

使用的插件API文档

轮播 Swiper: <https://www.swiper.com.cn/api/index.html>

滚动交互 ScrollMagic: <http://scrollmagic.io/docs/index.html>

补间动画 TweenMax: <https://www.TweenMax.com.cn/api/tweenmax/>

页面用到的插件

```
1 <script src="./js/jquery-3.4.1.min.js" type="text/javascript" charset="utf-8"></script>
2 <script src="./js/swiper.min.js" type="text/javascript" charset="utf-8">
  </script>
3
4 <script src="./js/ScrollMagic.min.js" type="text/javascript" charset="utf-8"></script>
5 <script src="./js/TweenMax.min.js" type="text/javascript" charset="utf-8"></script>
6 <!-- ScrollMagic本身没有动画 TweenMax 添加动画效果-->
7 <script src="./js/animation.gsap.min.js" type="text/javascript" charset="8"></script>
8 <!-- ScrollMagic 页面标识 -->
9 <script src="./js/debug.addIndicators.min.js" type="text/javascript" charset="utf-8"></script>
```

页面经常用到的

类名和图片名: 传播名-该dom层图片实际意义

```
1 <!-- div背景 css中 background-image: url('p30-bgimage-accessorize.jpg') -->
2 <div class='p30-container'>
3   <img class='p30-pic-accessorize.jpg' alt='HUAWEI 图片实际意义'>
4 </div>
```

<picture>书写规范

```
1 <picture>
2   <source type="image/webp" media="(max-aspect-ratio: 1/1)"
3     srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx_xs.webp">
4   <source type="image/webp" media="(min-aspect-ratio: 1/1) and (min-width: 1024px)"
```

```

5      srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx.webp 1x , /content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx@2x.webp 1.5x">
6
7      <source type="image/png" media="(max-aspect-ratio: 1/1)"
8          srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx_xs.png">
9      <source type="image/png" media="(min-aspect-ratio: 1/1) and (min-width: 1024px)"
10         srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx.png 1x , /content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx@2x.png 1.5x">
11
12      <source type="image/jpeg" media="(max-aspect-ratio: 1/1)"
13         srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx_xs.jpg">
14      <source type="image/jpeg" media="(min-aspect-ratio: 1/1) and (min-width: 1024px)"
15         srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx.jpg 1x , /content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx@2x.jpg 1.5x">
16
17      
18  </picture>

```

- 1、图片宽度使用相对单位 如100% 防止图片溢出；
- 2、img元素的srcset属性用于浏览器根据宽、高和像素密度来加载相应的图片；
- 3、srcset表示当屏幕宽度分别为移动设备、2k屏等设备时 src指向不同的图片地址

src表示默认渲染的图片地址 没有选项匹配时 则会渲染默认图片地址

注意顺序： 浏览器匹配第一个之后 不会再看其他项(按需加载，其他的不加载)；

Video标签

```

1  <video id="video"
2     style="object-fit:fill"
3     autoplay
4     poster=""

```

```

5 preload="none"
6 muted
7 loop="loop"
8 webkit-playsinline
9 playsinline
10 x5-video-player-type="h5"
11 x5-video-player-fullscreen="true"
12 x-webkit-airplay="true"
13 >
14 <source src="video/battery.mp4">
15 </video>
16
17 <!--
18 object-fit: fill 视频内容充满整个video容器
19 poster:"img.jpg" 视频封面
20 autoplay: 自动播放
21 auto - 当页面加载后载入整个视频
22 meta - 当页面加载后只载入元数据
23 none - 当页面加载后不载入视频
24
25 muted: 音频静音播放
26
27 webkit-playsinline="true"// 让视频在小窗内播放，也就是不是全屏播放，
28 兼容iOS、Safari和安卓的一些浏览器
29 playsinline="true" // IOS微信浏览器支持小窗内播放*/ 内联播放
30 // app要支持这种模式 webview.allowsInlineMediaPlayback = YES;
31 X-webkit-airplay: // 使此视频支持ios的AirPlay功能
32 x5-video-player-type="h5" : 启用x5内核H5播放器 兼容安卓微信内置浏览器
33 x5-video-player-fullscreen="true" 全屏设置 ture和false的设置会导致布局上的
34 不一样
35 x5-video-orientation="portrait" : 声明播放器支持的方向 可选值landscape 横
36 屏 portrait竖屏
37 默认值portrait 无论是直播还是全屏H5一般都是竖屏播放
38 但是这个属性需要x5-video-player-type开启H5模式
39 -->
40

```

使用懒加载（是否需要使用懒加载，根据具体页面复杂度决定）

```

1 <picture>
2     <source type="image/webp" media="(max-aspect-ratio: 1/1)"

