

fullpage学习笔记

- 作者：周正
- 时间：2020.3.19
- 邮箱：zhouzheng5210@163.com

✓ 什么是fullpage?

fullPage.js 是一个基于 jQuery 的插件，它能够很方便、很轻松的制作出全屏网站。

如今我们经常能见到**全屏网站**，尤其是国外网站。这些网站用几幅很大的图片或色块做背景，再添加一些简单的内容，显得额外的高端大气上档次。如果你也希望你的网站能设计成全屏的，显得更上档次，你可以试试 fullPage.js。

fullPage.js的主要功能有：支持鼠标滚动、支持前进后退和键盘控制、多个回调函数、支持手机、平板触摸事件、支持 CSS3 动画、支持窗口缩放、窗口缩放时自动调整、可设置滚动宽度、背景颜色、滚动速度、循环选项、回调、文本对齐方式等等。

目前主流的全屏网站开发方式：

- ✓ swiper
- ✓ fullpage

注：

全屏滚动效果，原生js也很好实现，主要是用 mousewheel 鼠标滚轮滚动事件，来判断上滚动还是下滚动，之后设置每次滚动的高度为屏幕的高度即可。但是，虽然效果简单，但是兼容性很差，要做很多兼容处理及比较麻烦啦！（了解一下，有兴趣的同学可以研究一下）



经典案例实例：

[百度百科史记](#)

[360安全路由](#)

中文网站：<http://fullpage.81hu.com/>

✓ 环境搭建

下载插件

需要准备的东西：

- jquery
- fullpage.js
- easing.js (非必要)
- fullpage.css
- [animate.css](#)

注意：因为fullpage是基于jquery的，因此fullpage.js一定要在jquery之后引入。

下载地址：

(1) github: <https://github.com/alvarotrigo/fullPage.js>

(2) 官网: <https://alvarotrigo.com/fullPage/zh/#page1>

也可以使用一些开源的cdn。

引入文件

```
1 <link rel="stylesheet" href="./css/fullpage.css">
2 <link rel="stylesheet" href="./css/my.css"> <!-- 自己的css -->
3 <link href="https://cdn.bootcdn.net/ajax/libs/animate.css/4.1.1/animate.compat.css"
  rel="stylesheet"><!-- animate.css -->
4 <script src="https://cdn.bootcdn.net/ajax/libs/jquery/3.5.1/jquery.js"></script> <!-- jquery-->
5 <script src="./js/fullpage.js"></script>
```

HTML结构

```
1 <div id="fullpage">
2   <div class="section">第一屏</div>
3   <div class="section">第二屏</div>
4   <div class="section">
5     <div class="slide">第三屏的第一屏</div>
6     <div class="slide">第三屏的第二屏</div>
7     <div class="slide">第三屏的第三屏</div>
8     <div class="slide">第三屏的第四屏</div>
9   </div>
10  <div class="section">第四屏</div>
11 </div>
```

初始化

```
1 | $(function(){
2 |   $('#fullpage').fullpage();
3 | });
```

到此为止，开发环境搭建完毕。

✓ 参数详解

选项	类型	默认值	说明
verticalCentered	字符串	true	内容是否垂直居中
resize	布尔值	false	字体是否随着窗口缩放而缩放
sectionColor	函数	无	设置背景颜色（ 值为一个数组 ）
anchors	数组	无	定义锚链接
scrollingSpeed	整数	700	滚动速度，单位为毫秒
easing	字符串	easeInQuart	滚动动画方式
menu	布尔值	false	绑定菜单，设定的相关属性与 anchors 的值对应后，菜单可以控制滚动
navigation	布尔值	false	是否显示项目导航

选项	类型	默认值	说明
navigationPosition	字符串	right	项目导航的位置，可选 left 或 right
navigationTooltips	数组	空	项目导航的 tip
slidesNavigation	布尔值	false	是否显示左右滑块的项目导航
slidesNavPosition	字符串	bottom	左右滑块的项目导航的位置，可选 top 或 bottom
controlArrowColor	字符串	#fff	左右滑块的箭头的背景颜色
loopBottom	布尔值	false	滚动到最底部后是否滚回顶部
loopTop	布尔值	false	滚动到最顶部后是否滚底部
loopHorizontal	布尔值	true	左右滑块是否循环滑动
autoScrolling	布尔值	true	是否使用插件的滚动方式，如果选择 false，则会出现浏览器自带的滚动条 (把这个属性设为false的纯属脑子进水了)
scrollOverflow	布尔值	false	内容超过满屏后是否显示滚动条

