

CSS クイックリファレンス

セレクター、レイアウト、Flexbox、Grid、アニメーション、レスポンシブデザイン

セレクター

基本セレクター

*	ユニバーサル - すべての要素
div	タイプ - すべての <code><div></code> 要素
.class	クラス - クラスを持つ要素
#id	ID - id を持つ要素
[attr]	属性 - 属性を持つ要素
[attr="val"]	属性が値と一致

コンビネーター

A B	子孫 (任意の深さ)
A > B	直接の子のみ
A + B	隣接する兄弟 (次の要素)
A ~ B	一般的な兄弟 (以降すべて)

疑似クラス

:hover	マウスオーバー中の要素
:focus	フォーカスがある要素
:first-child	親の最初の子
:nth-child(n)	N 番目の子 (1 始まり、 odd 、 even 、 2n+1)
:not(sel)	否定 - 一致を除外
:has(sel)	親セレクター (一致を含む)

疑似要素

::before	要素の前にコンテンツを挿入
::after	要素の後にコンテンツを挿入
::placeholder	input のプレースホルダーテキストをスタイル
::selection	ハイライトされたテキストをスタイル

ボックスモデル

ボックスサイジング

```
/* padding/border を幅に含める */
*, ::before, ::after {
  box-sizing: border-box;
}
```

プロパティ

margin	ボーダーの外側のスペース
border	padding の周囲の枠
padding	ボーダーの内側のスペース
width / height	コンテンツの寸法
outline	margin の外側のリング (スペースなし)

ショートハンド

```
margin: 10px; /* 全辺 */
margin: 10px 20px; /* 垂直 | 水平 */
margin: 10px 20px 30px; /* 上 | 水平 | 下 */
margin: 10px 20px 30px 40px; /* 上 右 下 左 */
```

Flexbox

コンテナ

```
.flex {
  display: flex;
  justify-content: center; /* 主軸 */
  align-items: center; /* 交差軸 */
  gap: 1rem;
}
```

コンテナプロパティ

flex-direction	row column row-reverse column-reverse
flex-wrap	nowrap wrap wrap-reverse
justify-content	flex-start center space-between space-around space-evenly
align-items	stretch center flex-start flex-end baseline
align-content	複数行の交差軸方向の揃え
gap	フレックスアイテム間のスペース

アイテムプロパティ

flex: 1	利用可能なスペースを埋めるように伸長
flex: 0 0 200px	固定幅、伸長/縮小なし
align-self	1つのアイテムの align-items を上書き
order	表示順を変更 (デフォルト 0)

Grid

コンテナ

```
.grid {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: auto 1fr auto;
  gap: 1rem;
}
```

コンテナプロパティ

grid-template-columns	列トラックを定義: 1fr 2fr, repeat(3, 1fr)
grid-template-rows	行トラックを定義
grid-template-areas	名前付き領域: "header header" "nav main"
gap	行と列のギャップ
justify-items	インライン方向 (水平) でアイテムを揃え
align-items	ブロック方向 (垂直) でアイテムを揃え

アイテムの配置

```
.item {
  grid-column: 1 / 3; /* 列 1-2 にまたがる */
  grid-row: 1 / -1; /* すべての行にまたがる */
  grid-area: header; /* 名前付き領域 */
}
```

タイポグラフィ

フォントプロパティ

font-family	フォントスタック: 'Inter', sans-serif
font-size	サイズ: 1rem, 16px, clamp(1rem, 2vw, 2rem)
font-weight	normal (400) bold (700) 100~900
font-style	normal italic oblique
line-height	行間: 1.5 (単位なし推奨)
letter-spacing	文字間隔: 0.05em

テキストプロパティ

text-align	left center right justify
text-decoration	none underline line-through
text-transform	uppercase lowercase capitalize
text-overflow	ellipsis (overflow: hidden と併用)
white-space	nowrap pre pre-wrap
word-break	break-all break-word

色と背景

カラーフォーマット

```
color: #ff6600; /* hex */
color: rgb(255, 102, 0); /* rgb */
color: hsl(24, 100%, 50%); /* hsl */
color: oklch(70% 0.15 50); /* oklch */
```

背景プロパティ

background-color	単色塗りつぶし: #f0f0f0
background-image	url(img.jpg) またはグラデーション
background-size	cover contain 100px 200px
background-position	center top right 50% 50%
background-repeat	no-repeat repeat-x repeat-y

グラデーション

```
background: linear-gradient(to right, #f00, #00f);
background: radial-gradient(circle, #fff, #000);
background: conic-gradient(red, yellow, green);
```

トランジションとアニメーション

トランジション

```
.btn {
  transition: background 0.3s ease, transform 0.2s;
}
.btn:hover {
  background: #0056b3;
  transform: scale(1.05);
}
```

トランジションプロパティ

transition-property	アニメーションするプロパティ
transition-duration	長さ: 0.3s, 300ms
transition-timing-function	ease linear ease-in-out cubic-bezier()
transition-delay	開始前の待機時間

キーフレームアニメーション

```
@keyframes spin {
  from { transform: rotate(0deg); }
  to { transform: rotate(360deg); }
}
.icon { animation: spin 1s linear infinite; }
```

アニメーションプロパティ

animation-name	@keyframes 名を参照
animation-duration	1 サイクルの長さ
animation-iteration-count	1 3 infinite
animation-direction	normal alternate reverse
animation-fill-mode	forwards backwards both

レスポンシブデザイン

メディアクエリ

```
@media (max-width: 768px) {
  .sidebar { display: none; }
}
@media (prefers-color-scheme: dark) {
  body { background: #1a1a1a; color: #eee; }
}
```

CSS クイックリファレンス

一般的なブレイクポイント

max-width: 480px	モバイル端末
max-width: 768px	タブレット
max-width: 1024px	小型ノート PC
max-width: 1280px	デスクトップ

ビューポート単位

vw / vh	ビューポート幅 / 高さの %
dvh	動的ビューポート高さ (モバイル対応)
svh / lvh	小 / 大ビューポート高さ
cqi	コンテナクエリインラインサイズ

コンテナクエリ

```
.card-wrapper { container-type: inline-size; }
@container (min-width: 400px) {
  .card { flex-direction: row; }
}
```

ポジショニング

position の値

static	デフォルト - 通常のドキュメントフロー
relative	通常位置からオフセット; スペースを保持
absolute	最も近い位置付き祖先に対して配置
fixed	ビューポートに対して配置; スクロールしても固定
sticky	スクロールに基づいて relative/fixed を切り替え

センタリングパターン

```
/* Flex でセンタリング */
display: flex;
justify-content: center;
align-items: center;

/* Grid でセンタリング */
display: grid;
place-items: center;
```

スタッキング

z-index	重なり順 (大きいほど上); position が必要
isolation: isolate	新しいスタッキングコンテキストを作成

カスタムプロパティ

定義と使用

```
:root {
  --color-primary: #3b82f6;
  --spacing-md: 1rem;
}
.btn {
  background: var(--color-primary);
  padding: var(--spacing-md);
}
```

フォールバック値

```
color: var(--accent, #ff6600);
/* --accent が未定義の場合は #ff6600 を使用 */
```

動的テーマ

```
[data-theme="dark"] {
  --bg: #1a1a2e;
  --text: #e0e0e0;
}
body { background: var(--bg); color: var(--text); }
```