```

```

3      data-srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xx/img/xxx_xs.webp">
4      <source type="image/webp" media="(min-aspect-ratio: 1/1) and (min-width: 1024px)"
5      data-srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xx/img/xxx.webp 1x , /content/dam/huawei-cbg-site/common/mkt/pdp/phones/xx/img/xxx@2x.webp 1.5x">
6
7      <source type="image/png" media="(max-aspect-ratio: 1/1)"
8      data-srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xx/img/xxx_xs.png">
9      <source type="image/png" media="(min-aspect-ratio: 1/1) and (min-width: 1024px)"
10     data-srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx.png 1x , /content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx@2x.png 1.5x">
11
12     <source type="image/jpeg" media="(max-aspect-ratio: 1/1)"
13     data-srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx_xs.jpg">
14     <source type="image/jpeg" media="(min-aspect-ratio: 1/1) and (min-width: 1024px)"
15     data-srcset="/content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx.jpg 1x , /content/dam/huawei-cbg-site/common/mkt/pdp/phones/xxx/img/xxx@2x.jpg 1.5x">
16
17     
18 </picture>

```

@media媒体查询

```

1 @media (max-width: 640px), (max-aspect-ratio: 1/1) and (max-width: 1200px) {
2     // mob设备够用
3 }
4 /* ipad竖屏 */
5 @media only screen and (min-device-width : 768px) and (max-device-width : 1024px)
6 and (orientation : portrait){
7
8 }

```

```

9  /* ipad横屏 */
10 @media only screen and (min-device-width : 768px) and (max-device-width
   : 1024px)
11 and (orientation : landscape){
12  /* style */
13  }
14 @media screen and (max-aspect-ratio: 1/1){
15  // 屏幕宽高比最大为1时 竖屏
16  // 想让图片占满一屏时 竖屏时 h100vh
17  }
18 @media screen and (min-aspect-ratio: 1/1){
19  // 屏幕宽高比最大为1时 横屏 w100vw
20  }

```

手机适配

获取页面公共头部的高度

```

1  var navT = $(".v4.header").length && $("#second-navigation-v4").length ?
   $('#second-navigation-v4').innerHeight() : (isMob ? $('.app-nav').innerHeig
   ht() : $('#header-v3').innerHeight());

```

```

var navT = $(".v4.header").length && $("#second-navigation-v4").length ? $('#second-
navigation-v4').innerHeight() : (isMob ? $('.app-nav').innerHeight() : $('#header-
v3').innerHeight());

```

判断当前设备

```

1  var viewportWidth = window.innerWidth, viewportHeight = window.innerHeigh
   t, aspectRatio = viewportWidth / viewportHeight;
2  var userAgentInfo = navigator.userAgent.toLowerCase();
3  var isIE11 = !!window.ActiveXObject ? !!window.ActiveXObject : "ActiveXOb
   ject" in window,
4  isEdge = userAgentInfo.toUpperCase().indexOf("EDGE") > -1,
5  isIE = userAgentInfo.toUpperCase().indexOf("MSIE") > -1,
6  isHuaweiDevice = navigator.userAgent.toUpperCase().indexOf("HUAWEI") >
   -1,
7  isMob = viewportWidth <= 1200 && aspectRatio < 1 || viewportWidth <=
   600,
8  isIPadPortrait = viewportWidth > 600 && viewportWidth <= 1200 && aspectR
   atio < 1;
9  isRetina = window.devicePixelRatio > 1.5, isIPad = userAgentInfo.match(/i
   pad|macintosh/i),

```

```

10  isAndroid = userAgentInfo.match(/android/i),
11  isFireFox = userAgentInfo.match(/(?:Firefox)/i),
12  isTablet = userAgentInfo.match(/(?:iPad|PlayBook)/i) || (isAndroid &&
!userAgentInfo.match(/(?:Mobile)/i)) || (isFireFox && userAgentInfo.match(/
(?:Tablet)/i));

