汎用性に優れている、超定番のjQueryベースのスライダーライブラリといえば、「[slick](http://kenwheeler.github.io/slick/)」、「[bxSlider](https://bxslider.com/" \t "_blank)」、「[FlexSlider](http://flexslider.woothemes.com/" \t "_blank)」 あたりでしょうか。

しかし、これらの名だたるライブラリよりも、総合的に機能が豊富でカスタマイズ性にも富み、しかもjQueryに依存しないJavaScriptスライダーライブラリとして人気を集めているのが、「[Swiper.js](http://idangero.us/swiper/" \t "_blank)」です。  
  
試してみたところ、モバイルデバイスとレスポンシブ表示にも対応しているだけでなく、パラメータが豊富で非常に凝ったスライドショーでもあまりに簡単に実装できる優れものだったので、いくつかのサンプルと共にご紹介します。

INDEX [[表示](https://digipress.info/tech/introducing-swiper-js/)]

**Swiper.js の基本的な使い方**

Swiper.js を使用するには、headセクション内にCSSと、</body>の前にメインのJavaScriptファイルを読み込んでおきます。  
ライブラリ自体をダウンロードするか、CDN経由でも利用できます。

[Swiper CDN](https://cdnjs.com/libraries/Swiper)

**ライブラリの読み込み**

**CSS**

1. <head>
2. ...
3. <link rel="stylesheet" href="path/to/swiper.min.css" />
5. <!-- CDNの場合 -->
6. <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/Swiper/4.0.7/css/swiper.min.css" />
7. </head>

**JavaScript**

1. <body>
2. ...
3. <script src="path/to/swiper.min.js"></script>
5. <!-- CDNの場合 -->
6. <script src="https://cdnjs.cloudflare.com/ajax/libs/Swiper/4.0.7/js/swiper.min.js"></script>
7. </body>

**スライダーのHTML**

基本となるHTML構成は、以下のようにします。

1. <!-- スライダー全体を括るメインコンテナ -->
2. <div class="swiper-container">
3. <!-- 全スライドをまとめるラッパー -->
4. <div class="swiper-wrapper">
5. <!-- 各スライド -->
6. <div class="swiper-slide">Slide 1</div>
7. <div class="swiper-slide">Slide 2</div>
8. <div class="swiper-slide">Slide 3</div>
9. ...
10. </div>
12. <!-- ページネーションを表示する場合 -->
13. <div class="swiper-pagination"></div>
15. <!-- 前後スライドへのナビゲーションボタン(矢印)を表示する場合 -->
16. <div class="swiper-button-prev"></div>
17. <div class="swiper-button-next"></div>
19. <!-- スクロールバーを表示する場合 -->
20. <div class="swiper-scrollbar"></div>
21. </div>

スライドに画像やテキストなどを表示する場合は、.swiper-slide内にそれらの HTML を記述するのみです。

**CSS (スライダーのサイズを指定)**

必要なレイアウトは、ベースとなる swiper.min.css のみで配置してくれるので、ユーザー側で用意する最低限必要な CSS は、スライダー全体のサイズ(width/height)のみです。

