

```
.nav { position: sticky; top: 0; z-index: 100; }.sidebar { position: sticky; top: 20px; }
```

```
position: sticky; top: 0;
```



```
overflow: hidden;
```

CSS position:sticky 完全ガイド

基本概念から実践的な使い方、トラブルシューティングまで、
Web制作者必見のスクロール連動要素固定テクニック

```
.parent { overflow: visible; height: 100px; }
```

```
.sticky-element {  
  position: sticky;  
  top: 0;  
  z-index: 100;  
}
```

```
position: sticky; bottom: 0;
```

```
position: sticky; top: 0;
```



1 position:stickyとは？

position:sticky; は「スクロールに追従し、特定の場所に到達すると固定される」CSSプロパティです。

relativeとfixedの特性を併せ持ったユニークな配置方法です。

特徴1: スティッキーアイテム

position: sticky;を指定された要素自身

特徴2: コンテナ（重要）

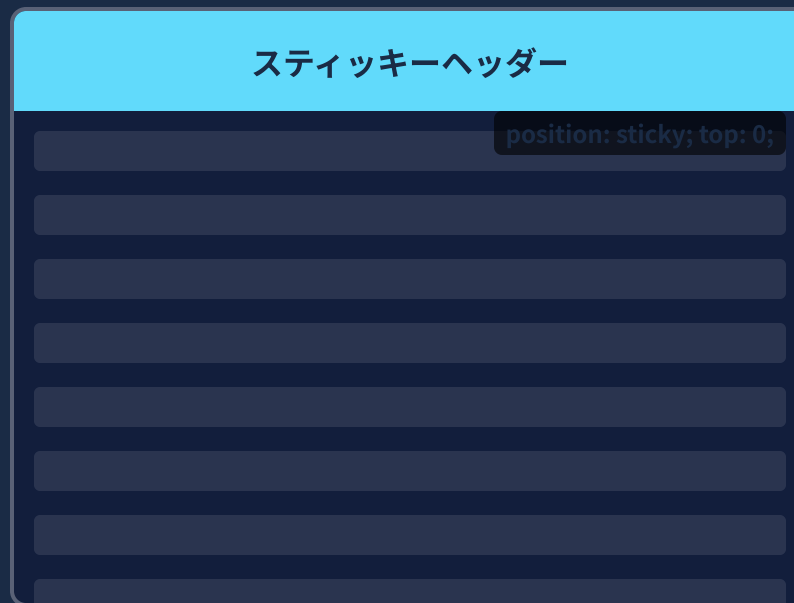
スティッキーアイテムを直接囲む親要素の範囲内でのみ固定される

特徴3: 固定位置の指定

top, bottom, left, rightのいずれかで固定位置を指定

```
position: sticky; top: 0;
```

スクロールに追従して固定されるイメージ



```
/* 基本的な書き方 */
.sticky-element {
  position: sticky;
  top: 0; /* 画面上端で固定 */

  /* Safari対応 */
  position: -webkit-sticky;
}

/* コンテナの範囲内でのみ固定される */
```

※ position: fixed との違い：fixed は常に画面の同じ位置に固定されるが、sticky はコンテナの範囲内でのみ固定される

2 position:sticky vs fixed

position:sticky と **position:fixed** はどちらも要素を固定できますが、その動作は大きく異なります。

適切な使い分けがユーザー体験の向上につながります。

固定範囲の違い

sticky：親要素の範囲内でのみ固定

fixed：ビューポート全体で常に固定

スクロール動作

sticky：特定位置まで通常スクロール、その後固定

fixed：スクロールと無関係に常に同じ位置

使い分けの判断基準

「常に同じ位置」→**fixed**（例：トップへ戻るボタン）

「特定セクション内だけ」→**sticky**（例：記事内サイドバー）

親要素の範囲

sticky

top: 50px

fixed

常に固定位置

特性	sticky	fixed
親要素	依存する	独立
スクロール追従	条件付き	常時
実装例	ヘッダー、サイドバー	クッキーバナー

```
/* sticky */
.sticky-header {
  position: sticky;
  top: 0;
}
```

```
/* fixed */
.fixed-button {
  position: fixed;
  bottom: 20px;
  right: 20px;
}
```

