

HTML5で作成する時の必要最小限のテンプレート

まずは、HTML5でWebページを作成する時にのシンプルなブランクのテンプレート。

- [A Generic HTML5 Template -GitHub](#)

ライセンスは[CC0 1.0](#)で、個人でも商用でもクライアントの案件でも無料で利用できます。利用する際に、許可やコピーライトの明記などは必要ありません。

テンプレートはIE9や8以下への最低限の配慮がとられています。

```
HTML5で作成する時の必要最小限のテンプレート
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta http-equiv="X-UA-Compatible" content="IE=edge">
5  <meta charset="utf-8">
6  <title></title>
7  <meta name="description" content="">
8  <meta name="author" content="">
9  <meta name="viewport" content="width=device-width, initial-scale=1">
10 <link rel="stylesheet" href="">
11 <!--[if lt IE 9]>
12 <script src="//cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.2/html5shiv.min.js"></script>
13 <script src="//cdnjs.cloudflare.com/ajax/libs/respond.js/1.4.2/respond.min.js"></script>
14 <![endif]-->
15 <link rel="shortcut icon" href="">
16 </head>
17 <body>
18
19 <!-- コンテンツを配置 -->
20
21 <!-- SCRIPTS -->
22 <!-- Example: <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
23 </body>
24 </html>
```

上記の各コードについて説明した、コメント版。

※コメントは訳したもので、一部、補足を加えました。

```
HTML5で作成する時の必要最小限のテンプレート（コメント版）
1  <!DOCTYPE html>
2  <!-- 互換性とアクセシビリティのために<html>タグの「lang=""」属性をセット。
3  HTMLタグのほかの属性については下記ページを参考に。
4  参考: http://www.w3.org/TR/html-markup/global-attributes.html -->
5  <html>
6
7  <head>
8
9  <!-- IE8+に対して「IE=edge」と指定することで、
10  利用できる最も互換性の高い最新のエンジンを使用するよう指示できます。
11  参考: https://www.modern.ie/en-us/performance/how-to-use-x-ua-compatible -->
12 <meta http-equiv="X-UA-Compatible" content="IE=edge">
13
14 <!-- パフォーマンスのために使用する文字のエンコーディングを記述。
15  最初のオプションとしてHTTPヘッダで指定し、
16  HTTPヘッダとmetaタグでの指定が同じであることを確認。
17  参考: https://developers.google.com/speed/docs/best-practices/rendering#SpecifyCharsetEarly -->
18 <meta charset="utf-8">
19
20 <!-- ページのタイトルを記述 -->
21 <title></title>
22
23 <!-- Scontent属性にページの紹介文を記述 -->
24 <meta name="description" content="">
25
```

```

26 <!-- content属性にページの著者情報を記述 -->
27 <meta name="author" content="">
28
29 <!-- モバイル端末への対応、
30 ページをビューポートの幅に合わせてレンダリング（Android, iOS6以降）
31 ズームを許可しない設定「user-scalable=no」は加えない -->
32 <meta name="viewport" content="width=device-width, initial-scale=1">
33
34 <!-- スタイルシートはできるだけ早くレンダリングされるため、
35 HTMLドキュメントの上の方に記述
36 href属性にスタイルシートファイルのURIを記述 -->
37 <link rel="stylesheet" href="">
38
39 <!-- IE8以下用に2つのスクリプトを記述
40     html5shiv.js: IE8以下にHTML5の要素を認識するようにさせる
41     respond.js: IE8以下にMedia Queriesの代替機能を提供 -->
42 <!--[if lt IE 9]>
43 <script src="//cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.2/html5shiv.min.js"></script>
44 <script src="//cdnjs.cloudflare.com/ajax/libs/respond.js/1.4.2/respond.min.js"></script>
45 <![endif]-->
46
47 <!-- href属性にファビコンファイルのURIを記述 -->
48 <link rel="shortcut icon" href="">
49
50 <!-- コメントアウトしてあるコードは、iOS/Android用のアイコン指定 -->
51 <!--
52 <meta name="mobile-web-app-capable" content="yes">
53 <link rel="icon" sizes="196x196" href="">
54 <link rel="apple-touch-icon" sizes="152x152" href="">
55 -->
56
57 <!-- スクリプトでブロッキングを起こさないものはここに記述
58 可能であれば「async（文書の読み込みが完了した時点でスクリプトを実行）」を使用
59 例: <script src="" async></script> -->
60 </head>
61 <body>
62
63 <!-- コンテンツを配置 -->
64
65 <!-- スクリプトでブロッキングを起こすものはここに記述
66 ブロッキングを起こす原因としては、
67 CSSのセレクタ操作（IE）、負荷の高いDOM操作、多数のスクリプトなど -->
68 <!-- SCRIPTS -->
69 <!-- 例: <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script> -->
70 </body>
71 </html>