1. .swiper-container {
2. width: 600px;
3. height: 300px;
4. }

フルスクリーン表示にするのであれば、以下のようにするだけです。

1. .swiper-container {
2. width: 100%;
3. height: 100vh;
4. }

**JavaScript (スライダーを生成)**

あとは、必要なパラメータをまとめて、Swiperオブジェクトを生成するのみでスライダーが表示されます。

1. var mySwiper = new Swiper ('.swiper-container', {
2. // オプションパラメータ(一部のみ抜粋)
3. loop: true, // 最後のスライドまで到達した場合、最初に戻らずに続けてスライド可能にするか。
4. speed: 600, // スライドが切り替わるトランジション時間(ミリ秒)。
5. slidesPerView: 4, // 何枚のスライドを表示するか
6. spaceBetween: 10, // スライド間の余白サイズ(ピクセル)
7. direction: 'horizontal', // スライド方向。 'horizontal'(水平) か 'vertical'(垂直)。effectオプションが 'slide' 以外は無効。
8. effect: 'slide', // "slide", "fade"(フェード), "cube"(キューブ回転), "coverflow"(カバーフロー) または "flip"(平面回転)
10. // スライダーの自動再生
11. // autoplay: true 　のみなら既定値での自動再生
12. autoplay: {
13. delay: 3000, // スライドが切り替わるまでの表示時間(ミリ秒)
14. stopOnLast: false, // 最後のスライドまで表示されたら自動再生を中止するか
15. disableOnInteraction: true // ユーザーのスワイプ操作を検出したら自動再生を中止するか
16. },
18. // レスポンシブ化条件
19. breakpoints: {
20. // 980ピクセル幅以下になったら
21. 980: {
22. slidesPerView: 3,
23. spaceBetween: 30
24. },
25. // 640ピクセル幅以下になったら
26. 640: {
27. slidesPerView: 2,
28. spaceBetween: 20
29. }
30. },
32. // ページネーションを表示する場合
33. pagination: {
34. el: '.swiper-pagination',　 // ページネーションを表示するセレクタ
35. },
37. // 前後スライドへのナビゲーションを表示する場合
38. navigation: {
39. nextEl: '.swiper-button-next', // 次のスライドボタンのセレクタ
40. prevEl: '.swiper-button-prev', // 前のスライドボタンのセレクタ
41. },
43. // スクロールバーを表示する場合
44. scrollbar: {
45. el: '.swiper-scrollbar', // スクロールバーを表示するセレクタ
46. }
47. });

他にもたくさんのパラメータがあるので、詳しくは公式ドキュメントを参照してください。  
[Swiperパラメータ一覧](http://idangero.us/swiper/api/#parameters)

**Swiper.js イベントリスナー**

Swiper.js のイベントも豊富に用意されており、スライダーを生成後に、on メソッドでリスナーを登録できます。  
なお、イベント関数内で実行中の対象スライダーインスタンスを取得するには、thisキーワードを指定します。

1. // スライダーを生成
2. var mySwiper = new Swiper('.swiper-container', {
3. init: false, // 初期化はしない
4. // その他のパラメータ
5. ...
6. });
7. // スライダーの初期化(実行)
8. mySwiper.init();

11. // スライダーが初期化されたとき
12. mySwiper.on('init', function () {
13. console.log('initialized');
14. });
16. // スライドが切り替わったとき
17. mySwiper.on('slideChange', function () {
18. console.log('slide changed');
19. });
21. // 切り替わりのアニメーションが開始されたとき
22. mySwiper.on('slideChangeTransitionStart', function () {
23. console.log('transition start');
24. });
26. // 切り替わりのアニメーションが終了したとき
27. mySwiper.on('slideChangeTransitionEnd', function () {
28. // this キーワードで、対象のswiperインスタンスにアクセス可能
29. console.log('transition end : active is ' + this.activeIndex);
30. });
32. // スライドの画像の読み込みが完了したとき(※スライド内にimgタグがある場合のみ)
33. mySwiper.on('imagesReady', function () {
34. console.log('images ready');
35. });

イベントリスナーは、on メソッドではなく、スライダーの onパラメータ で定義すれば、スライダーの生成と同時にイベントを登録しておくことができ、より簡潔なコードになります。

1. var mySwiper = new Swiper ('.swiper-container', {
2. ...
3. // イベント
4. on: {
5. init: function(){
6. // 任意の処理を実行
7. },
8. slideChange: function(){
9. // 任意の処理を実行
10. },
11. slideChangeTransitionStart: function(){
12. // 任意の処理を実行
13. },
14. slideChangeTransitionEnd: function(){
15. // 任意の処理を実行
16. },
17. imagesReady: function(){
18. // 任意の処理を実行
19. }
20. }
21. });

その他の Swiper.js のイベントについては公式ドキュメントをご覧ください。  
[Swiperイベント一覧](http://idangero.us/swiper/api/#events)

**スライド内コンテンツにパララックス効果を与える**

Swiper.js では、パラメータを追加するだけで、スライド(.swiper-slide)内に表示するテキストなどの任意のHTML要素にパララックス効果を施すこともできます。

**パララックス効果を有効にする**

parallax:true をオプションに含めるだけで、スライドが切り替わる際にスライド内の任意のコンテンツがパララックス効果を伴って表示されます。

1. var options = {
2. ...
3. parallax: true
4. };
5. var mySwiper = new Swiper('.swiper-container', options);

