color: #007bff;  
  color: white;  
  border-color: #007bff;  
}  
  
.pagination .disabled a {  
  color: #ccc;  
  pointer-events: none;  
}
```

Dropdown Menus

Create interactive dropdown navigation.

Dropdown CSS

CSS

```
.dropdown {  
  position: relative;  
  display: inline-block;  
}  
  
.dropdown-toggle {  
  padding: 10px 20px;  
  background-color: #333;  
  color: white;  
  border: none;  
  cursor: pointer;  
}  
  
.dropdown-menu {  
  position: absolute;  
  top: 100%;  
  left: 0;  
  background-color: white;  
  min-width: 200px;  
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
  display: none;  
  z-index: 1000;  
}  
  
.dropdown:hover .dropdown-menu {  
  display: block;  
}  
  
.dropdown-item {  
  display: block;  
  padding: 10px 20px;  
  color: #333;  
  text-decoration: none;  
  transition: background-color 0.3s;  
}  
  
.dropdown-item:hover {  
  background-color: #f0f0f0;  
}
```

Navigation Bar

Create horizontal and vertical navigation bars.

Horizontal Navigation

css

```
.navbar {  
  background-color: #333;  
  overflow: hidden;  
}  
  
.navbar ul {  
  list-style: none;  
  margin: 0;  
  padding: 0;  
  display: flex;  
}  
  
.navbar li {  
  flex: 0 0 auto;  
}  
  
.navbar a {  
  display: block;  
  color: white;  
  text-align: center;  
  padding: 14px 20px;  
  text-decoration: none;  
  transition: background-color 0.3s;  
}  
  
.navbar a:hover {  
  background-color: #555;  
}  
  
.navbar .active {  
  background-color: #04AA6D;  
}
```

Vertical Navigation

CSS

```
.sidebar {  
  width: 200px;  
  background-color: #f1f1f1;  
  height: 100vh;  
  position: fixed;  
}  
  
.sidebar ul {  
  list-style: none;  
  padding: 0;  
  margin: 0;  
}  
  
.sidebar a {  
  display: block;  
  padding: 16px;  
  color: #333;  
  text-decoration: none;  
  border-bottom: 1px solid #ddd;  
}  
  
.sidebar a:hover {  
  background-color: #555;  
  color: white;  
}
```

Website Layout

Common website layout patterns.

Basic Layout Structure

CSS

```
/* Full page layout */
body {
  margin: 0;
  font-family: Arial, sans-serif;
}

/* Header */
.header {
  background-color: #333;
  color: white;
  padding: 20px;
  text-align: center;
}

/* Navigation */
.navbar {
  background-color: #444;
  overflow: hidden;
}

/* Container */
.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}

/* Two column layout */
.row {
  display: flex;
  gap: 20px;
}

.sidebar {
  flex: 0 0 250px;
  background-color: #f1f1f1;
  padding: 20px;
}

.main-content {
  flex: 1;
  padding: 20px;
}
```

```
/* Footer */  
.footer {  
  background-color: #333;  
  color: white;  
  text-align: center;  
  padding: 20px;  
  margin-top: 40px;  
}
```

Responsive Layout

```
CSS  
  
/* Mobile first approach */  
.column {  
  width: 100%;  
  padding: 15px;  
}  
  
/* Tablet and up */  
@media (min-width: 768px) {  
  .row {  
    display: flex;  
    gap: 20px;  
  }  
  
  .column {  
    flex: 1;  
  }  
}  
  
/* Desktop */  
@media (min-width: 1024px) {  
  .container {  
    max-width: 1200px;  
  }  
}
```

Image Gallery

Create responsive image galleries.

Grid Gallery

CSS


```
.gallery {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));  
  gap: 20px;  
  padding: 20px;  
}
```

```
.gallery-item {  
  position: relative;  
  overflow: hidden;  
  border-radius: 8px;  
  box-shadow: 0 2px 8px rgba(0,0,0,0.1);  
}
```

```
.gallery-item img {  
  width: 100%;  
  height: 250px;  
  object-fit: cover;  
  display: block;  
  transition: transform 0.3s;  
}
```

```
.gallery-item:hover img {  
  transform: scale(1.1);  
}
```

```
.gallery-caption {  
  position: absolute;  
  bottom: 0;  
  left: 0;  
  right: 0;  
  background: rgba(0,0,0,0.7);  
  color: white;  
  padding: 10px;  
  transform: translateY(100%);  
  transition: transform 0.3s;  
}
```

```
.gallery-item:hover .gallery-caption {  
  transform: translateY(0);  
}
```

Masonry Gallery

CSS

```
.masonry {  
  column-count: 3;  
  column-gap: 20px;  
}  
  
.masonry-item {  
  break-inside: avoid;  
  margin-bottom: 20px;  
}  
  
.masonry-item img {  
  width: 100%;  
  display: block;  
  border-radius: 8px;  
}  
  
@media (max-width: 768px) {  
  .masonry {  
    column-count: 2;  
  }  
}  
  
@media (max-width: 480px) {  
  .masonry {  
    column-count: 1;  
  }  
}
```

Icons

Styling and using icons effectively.

Icon Basics

CSS

```
/* Using icon fonts (Font Awesome, etc.) */
```

```
.icon {  
  display: inline-block;  
  width: 1em;  
  height: 1em;  
  font-size: 24px;  
}
```