选项	类型	默认值	说明
css3	布尔值	false	是否使用 CSS3 transforms 滚动
paddingTop	字符串	0	与顶部的距离
paddingBottom	字符串	0	与底部距离
fixedElements	字符串	无	
normalScrollElements		无	
keyboardScrolling	布尔值	true	是否使用键盘方向键导航
touchSensitivity	整数	5	
continuousVertical	布尔值	false	是否循环滚动，与 loopTop 及 loopBottom 不兼容
animateAnchor	布尔值	true	
normalScrollElementTouchThreshold	整数	5	

下面挑选几个常用的参数进行讲解：

sectionColor

作用：设置每屏的颜色，参数为**数组**。

1 | `sectionsColor: ['#f00', '#4BBFC3', '#7BAABE', '#0f0', '#00f']`

scrollingSpeed

作用：设置滚动速度，单位为毫秒 默认值为700

loopHorizontal

作用：设置包含slide的轮播图是否能循环播放 默认为true

keyboardScrolling

作用：是否能用键盘控制滚动 默认为true

continuousVertical

作用：控制上下是否循环滚动，默认为false

navigation

作用：是否显示侧导航（小圆点）默认为false

navigationPosition

作用：控制侧点航（小圆点）的位置，默认为right

menu和anchors

作用：这两个参数用来快速实现fullpage的**导航**。他们是一套**组合技能**。这是最复杂也是最重要的两个参数。

menu:

默认值：false，一个选择器可以用来指定要与滚动互动的导航菜单，**有点类似与Bootstrap的滚动监听**。这样到滚动到某个section时，对应的菜单会被自动添加**active**类。

注意，这个选项不会自动生成一个导航菜单，**仅仅是给指定的菜单中当前菜单项添加一个active活动类**。

言外之意就是需要我们自定义active类里都有什么。

为了让自定义导航菜单和屏幕section互动，需要给菜单添加一个HTML5的自定义属性（**data-menuanchor**），需要和锚文本设置相同的内容，例如下面的示例代码：

```
1 <ul id="menu">
2   <li data-menuanchor="firstPage" class="active"><a href="#firstPage">First section</a></li>
3   <li data-menuanchor="secondPage"><a href="#secondPage">Second section</a></li>
4   <li data-menuanchor="thirdPage"><a href="#thirdPage">Third section</a></li>
5   <li data-menuanchor="fourthPage"><a href="#fourthPage">Fourth section</a></li>
6 </ul>
7
```

```
1 $('#fullpage').fullpage({
2   anchors: ['firstPage', 'secondPage', 'thirdPage', 'fourthPage', 'lastPage'],
3   menu: '#menu'
4 });
```

注意：注意这个自定义的**菜单代码应该放置到插件设置的内容外面**，避免因为排版不兼容问题可以使用css3:true，否则将被附加到body的插件本身。

anchors

默认值：[]，定义导航的锚文本信息例如（#example），每个导航文本之前用英文逗号（,）分隔，快速导航的锚文本URL也是使用的这个文本，浏览器通过此锚文本链接可以支持前进和后退按钮功能。

此选项的设置还可以作为用户的书签，当用户打开带有锚文本的URL时，Fullpage可以自动跳转到对应的section屏幕或者slide幻灯片位置。

注意，如果你使用了此选项，那么网页中不能有相同的ID，一来Fullpage插件无法准确的定位到section屏幕或者slide幻灯片位置，二来这也有悖网页中CSS的编写规范。

到此基本的参数讲解完毕。

✓ 回调函数

回调函数可以帮助我们实现一些动画效果。

首先我们先回顾一下实现动画有几种方式：

- css3
- animate
- 原生js（配合setInterval（）和setTimeout（）方法）
- jquery的animate()方法

名称	说明
afterLoad	滚动到某一屏后的回调函数，接收 anchorLink 和 index 两个参数，anchorLink 是锚链接的名称， index 是序号 ，从1开始计算
onLeave	滚动前的回调函数，接收 index 、 nextIndex 和 direction 3个参数：index 是离开的“页面”的序号，从1开始计算；nextIndex 是滚动到的“页面”的序号，从1开始计算；direction 判断往上滚动还是往下滚动，值是 up 或 down。
afterRender	页面结构生成后的回调函数，或者说页面初始化完成后的回调函数
afterSlideLoad	滚动到某一水平滑块后的回调函数，与 afterLoad 类似，接收 anchorLink、index、slideIndex、direction 4个参数
onSlideLeave	某一水平滑块滚动前的回调函数，与 onLeave 类似，接收 anchorLink、index、slideIndex、direction 4个参数