**スライド内の要素にパララックス効果を定義する**

パララックス効果を施すスライド内の要素には、data属性でスライドが切り替わる前の状態(移動距離、スケール、不透明度)と、元に状態に表示されるまでのアニメーション時間を指定しておきます。

**パララックス効果用のdata属性**

* data-swiper-parallax  
  スライド方向に沿った表示前の移動距離を、数値(ピクセル)かパーセンテージ(%)で指定します。  
  例 : data-swiper-parallax=”-80″, data-swiper-parallax=”-30%”
* data-swiper-parallax-x  
  水平方向の表示前の移動距離を、数値(ピクセル)かパーセンテージ(%)で指定します。
* data-swiper-parallax-y  
  垂直方向の表示前の移動距離を、数値(ピクセル)かパーセンテージ(%)で指定します。
* data-swiper-parallax-scale  
  表示前の拡大率を指定します。
* data-swiper-parallax-opacity  
  表示前の不透明度(0〜1)を指定します。
* data-swiper-parallax-duration  
  パララックス効果のアニメーション時間をミリ秒で指定します。

1. <div class="swiper-wrapper">
2. <div class="swiper-slide">
3. <!-- 移動距離の指定 -->
4. <div class="title" data-swiper-parallax="-100">Slide 1</div>
5. <div class="subtitle" data-swiper-parallax="-200">Subtitle</div>
6. <!-- 移動距離とアニメーション時間 -->
7. <div class="text" data-swiper-parallax="-300" data-swiper-parallax-duration="600">
8. <p>Lorem ipsum dolor sit amet, ...</p>
9. </div>
10. <!-- 不透明度の指定 -->
11. <div data-swiper-parallax-opacity="0.5" >I will change opacity</div>
12. <!-- 拡大率の指定 -->
13. <div data-swiper-parallax-scale="0.15" >I will change scale</div>
14. </div>
15. ...
16. </div>

Swiper.js の基本的な仕組みと使い方が理解できたところで、Swiperだから簡単に実装できる少し凝ったスライダーを作ってみましょう。

**Swiper.js で簡単に作れるスライダーのサンプル**

**カバーフロースライダー**

中央にメインスライドを表示し、3次元効果でスライドが切り替わっていくカバーフローは、centeredSlides:trueとeffect:'coverflow'　というパラメータを指定するだけで表現できます。

このサンプルでは、画像がすべて読み込まれてからスライダーをフェードインで表示させるため、コンテナ要素(.swiper-container)は見えないようにしておき、SwiperのimagesReady イベントで画像の読み込みが完了したら、コンテナ要素から非表示用の.loadingセレクタを削除してスライダーが表示されるようにしています。

**SwiperコンテナのHTML**

1. <section class="swiper-container loading">
2. ...
3. </section>

**imagesReadyイベント**

1. var options = {
2. effect: 'coverflow',
3. ...
4. on: {
5. imagesReady: function(){
6. this.el.classList.remove('loading');
7. }
8. }
9. var mySwiper = new Swiper('.swiper-container', options);

ちなみに、このサンプルでは実際に表示される画像はスライド要素(.swiper-slide)の背景画像(background-image)であり、img 要素は imagesReady イベントで画像ファイルの読み込み完了を検知するためだけに背景画像と同じURLを指定して、これ自体はCSS で非表示(display:none)にしています。

**3D キューブ回転スライダー**

スライドを立体に見えるように配置し、回転するキューブ状のスライダーを表示するには、effect:'cube' というパラメータを付けます。

このサンプルでは、キューブが自動または手動で回転する際に、スライドが見切れないよう、Swipeのイベントを利用してトランジションの開始時に縮小し、トランジション終了後に元のスケールに戻すようにしています。