```
/* Icon sizing */
```

```
.icon-small { font-size: 16px; }  
.icon-medium { font-size: 24px; }  
.icon-large { font-size: 32px; }
```

```
/* Icon with text */
```

```
.icon-text {  
  display: inline-flex;  
  align-items: center;  
  gap: 8px;  
}
```

```
/* Circular icons */
```

```
.icon-circle {  
  width: 40px;  
  height: 40px;  
  border-radius: 50%;  
  background-color: #007bff;  
  color: white;  
  display: inline-flex;  
  align-items: center;  
  justify-content: center;  
}
```

```
/* Icon buttons */
```

```
.icon-button {  
  background: none;  
  border: none;  
  cursor: pointer;  
  padding: 8px;  
  border-radius: 4px;  
  transition: background-color 0.3s;  
}
```

```
.icon-button:hover {
```

```
background-color: #f0f0f0;
}
```

SVG Icons

CSS

```
.svg-icon {
  width: 24px;
  height: 24px;
  fill: currentColor;
}

/* Inline SVG styling */
svg {
  vertical-align: middle;
}
```

Flexbox

Powerful layout system for arranging elements.

Flex Container

CSS

```
.flex-container {  
  display: flex;  
  
  /* Direction */  
  flex-direction: row;      /* row, column, row-reverse, column-reverse */  
  
  /* Wrapping */  
  flex-wrap: nowrap;        /* nowrap, wrap, wrap-reverse */  
  
  /* Main axis alignment */  
  justify-content: flex-start; /* flex-start, flex-end, center, space-between, space-around, space-evenly */  
  
  /* Cross axis alignment */  
  align-items: stretch;     /* stretch, flex-start, flex-end, center, baseline */  
  
  /* Multiple lines alignment */  
  align-content: flex-start; /* flex-start, flex-end, center, space-between, space-around, stretch */  
  
  /* Gap between items */  
  gap: 20px;  
  row-gap: 20px;  
  column-gap: 10px;  
}
```

Flex Items

CSS

```
.flex-item {  
  /* Growth */  
  flex-grow: 1;      /* How much item grows (default: 0) */  
  
  /* Shrinkage */  
  flex-shrink: 1;     /* How much item shrinks (default: 1) */  
  
  /* Base size */  
  flex-basis: 200px;  /* Initial size (default: auto) */  
  
  /* Shorthand */  
  flex: 1 1 200px;    /* grow shrink basis */  
  flex: 1;            /* Common: equal distribution */  
  
  /* Individual alignment */  
  align-self: center; /* auto, flex-start, flex-end, center, baseline, stretch */  
  
  /* Order */  
  order: 2;           /* Change visual order (default: 0) */  
}
```

Common Flexbox Patterns

CSS

```
/* Center everything */
```

```
.center-all {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
}
```

```
/* Equal columns */
```

```
.columns {  
  display: flex;  
  gap: 20px;  
}
```

```
.column {  
  flex: 1;  
}
```

```
/* Sidebar layout */
```

```
.layout {  
  display: flex;  
  gap: 20px;  
}
```

```
.sidebar {  
  flex: 0 0 250px;  
}
```

```
.main {  
  flex: 1;  
}
```

```
/* Card layout */
```

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

```
.card-header {  
  flex: 0 0 auto;  
}
```

```
.card-body {
```

```
    flex: 1;
  }

  .card-footer {
    flex: 0 0 auto;
  }

  /* Responsive navigation */
  .nav {
    display: flex;
    justify-content: space-between;
    align-items: center;
    flex-wrap: wrap;
  }
```

Flexbox vs Grid

- **Flexbox:** One-dimensional layout (row OR column), content-driven
 - **Grid:** Two-dimensional layout (rows AND columns), layout-driven
-

Complete Example

Combining multiple concepts:

CSS


```
/* Full page layout with flexbox */
```

```
body {  
  margin: 0;  
  font-family: Arial, sans-serif;  
  display: flex;  
  flex-direction: column;  
  min-height: 100vh;  
}  
  
.header {  
  background: linear-gradient(to right, #667eea, #764ba2);  
  color: white;  
  padding: 20px;  
  position: sticky;  
  top: 0;  
  z-index: 100;  
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
}  
  
.navbar {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  max-width: 1200px;  
  margin: 0 auto;  
}  
  
.nav-links {  
  display: flex;  
  gap: 20px;  
  list-style: none;  
}  
  
.nav-links a {  
  color: white;  
  text-decoration: none;  
  padding: 8px 16px;  
  border-radius: 4px;  
  transition: background-color 0.3s;  
}  
  
.nav-links a:hover {  
  background-color: rgba(255,255,255,0.2);
```

```
}
```

```
.main-content {  
  flex: 1;  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 40px 20px;  
  width: 100%;  
}
```

```
.gallery {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
  gap: 20px;  
}
```

```
.gallery-item {  
  position: relative;  
  overflow: hidden;  
  border-radius: 8px;  
  box-shadow: 0 4px 6px rgba(0,0,0,0.1);  
  transition: transform 0.3s;  
}
```

```
.gallery-item:hover {  
  transform: translateY(-5px);  
}
```

```
.footer {  
  background-color: #2d3748;  
  color: white;  
  padding: 20px;  
  text-align: center;  
}
```