```

補足：スクリプトの配置

スクリプトはページのレンダリングをブロックするため、ドキュメントの一番下、</body>の上に記述することは良い習慣です。これだけでWebページのパフォーマンスはだいぶ改善されます。

しかし、一番下に記述することは常に可能というわけではありません。例えばページのレンダリングに関するスクリプトは上に記述しておかないとページのレンダリングが遅くなります。そういった場合は可能であれば、「async」を加え並列にロードするように指定しておきます。

スクリプトの配置	
1	<script src="my-script.js" async></script>

補足：metaタグのviewportの指定

スマホなどの小さいサイズのスクリーン用に下記を記述します。

metaタグのviewportの指定	
1	<meta name="viewport" content="width=device-width, initial-scale=1">

この指定は、ページをデバイスのビューポートの幅に合わせて、コンテンツの幅を調整します。

補足：IE9と以下のサポート

IEも最新のバージョンを使用するようにさまざまなキャンペーンを行っていますが、実際にはIE9や8以下を使用する人がまだ多くいます。HTML5の新しい要素とMedia Queriesは、html5shiv.jsとrespond.jsを加えることで問題ないでしょう。テンプレートでは信頼できる公共のCDNを利用しています。また条件付きコメントを使い、IE8以下のみロードするようにしています。

IE用にはもう一つ、次のmetaタグを加えています。

```
IE9と以下のサポート
1 <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

これはIEで利用できる最も互換性の高い最新のエンジンを使用するための指定です。古いIEを使用するユーザーがいなくなった時に、こういった固有のブラウザ用の記述がテンプレートから取り除かれることを望みます。

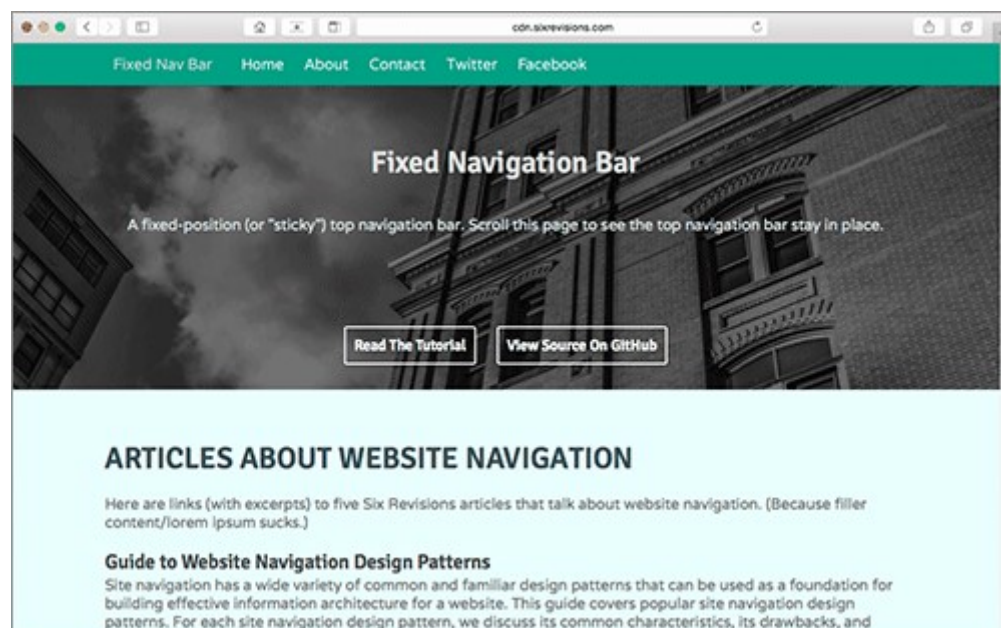
現状としては、少し余分のマークアップを加えるだけで、IE8以下でもHTML5の要素とMedia Queriesを利用できるということは、合理的な妥協点でしょう。

ナビゲーションを上部に固定配置

最近多く見かける、ナビゲーションのバーをスクロールしても上部に固定させるテンプレートです。

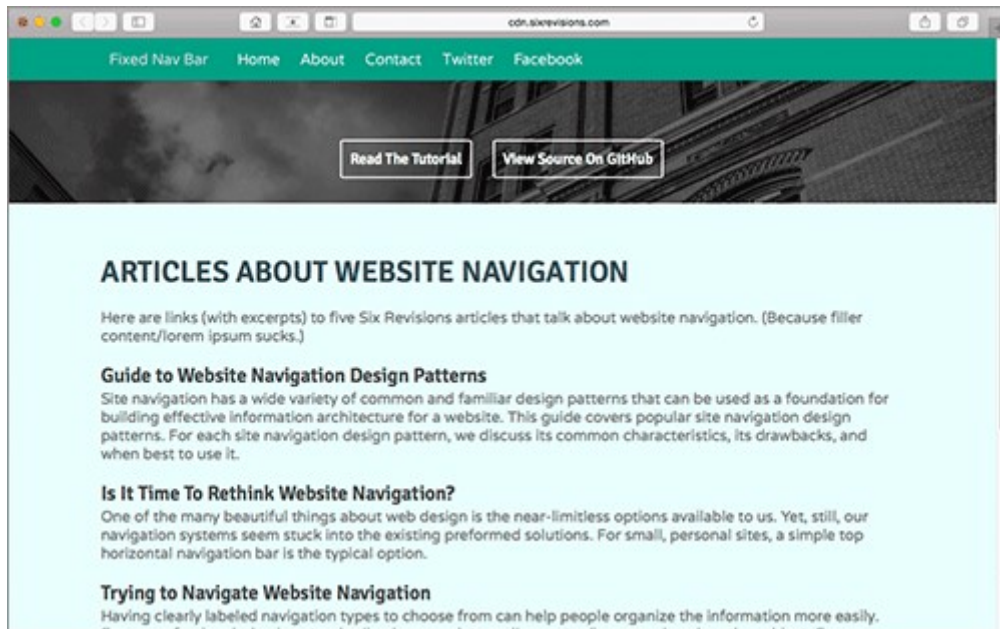
- [Fixed Navigation Bar](#)

ライセンスは[CC0 1.0](#)で、個人でも商用でもクライアントの案件でも無料で利用できます。利用する際に、許可やコピーライトの明記などは必要ありません。



デモページ

スクロールしても、上部に固定配置されています。



[デモページ](#): 少しスクロールしたところ

HTML

nav要素と「.fixed-nav-bar」が最小構成です。この中にロゴやメニューを配置します。