最为常用的两个方法 为afterLoad和onLeave

afterLoad初体验：

简单来说就是在某个页面滚动结束后，执行的回调函数。

```
1  afterLoad: function(anchor, MyIndex) {
2      var index = ++MyIndex.index;
3      console.log(index)
4      if (index == 1) {
5          alert(1)
6      }
7
8      if (index == 2) {
9          alert(2)
10     }
11
12     if (index == 3) {
13         alert(3)
14     }
15
16     if (index == 4) {
17         alert(4)
```

```
18     }
19
20     if (index == 5) {
21         alert(5)
22     }
23 }
```

注意：

可能是由于版本迭代的原因,现在的afterLoad回调函数的index参数并不是一个数值类型的值,而是一个对象, index值包含在了该对象中, 并且也不是从1开始的, 而是从0开始的。下面的onLeave函数同理。

这个坑卡了我很长时间。

onLeave初体验：

简单来说就是在此页面离开前执行的回调函数。

```
1  onLeave: function(Index, nextIndex, direction) {
2
3      var index = ++Index.index;
4      // console.log(index)
5      if (index == 1) {
6          alert(1)
7      }
8
9      if (index == 2) {
10         alert(2)
11     }
12
13     if (index == 3) {
14         alert(3)
15     }
16
17     if (index == 4) {
18         alert(4)
19     }
20
21     if (index == 5) {
22         alert(5)
23     }
24 }
```

大家可以自行输出一下这几个参数都是什么东西, 这里再做演示。

✓ fullPage.js 方法

注意方法的使用时需要添加：

`$.fn.fullpage` 比如

```
1 | $.fn.fullpage.moveTo(1);
```

名称	说明
<code>moveSectionUp()</code>	向上滚动
<code>moveSectionDown()</code>	向下滚动
<code>moveTo(section, slide)</code>	滚动到
<code>moveSlideRight()</code>	slide 向右滚动
<code>moveSlideLeft()</code>	slide 向左滚动
<code>setAutoScrolling()</code>	设置页面滚动方式，设置为 true 时自动滚动
<code>setAllowScrolling()</code>	添加或删除鼠标滚轮/触控板控制
<code>setKeyboardScrolling()</code>	添加或删除键盘方向键控制
<code>setScrollingSpeed()</code>	定义以毫秒为单位的滚动速度

这几个方法在此不做演示了，有兴趣的同学请自行查阅文档。

✓ 动画演示

使用fullpage制作的网页可以配合回调函数制作出许多精彩的动画效果，下面做简单演示。

基本思路：动画的脚本（js代码）都放置在fullpage的回调函数里，一般来说，如果使用css3的制作动画，通常都是预定

义一些类，**当滚动到某一页面时，通过加载这些类呈现动画效果**（使用afterLoad），同时为了每次滚动到此页面使都有

动画效果，在每次划出该页面时还需清空掉这些加载的类（使用onLeave）；当然也可以使用jquery的animate的动画队列等。

不过为了便捷开发，使用网上的一些插件是个不错的选择。例如animate.css。

最后附上本次demo的完整代码。

html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <link href="https://cdn.bootcdn.net/ajax/libs/animate.css/4.1.1/animate.compat.css"
  rel="stylesheet">
9   <link rel="stylesheet" href="/css/fullpage.css">
10  <link rel="stylesheet" href="/css/main.css">
11  <script src="https://cdn.bootcdn.net/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
12  <script src="/js/fullpage.js"></script>
13  <title>演示fullpage</title>
14 </head>
15
16 <body>
17
18   <ul id="menu">
19     <li data-menuanchor="page1">
20       <a href="#page1">第一</a>
21     </li>
22     <li data-menuanchor="page2">
23       <a href="#page2">第二</a>
24     </li>
25     <li data-menuanchor="page3">
26       <a href="#page3">第三</a>
27     </li>
28     <li data-menuanchor="page4">
29       <a href="#page4">第四</a>
30     </li>
31     <li data-menuanchor="page5">
32       <a href="#page5">第五</a>
33     </li>
34   </ul>
35
36
37   <div id="fullpage">
38     <div class="section">第一屏</div>
39     <div class="section">
40       <div class="show animated">
41         演示animate动画
42       </div>
43     </div>
```