4 position:stickyの基本実装方法

position:sticky; の実装は非常にシンプルです。ただし**2つの必須プロパティ**が必要です。

これらがないと、ただの通常要素のように振る舞います。

必須プロパティ1: position: sticky

要素にstickyポジショニングを適用する基本プロパティ

必須プロパティ2: 位置指定

top, bottom, left, right のいずれか最低1つ

ブラウザ対応

Safari対応には -webkit-sticky プレフィックスの併記が推奨

スティッキーヘッダー

`position: sticky; top: 0;`

← スクロールしてみてください

コンテンツが流れても

ヘッダーは固定されます

これがposition: stickyの魅力です

コンテナの範囲内でのみ有効

```
/* 最小限の実装 */
.sticky-element {
  position: -webkit-sticky; /* Safari対応 */
  position: sticky;
  top: 0; /* 画面上端で固定開始 */
  z-index: 100; /* 重なり順の指定 */
}
```

`top: 0` ビューポート上端で固定（ヘッダー向き）

`bottom: 0` ビューポート下端で固定（フッター向き）

`left: 0` ビューポート左端で固定（サイドバー向き）

ヘッダーが固定されるイメージ `position: sticky; top: 0;`

5 活用例① ヘッダー固定

ヘッダーをスクロールに応じて画面上部に固定する実装は、**ナビゲーションの利便性**を大きく向上させます。

実装のポイント

`top: 0` で画面上部に固定

`background-color` で下のコンテンツが透けないよう対策

`z-index` で他要素より前面に表示

デザインのコツ

`box-shadow` でヘッダーと本文の境界を明確に

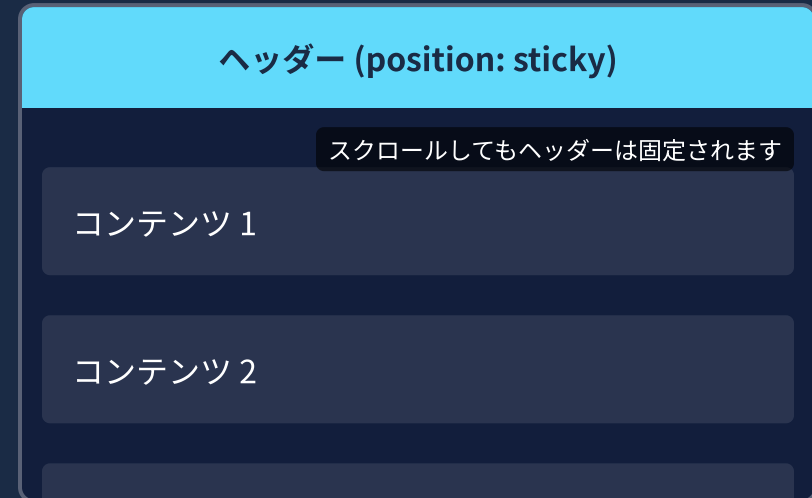
ファーストビューでは通常表示され、自然な動きに

使用シーン

- コーポレートサイト・ポートフォリオ
- ECサイト（検索バーやカートボタンの固定）
- 長文コンテンツのブログ

📖 レスポンシブ対応では、モバイル時のヘッダーの高さを調整するとよいでしょう

`z-index: 1000;`



```
/* HTML */
<header class="site-header">
  /* ヘッダーコンテンツ */
</header>
<main>
  /* メインコンテンツ */
</main>

/* CSS */
.site-header {
  position: -webkit-sticky; /* Safari対応 */
  position: sticky;
  top: 0;
  width: 100%;
  background-color: #ffffff;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  z-index: 1000;
}
```