```
ナビゲーションを上部に固定配置
1 <nav class="fixed-nav-bar">
2   <!-- ナビゲーションバーに配置するコンテンツ -->
3 </nav>
```

ナビゲーションバーに配置するコンテンツは、例えばこんな感じに。

```
ナビゲーションバーに配置するコンテンツの例
1 <nav class="fixed-nav-bar">
2   <div id="menu" class="menu">
3     <a class="sitename" href="fixed-navigation-bar.html">Fixed Nav Bar</a>
4     <!-- Example responsive navigation menu -->
5     <a class="show" href="#menu">Menu</a><a class="hide" href="#hidemenu">Menu</a>
6     <ul class="menu-items">
7       <li><a href="//sixrevisions.com">Home</a></li>
8       <li><a href="//sixrevisions.com/about/">About</a></li>
9       <li><a href="//sixrevisions.com/contact/">Contact</a></li>
10      <li><a href="//twitter.com/sixrevisions">Twitter</a></li>
11      <li><a href="//www.facebook.com/sixrevisions">Facebook</a></li>
12    </ul>
13  </div>
14 </nav>
```

CSS

```
ナビゲーションを上部に固定配置
1 .fixed-nav-bar {
2   position: fixed;
3   top: 0;
4   left: 0;
5   z-index: 9999;
6   width: 100%;
7   height: 50px;
8   background-color: #00a087;
9 }
```

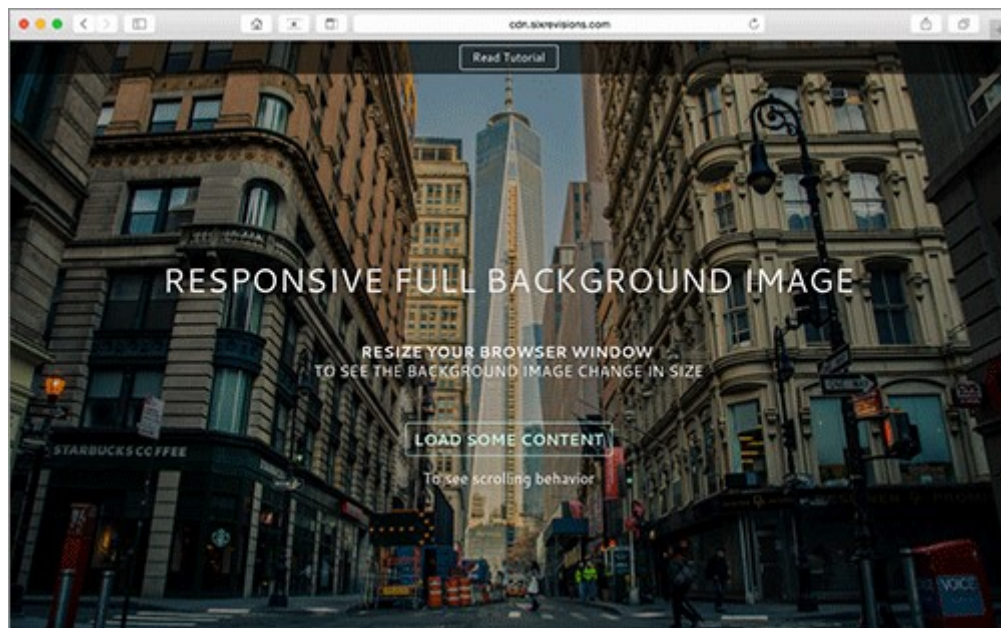
背景画像をブラウザいっぱいに表示

背景画像をブラウザいっぱいに表示します。

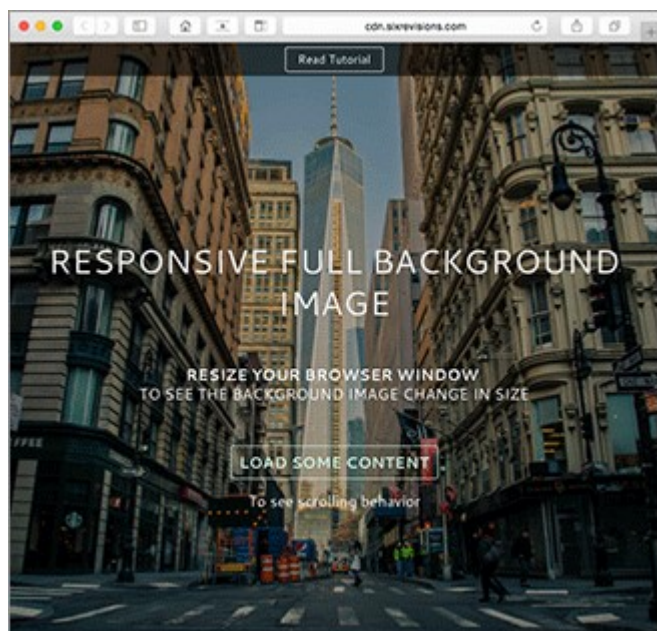
デスクトップだけでなくスマホやタブレットにも対応、しかもCSSはたった一行で記述できるので簡単便利です。

- [Responsive Full Background Image](#)

ライセンスは[CC0 1.0](#)で、個人でも商用でもクライアントの案件でも無料で利用できます。利用する際に、許可やコピーライトの明記などは必要ありません。



[デモページ](#): 幅1,200pxで表示



[デモページ](#): 幅780pxで表示



[デモページ](#): 幅480pxで表示

HTML

必要最小限のHTMLで、コンテンツは自由に配置できます。

背景画像をブラウザいっぱいに表示