```
44     <div class="section">
45         <div class="slide">第三屏的第一屏</div>
46         <div class="slide">第三屏的第二屏</div>
47         <div class="slide">第三屏的第三屏</div>
48         <div class="slide">第三屏的第四屏</div>
49     </div>
50     <div class="section">
51         <div class="con">
52             
53         </div>
54     </div>
55     <div class="section">第五屏</div>
56 </div>
57
58 </body>
59
60 <script>
61     $(function() {
62         $('#fullpage').fullpage({
63             menu: '#menu',
64             sectionsColor: ['#f00', '#4BBFC3', '#7BAABE', '#0f0', '#00f'],
65             scrollingSpeed: 1000,
66             loopHorizontal: false,
67             keyboardScrolling: false,
68             continuousVertical: true,
69             navigation: true,
70             navigationPosition: 'left',
71             controlArrowColor: '#f00',
72             anchors: ["page1", "page2", "page3", 'page4', 'page5'], //为每页添加锚点
73             afterLoad: function(anchor, MyIndex) {
74                 var index = ++MyIndex.index;
75
76                 if (index == 2) {
77                     $(".show").addClass("hinge")
78                 }
79
80                 if (index == 4) {
81                     $(".img").addClass("active")
82                 }
83             },
84             onLeave: function(Index, nextIndex, direction) {
85
86                 var index = ++Index.index;
87
88                 if (index == 2) {
89                     $(".show").removeClass("hinge")
90                 }
91
92                 if (index == 4) {
93                     $(".img").removeClass("active")
94                 }
95             }
96         });
97     });
98 </script>
99
100 </html>
```

CSS:

```
1  * {
2    padding: 0;
3    margin: 0;
4  }
5
6  li {
7    list-style-type: none;
8  }
9
10 a {
11   text-decoration: none;
12 }
13
14 .section {
15   text-align: center;
16   font-size: 40px;
17   color: black;
18   font-weight: bold;
19 }
20
21 #menu {
22   position: fixed;
23   right: 10%;
24   width: 100px;
25   z-index: 1000;
26   top: 50%;
27   transform: translateY(-50%);
28   text-align: center;
29 }
30
31 #menu li {
32   margin-bottom: 20px;
33   font-size: 20px;
34 }
35
36 #menu li a {
37   display: block;
38   color: #fff;
39 }
40
41 #menu .active a {
42   background-color: red;
43 }
44
45 .con {
46   width: 662px;
47   height: 369px;
48   margin: 0 auto;
49   overflow: hidden;
50 }
51
52 .img {
53   display: block;
54   margin-top: -369px;
55   transition: all 1s linear;
56 }
57
58 .img.active {
```

```

59     transform: translateY(369px)
60 }
61
62 .show {
63     width: 50%;
64     margin: 0 auto;
65     color: #fff;
66     font-size: 40px;
67     font-weight: bold;
68     text-align: center;
69     background-color: red;
70 }
71
72
73 /* #fullpage .section:nth-child(1) {
74     background-color: red;
75 }
76
77 #fullpage .section:nth-child(2) {
78     background-color: green;
79 }
80
81 #fullpage .section:nth-child(3) {
82     background-color: pink;
83 }
84
85 #fullpage .section:nth-child(4) {
86     background-color: yellow;
87 }
88
89 #fullpage .section:nth-child(5) {
90     background-color: aqua;
91 } */

```

js:

```

1  $(function() {
2      $('#fullpage').fullpage({
3          menu: '#menu',
4          sectionsColor: ['#f00', '#4BBFC3', '#7BAABE', '#0f0', '#00f'],
5          scrollingSpeed: 1000,
6          loopHorizontal: false,
7          keyboardScrolling: false,
8          continuousVertical: true,
9          navigation: true,
10         navigationPosition: 'left',
11         controlArrowColor: '#f00',
12         anchors: ["page1", "page2", "page3", 'page4', 'page5'], //为每页添加锚点
13         afterLoad: function(anchor, MyIndex) {
14             var index = ++MyIndex.index;
15
16             if (index == 2) {
17                 $(".show").addClass("hinge")
18             }
19
20             if (index == 4) {
21                 $(".img").addClass("active")
22             }
23         },
24         onLeave: function(Index, nextIndex, direction) {

```

```
25
26     var index = ++Index.index;
27
28     if (index == 2) {
29         $(".show").removeClass("hinge")
30     }
31
32     if (index == 4) {
33         $(".img").removeClass("active")
34     }
35 }
36 });
37 });
```