**イベント処理**

1. var isMove = false,
2. options = {
3. ...
4. // イベント
5. on: {
6. ...
7. // ユーザー操作によりスライドが移動されたとき
8. touchMove: function(event){
9. if (!isMove) {
10. this.el.classList.remove('scale-in');
11. this.el.classList.add('scale-out');
12. isMove = true;
13. }
14. },
15. // ユーザー操作が解除されたとき
16. touchEnd: function(event){
17. this.el.classList.remove('scale-out');
18. this.el.classList.add('scale-in');
19. setTimeout(function(){
20. isMove = false;
21. }, 300);
22. },
23. // トランジション開始時
24. slideChangeTransitionStart: function(){
25. if (!isMove) {
26. this.el.classList.remove('scale-in');
27. this.el.classList.add('scale-out');
28. }
29. },
30. // トランジション終了時
31. slideChangeTransitionEnd: function(){
32. if (!isMove) {
33. this.el.classList.remove('scale-out');
34. this.el.classList.add('scale-in');
35. }
36. }
37. }
38. };

**水平パララックス効果を伴うスライダー**

最後は、スライドが切り替わる際、スライド内に表示させている背景画像要素を微妙に移動(translate)させることで、スライド自体の移動の動きと背景画像の動きにずれを生じさせて視差(パララックス)効果を施したサンプルです。

各スライドのトランジション時の進捗状況は、watchSlidesProgress:true というパラメータを指定し、progressイベントで受け取ることができます。  
さらに、touchStart イベントでスライドがクリック(タップ)された際にはすべてのスライド要素の transition を一旦クリアし、setTransition イベントでトランジションアニメーションが開始された際に、トランジション時間(transition-duration)を受け取り、スライド要素(.swiper-slide)と背景画像要素(.slide-bgimg)に transition を再びセットします。

このサンプルでは、切り替わる際の動きはスライド自体とは別で処理させるため、スライド要素(.swiper-slide)に background-image は指定せず、その配下に別途背景画像用の要素(figure)を設けています。

1. <div class="swiper-slide">
2. <figure class="slide-bgimg" style="background-image:url(画像URL)">
3. <img src="画像URL" class="entity-img" />
4. </figure>
5. </div>

**参考**

<https://github.com/nolimits4web/Swiper/issues/1497>

さらに、メインスライダーとは別にページネーション用に同じスライド数のスライダーを用意し、slideToClickedSlide:true パラメータによって選択されたスライドへの移動を有効にし、双方が制御関係にあるように紐付けることで、サムネイルナビゲーション付きのスライダーができあがります。

1. mainSlider.controller.control = navSlider;
2. navSlider.controller.control = mainSlider;

**data-swiper-parallax 属性による水平パララックス効果を表現した例**

**2019/4/9:追記**

Swiper.js がサポートするパララックスコンポーネントを利用すれば、対象要素に data-swiper-parallax 属性を与えるだけで、上記サンプルの progress イベントなどを利用して JavaScript を組まなくても、より簡素なコードで上記同様のパララックススクロールで切り替わるスライドを表示できます。  
[パララックスコンポーネントについてはこちら](http://idangero.us/swiper/api/#parallax)

Swiper.js を利用したスライダーのサンプルをご紹介しましたが、Swiper.js はjQueryが不要であるだけでなく、特に3次元のトランジション動作を施したスライダーでも非常に簡単に実装できることが、定番のSlickやbxSliderなどと比べても優れていると感じました。

以前公開した「[slick.js に動画スライドを含めて自動再生するスライダーの実装方法](https://digipress.info/tech/how-to-include-youtube-vimeo-video-with-autoplay-slick-js/)」の内容を、Swiper.js に移行することも容易だと思いますので、今後リリースするWordPressテーマには、Swiper.js をベースにした画像と動画(YouTube, Vimeo, HTML5 Video)の混在スライダーを組み込んでみたいと思います。

ちなみに、[slick.js に動画スライドを含めて自動再生するスライダーの実装方法](https://digipress.info/tech/how-to-include-youtube-vimeo-video-with-autoplay-slick-js/)のサンプルコードは、記事公開後に、ある業者が販売しているWordPressテーマのトップページ用のスライダーに必要な修正もなくほぼそのまま転用された状態で組み込まれ、販売されていました。。  
あくまでサンプルで公開したもので汎用性や完全性は何も考慮していないのですが、そのテーマでは関数やセレクタ名もサンプルそのままで、インラインで未圧縮のまま日本語のコメントだらけのコードだったので、何かバッティングやAPIの仕様変更が起きた際にはちゃんと処理を理解して対応する意志があるといいのですが。。