※ スクロールするとヘッダーが自動的に画面上部に固定されます

CSS position: sticky 完全ガイド

2 活用例② サイドバー追従

2カラムレイアウトでのサイドバー固定は、ブログやドキュメントサイトでよく使われる実装です。

記事を読み進める間、目次や関連コンテンツを**常に表示**させておけるのが大きなメリットです。

実装ポイント

ネスト構造に注意：sticky要素の親要素の高さが十分あること

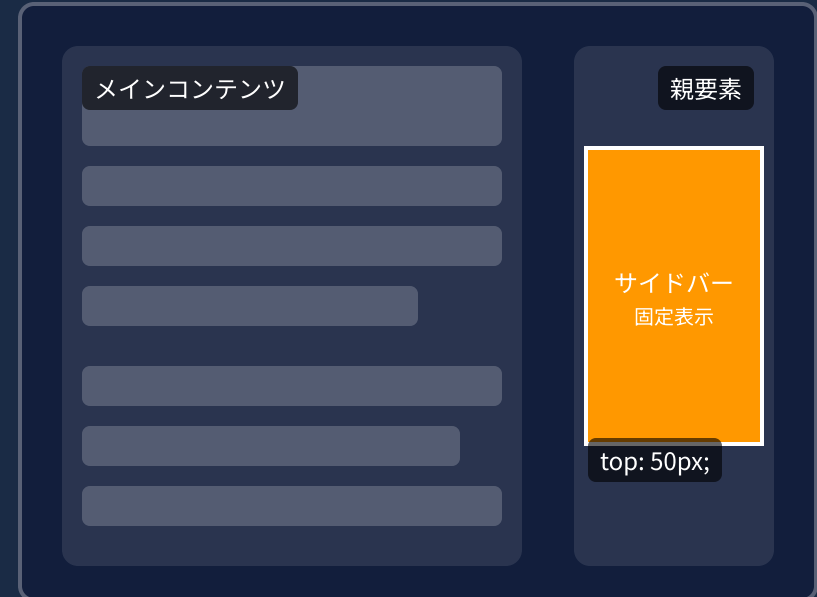
よくある問題

親要素がメインコンテンツより短いと途中で固定が解除されます

応用例

目次・関連記事・広告表示・検索フォームなど

```
position: sticky; top: 20px;
```



```
/* HTML構造 */
<div class="container">
  <main class="main-content">...</main>
  <aside class="sidebar">
    <div class="sidebar-inner">...</div>
  </aside>
</div>

/* CSS */
.container {
  display: flex;
  align-items: flex-start;
}
.main-content { flex: 3; }
.sidebar { flex: 1; }
.sidebar-inner {
  position: sticky;
  top: 20px;
}
```

※ メディアクエリを使って、モバイル表示では固定を解除するとよいでしょう

position: sticky; top: 0;

テーブルヘッダーが固定されるイメージ

3 テーブルヘッダー固定

長いテーブルでスクロールしても、**ヘッダー行が常に表示されたまま**になる実装方法です。

ユーザーが表を見やすくなり、データの列の意味を見失わずに済みます。

特徴1: コンテナにオーバーフロー設定

テーブルを囲むコンテナにoverflow-y: autoを指定

特徴2: ヘッダーにsticky設定

thead thにposition: sticky; top: 0;を指定

特徴3: 背景色設定が必須

ヘッダーにbackground-colorを指定しないと下の行が透けて見える

💡 **応用テクニック**：横スクロールが必要な広いテーブルでは、左列をposition: sticky; left: 0;で固定することも可能です。

商品名	価格	在庫状況	評価
商品A	¥2,500	在庫あり	★★★★☆
商品B	¥3,800	残りわずか	★★★★★
商品C	¥1,200	在庫あり	★★★☆☆
商品D	¥4,500	在庫なし	★★★★☆

```
/* HTML */
<div class="table-container">
  <table>
    <thead><tr><th>項目</th></tr></thead>
    <tbody><tr><td>データ</td></tr></tbody>
  </table>
</div>

/* CSS */
.table-container {
  max-height: 400px;
  overflow-y: auto;
}
thead th {
  position: -webkit-sticky;
  position: sticky;
  top: 0;
  background: #f2f2f2; /* 必須 */
  z-index: 10;
}
```