```
1 <!doctype html>
2 <html>
3 <body>
4   <!-- コンテンツを配置 -->
5 </body>
6 </html>
```

ブラウザのビューポート全体をカバーするように画像を配置するのは、bodyに背景画像として表示します。このテクニックはbodyだけでなく、div, article, form などのブロックレベル要素にも利用することができます。

CSS

背景を表示するために、bodyにスタイルシートを定義します。

背景画像をブラウザいっぱいに表示

```
1 body {
2   /* 画像ファイルの指定 */
3   background-image: url(images/background-photo.jpg);
4
5   /* 画像を常に天地左右の中央に配置 */
6   background-position: center center;
7
8   /* 画像をタイル状に繰り返し表示しない */
9   background-repeat: no-repeat;
10
11  /* コンテンツの高さが画像の高さより大きい時、動かないように固定 */
12  background-attachment: fixed;
13
14  /* 表示するコンテナの大きさに基づいて、背景画像を調整 */
15  background-size: cover;
16
17  /* 背景画像が読み込まれる前に表示される背景のカラー */
18  background-color: #464646;
19 }
```

「background-size: cover;」のプロパティと値の組み合わせは要素の幅と高さに対して画像を相対的に調整します。また、画像がビューポートの天地左右の中央に配置されるように「background-position: center center;」も忘れな

いように指定します。

一行のスタイルシートで記述

上記はすべて「background」の指定のため、ショートハンドを使って短く記述できます。

背景画像をブラウザいっぱいに表示

```
1 body {  
2   background: url(background-photo.jpg) center center / cover no-repeat fixed;  
3 }
```

オプション：スマホに最適化

スマホやタブレットなどの小さいスクリーンサイズ用にはMedia Queriesを使って、より小さいサイズの画像を背景に表示することもできます。

スマホに最適化

```
1 @media only screen and (max-width: 767px) {  
2   body {  
3     background-image: url(images/background-photo-mobile-devices.jpg);  
4   }  
5 }
```

IE8以下にも対応したい場合

IE8以下は「background-size」に対応していません。しかしIE8以下にも、いやIE6/7/8にも対応したい時は下記のスクリプトを使用すると、IE6/7/8でも利用できます。

- [backgroundSize.js -GitHub](#)