```

scrollMagic：根据屏幕滚动到页面位置设置动画-触发动画

对每个滚动容器使用一个控制器，并附加多个场景，以定义在页面的哪个部分应该发生什么；

```

1  // 初始化控制器 Controller用于总体的调度
2  var controller = new ScrollMagic.Controller({
3    // 这其中的选项将会被传递到每一个场景
4    globalSceneOptions: {
5      reverse: true // 反向滚动时执行动画
6    }
7  });
8  // 创建新的时间轴
9  var s3_animate2 = new TimelineLite();
10 // 在时间轴上添加动画效果
11 s3_animate3.to('做动画的元素', 1, { opacity: 1, ease: Power2.easeInOut
});
12 // Scene场景类 用于设计具体的变换
13 new ScrollMagic.Scene({
14   triggerElement: '#scroll', // 触发元素(一般放在要做动画元素的外部)
15   // 触发元素相对于窗口的位置(0-1之间的数字) 到此位置时才会触发
16   // 1=>'onEnter' 0.5=>'onCenter' 0=>'onLeave'
17   triggerHook: 0.5,
18   duration: 300, // 效果持续距离 也可以使用百分比(一般在100%-300% 即1到3倍的
    屏幕高度)
19   offset: 0 // 滚动到什么位置触发效果
20 });
21 .setPin('#stick', { pushFollowers: true }) // 要固定的元素 pushFollowers
    将滚动距离添加至动画元素高
22 // 添加动画效果 需要插件animation.gsap.min.js
23 // 引入此插件后 动画时间不再生效 只控制进度
24 // 如第一个动画时间1秒 动画总时长2秒 duration为100% 则第一个动画占duration的
    50%
25 .setTween(s3_animate3)

```

```

26 .setClassToggle('选择器', '要切换的类名') // 切换选中元素类名
27 .addIndicators() // 添加页面的标识信息
28 .addTo(controller); // 在控制器中注册此场景
29 // 注意：为了兼容不同分辨率的设备 持续距离、触发位置可以用下例表示 如
30 // window.innerHeight * 3 / 4 表示在屏幕3 / 4位置处触发效果 或效果持续3 / 4
  屏幕
31 // 做动画的元素宽度不要超过屏幕宽度 否则可能造成计算错误

```

scrollMagic本身没有动画 要配合tweenMax一起使用

```

1 var s5_animate = new TimelineLite();
2 // 在创建的时间轴上定义动画 s5-bg-pic向左偏移50% 放大至屏幕大小 取消圆角边框
3 s5_animate.to('.s5-bg-pic', 1, { x: '-50%', ease: Power2.easeInOut });
4 .to('.s5-bg-pic', 1, {
5   width: '100vw',
6   height: '100vh',
7   'border-radius': 0,
8   ease: Power2.easeInOut
9 });
10 // 定义ScrollMagic对象 加入s5_animate动画 s5-trigger离开时触发 触发期间固定section5元素
11 new ScrollMagic.Scene({ triggerElement: '#s5-trigger', duration: '100%',
  triggerHook: 0 })
12 .setTween(s5_animate)
13 .setPin('.section5')
14 .addTo(controller);

```

scrollMagic事件

```

1 var scene = new ScrollMagic.Scene({});
2 scene.on("start", function (event) { // 定义滚动到定义距离时的回调函数
3   // 可以使用setTimeout来实现延时触发效果
4   // start进入 end离开 progress进行中 enter leave
5   console.log("进入场景");
6   if (event.scrollDirection == 'FORWARD') {
7     console.log("向下滚动");
8   }
9   if (event.scrollDirection == 'REVERSE') {
10    console.log("向上滚动");

```

```
11 }  
12 });
```

tweenMax

tweenMax是一个动画库

经常用到的一些动画结构属性、事件

1、TweenMax()

```
1 // TweenMax的构造函数 创建TweenMax对象  
2 // 其中包含三个参数 分别是 动画对象 执行时间 动画效果  
3 new TweenMax('.box', 3, {  
4   x: 500,  
5   y() { // 也可以写成回调函数  
6     return Math.random() * 300  
7   },  
8   alpha : 0.3,  
9 });
```

2、TweenMax.to(): 需给出起始样式, js中给出动画终点样式

```
1 // 对多个目标进行动画 第一个参数为动画对象 也可以写成.box  
2 // 需要反向动画, 需要在css中设置初始值, 不然回不去, 曾经犯过的错误 fromTo就不用了  
3 TweenMax.to(['.box1', '.box2', '.box3'], 1, {  
4   x(index) {  
5     return (index + 1) * 100 // 100 200 300  
6   },  
7   alpha: 0.3,  
8 });
```

eg:

```
1 var s_animate = new TimelineMax();  
2 s_animate.add([  
3   TweenMax.to($('#s2-img'), 0.7, {scale: 0.19, top: s2ImgT + 'px'}),  
4   TweenMax.to($('#s2-title'), 0.7, {scale: 0.7, top: '-50%'})
```


3、TweenMax.from(): 需给出终止样式, js中给出动画起始样式;

4、TweenMax.fromTo(): 在js中给出动画的起始和终点两个状态样式;

```
1 // 第三、第四个参数分别可以设置动画的起点和终点
2 // 比如元素从opacity: 0, y: 50的位置移动到opacity: 1, y: 0 上浮的效果
3 TweenMax.fromTo($('.s-img-item'), 0.5,
4 {
5   opacity: 0,
6   y: 50,
7   ease: Power2.easeInOut
8 }, {
9   opacity: 1,
10  y: 0,
11  ease: Power2.easeInOut // 过渡效果的速度曲线
12 });
```