※ スクロールしてテーブル内容を確認すると、ヘッダーが固定されている様子がわかります

よくあるトラブルの解決法

8 よくある「効かない」原因と解決法

「コードは合っているはずなのに、なぜかstickyが効かない！」ときは、以下をチェック:

✔ top, bottom, left, rightの指定忘れ

position: stickyだけでは動作しません。必ず固定位置を指定しましょう。

⚠ 親要素にoverflow指定がある

最重要チェック項目! 親要素にoverflow: hidden/scroll/autoがあると機能しません。

✔ 親要素の高さが不足している

親要素の高さ=子要素の高さだと、スクロールする余地がなく固定されません。

✔ display: inlineになっている

インライン要素には効かない場合があります。ブロックレベル要素にしましょう。

✔ Safariへの対応が不足

-webkit-stickyプレフィックスを併記すると安全です。

⚠ overflow問題

開発者ツールで親要素を一つずつ遡り、overflowプロパティを確認

✔ 解決策

不要なoverflow指定を削除するか、HTML構造を見直す

⚠ 高さ不足の問題

親要素と子要素の高さが同じで固定されない

✔ 解決策

```
親要素にheightやmin-heightを指定するか、他の  
/* 両方のプレフィックスを指定してSafari対応 */  
sticky-element {  
  position: -webkit-sticky; /* Safari用 */  
  position: sticky;  
  top: 0;  
  z-index: 10;  
}
```

※ 効かない場合は、まず「topなどの指定」と「親のoverflow」を疑いましょう

overflow: visible !important;

CSS position: sticky 完全ガイド

9 ベストプラクティス & Tips

```
position: sticky; top: 0;
```

実践で活用するための重要なポイントとテクニックを紹介します



パフォーマンスへの配慮

- 必要最低限の要素にのみ `sticky` を使用する
- 固定要素内のコンテンツをシンプルに保つ
- `box-shadow` など描画負荷の高いスタイルは控えめに



レスポンス対応

- メディアクエリで画面サイズに応じた制御
- スマホではヘッダー高さを小さく調整
- 小画面では `sticky` を無効化する判断も



アクセシビリティ配慮

- ページ内リンクのジャンプ先に `scroll-margin-top` を指定
- 固定ヘッダーが重要コンテンツを隠さないよう注意
- キーボードフォーカスの視認性を確保



ブラウザ対応とTips

- Safariには `-webkit-sticky` プレフィックスを併記
- `z-index` を適切に設定して重なり順を制御
- Internet Explorerは非対応（代替案検討が必要）

```
/* レスポンス対応 + Safari対応 */
@media (min-width: 768px) {
  .site-header {
    position: -webkit-sticky; /* Safari対応 */
    position: sticky;
    top: 0;
    z-index: 100;
  }
}

/* アクセシビリティ向上：ジャンプ先の調整 */
.section {
  position: sticky; bottom: 0;
  scroll-margin-top: 60px; /* ヘッダー高さ分 */
}
```

10 まとめ・さらなる学び

position: sticky の重要ポイントを再確認しましょう：

コンテナの範囲内でのみ固定

親要素がビューポートから外れると固定が解除される

top/bottom/left/right の指定が必須

固定する位置を明示的に指定する必要がある

親要素のoverflow設定に注意

overflow: hiddenなどが指定されていると機能しない

ブラウザ対応に配慮

Safari対応には-webkit-stickyプレフィックスを追加

トラブル時のチェックポイント：

- ✓ 位置指定プロパティ(top等)の確認
- ✓ 親要素のoverflow設定をチェック
- ✓ コンテナの高さは十分か

● Flexbox

一次元レイアウトの習得で柔軟なデザインを実現

● CSS Grid

二次元レイアウトで複雑な構造も簡単に

● CSS変数

再利用可能なスタイル設定でメンテナンス性向上

● アニメーション

動きのあるインタラクションでUX向上

```
/* モダンなレイアウト技術の組み合わせ */
.container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
}
.sidebar {
  position: sticky;
  top: 20px;
}
```

z-index: 100;

※ モダンなCSSテクニックを組み合わせることで、より高度なレイアウトが実現できます