5、onComplete : 动画结束时触发的回调函数;

```
1 TweenMax.to('.box', 2, {
2   x: 500,
3   onComplete: function() {
4     ...
5   }
6 });
```

6、.play() (动画播放)

```
1 // 1.动画开始播放时间 不传此参数代表从当前位置开始播放
2 // 2.如果true(默认值) 当播放头移动到from参数中定义的新位置 不会触发任何事件或回调
3 myAnimation.play(2, false); // false时不阻止期间的函数或事件
```

7、.pause() (动画暂停)

```
1 // 跳转到第2秒并暂停动画 第二个参数为false时 不会抑制事件和函数触发
```

```
2 myAnimation.pause(2, false); // 不传参数只暂停
```

8、.reverse()

```
1 // 当前位置反向播放
2 myAnimation.reverse();
3
4 // 2秒位置反向播放
5 myAnimation.reverse(2);
6
7 // 2秒位置反向播放 不抑制事件触发
8 myAnimation.reverse(2, false);
9
10 // 动画末端反向播放
11 myAnimation.reverse(0);
12
13 // 动画末端往回1秒位置反向播放
14 myAnimation.reverse(-1);
15
16 // 切换方向(如前进方向 则反向播放 如反向 则向前播放)
17 if (myAnimation.reversed()) {
18   myAnimation.play();
19 } else {
20   myAnimation.reverse();
21 }
```

eg:

```
1 var s16_animate = new TimelineMax();
2 s16_animate.add([
3   TweenMax.fromTo($('.stia-s16-img-1'), 1.5, {left: $('.stia-s16-img-1').innerWidth(), top: $('.stia-s16-img-1').height() / 2, ease: Power2.easeInOut}, {left: _toLeft, top: 0, ease: Power2.easeInOut}),
4   TweenMax.fromTo($('.stia-s16-img-2'), 1.6, {right: -$('.stia-s16-img-1').innerWidth(), top: $('.stia-s16-img-1').height() / 2 + Number($('.stia-s16-img-2').css('top').split('px')[0]), ease: Power2.easeInOut}, {top: _toTop, right: 0, ease: Power2.easeInOut})
5   ]);
6
7 s16_animate.pause(); // 暂停动画，避免页面加载完成就加载了动画，到位置时已经完事儿了
```

```

8
9  new ScrollMagic.Scene({triggerElement: "#s16-trigger", duration: 0, triggerHook: 0.4, offset: 0})
10  .on('enter', function () {
11    s16_animate.play(); // 进入到定义的位置时开始播放动画
12  })
13  .on('leave', function () {
14    s16_animate.reverse();
15  })
16  .addTo(stia.controller);

```

其他看文档吧！

锁屏！

两种方式：

- 1、scrollMagic 的 setPin();
- 2、css样式 position:sticky (同时具备position的relative和fixed定位的特性；)

position:sticky;

<http://www.ruanyifeng.com/blog/2019/11/css-position.html>

sticky生效的前提是，必须搭配top、bottom、left、right其中一个，否则等同于relative定位，不产生"动态固定"的效果。

它的具体规则是，当页面滚动，父元素开始脱离视口时（即部分不可见），只要与sticky元素的距离达到生效门槛，relative定位自动切换为fixed定位；等到父元素完全脱离视口时（即完全不可见），fixed定位自动切换回relative定位。

请看下面的示例代码。（注意，除了已被淘汰的 IE 以外，其他浏览器目前都支持sticky。但是，Safari 浏览器需要加上浏览器前缀-webkit-。）

```

1  #toolbar {
2
3    position: -webkit-sticky; /* safari 浏览器 */
4
5    position: sticky; /* 其他浏览器 */
6

```

```
7 top: 20px;  
8  
9 }
```

- ①必须有top、right、left、bottom其中一个
- ②父容器的高度决定内容锁屏的时长
- ③sticky设置的div以上的所有父级不能有overflow: hidden
IE不兼容不能达到这个效果 Safari 必须加-webkit-前缀

页面制作相关规范：

①、物料图片文件夹规范

除序列帧之外其他物料尽量保证每个文件夹内不超过100个。可以每屏一个文件夹（适合屏数少的产品）或者讲述同一个功能的几屏图片放在一个文件夹

②、附件已经打包好了一份通用js代码（utilsPlugin.js），里面包含（scrollMagic.js、GreenSock.js - 包含常用的tweenMax/TimelineMax等、lazysizes.js），

正常情况下制作新的产品页面，只需引入 **utilsPlugin.js** + **xx-main.js** 这两个js文件即可。

③、以图片清晰为目的，保证图片清晰的情况下，图片体积可以适当加大