

H5C3

《购物车宣传》案例

综合运用h5c3知识且进行一定扩展
能使用jquery完成网页常见交互行为
实际工作开发当中能应对活动宣传页

FullPage插件

- 插件功能介绍
 - 基于 jQuery 的插件，它能够帮你很方便、很轻松的制作出全屏网站。
 - 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。
 - 参考文档：<http://www.dowebok.com/demo/2014/77/>
 - 原理：window.onmousewheel = function(){ console.log('ok') };
- 使用步骤
 - 引用文件

```
<link rel="stylesheet" href="css/jquery.fullPage.css">
<script src="js/jquery.min.js"></script>
<script src="js/jquery.fullPage.js"></script>
```

- html结构

```
<div id="fullpage">
  <div class="section">第一屏</div>
  <div class="section">第二屏</div>
  <div class="section">
    <div class="slide">第三屏的第一屏</div>
    <div class="slide">第三屏的第二屏</div>
    <div class="slide">第三屏的第三屏</div>
    <div class="slide">第三屏的第四屏</div>
  </div>
  <div class="section">第四屏</div>
</div>
```

- js初始化

```
$(function(){
  $('#fullpage').fullpage();
});
```

- 配置

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
slidesColor	函数	无	设置背景颜色
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
navigationPosition	字符串	right	项目导航的位置，可选 left 或 right
navigationColor	字符串	#000	项目导航的颜色
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

选项	类型	默认值	说明
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
scrollOverflow	布尔值	false	内容超过满屏后是否显示滚动条
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

选项	类型	默认值	说明
animateAnchor	布尔值	true	
normalScrollElementTouchThreshold	整数	5	-

- 方法

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

- 回调函数

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

搭建页面开发结构

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```

<title>购物网站宣传页面</title>
<link rel="stylesheet" href="css/jquery.fullpage.min.css">
<link rel="stylesheet" href="css/index.css">
</head>
<body>
<!--更多精彩-->
<div class="more"></div>
<!--全屏滚动-->
<div class="container">
  <section class="section"></section>
  <section class="section"></section>
  <section class="section"></section>
  <section class="section"></section>
  <section class="section"></section>
  <section class="section"></section>
  <section class="section"></section>
  <section class="section"></section>
</div>
<script src="js/jquery.min.js"></script>
<script src="js/jquery.fullpage.min.js"></script>
<script>
  $(function() {
    $(''.container').fullpage();
  })
</script>

```

完成八屏页面动画

- 模块一: 标题(title)
- 模块二: 商品列表(list)
- 模块三: 选购(buy)
- 模块四: 填写信息(info)
- 模块五: 付款(payment)
- 模块六: 送货(delivery)
- 模块七: 评价(appraise)
- 模块八: 购物(shopping)

具体需求参照示例

《QQ TIM》案例

Stellar插件

- 描述
 - 视差滚动 (Parallax Scrolling) 指网页滚动过程中, 多层次的元素进行不同程度的移动, 视觉上形成立体运动效果的网页展示技术。主要核心就是前景和背景以不同的速度移动, 从而创造出3D效果。这种效果可以给网站一个很好的补充。
- 特性
 - 视差滚动效果酷炫, 适合于个性展示的场所。
 - 视差滚动徐徐展开, 适合于娓娓道来, 讲故事的场合。
 - 视差滚动容易迷航, 需要具备较强的导航功能。
- 原理
 - 传统的网页的文字、图片、背景都是一起按照相同方向相同速度滚动的, 而视差滚动则是在滚动的时候, 内容和多层次的背景实现或不同速度, 或不同方向的运动。

有的时候也可以加上一些透明度、大小的动画来优化显示。利用background-attachment属性实现。

- 使用步骤
 - 引用文件

```
<script src="jquery/jquery.min.js"></script>
<script src="jquery.stellar.min.js"></script>
```

- html结构

```
<div class="content" id="content1">
  <p>TEXT HERE</p>
</div>
<div class="content" id="content2">
  <p>TEXT HERE</p>
</div>
<div class="content" id="content3" data-stellar-background-ratio="0.5">
  <p>TEXT HERE</p>
</div>
<div class="content" id="content4" data-stellar-background-ratio="0.5">
  <p>TEXT HERE</p>
</div>
<div class="content" id="content5" data-stellar-background-ratio="0.5">
  <p>TEXT HERE</p>
</div>
<div class="content" id="content6" data-stellar-background-ratio="0.5">
  <p>TEXT HERE</p>
</div>
```

- 基本样式

```
.content {
  background-attachment: fixed;
  height: 400px;
}
#content1 {
  background-image: url("../");
}
#content2 {
  background-image: url("../");
}
#content3 {
  background-image: url("../");
}
#content4 {
  background-image: url("../");
}
#content5 {
  background-image: url("../");
}
#content6 {
  background-image: url("../");
}
```

- js初始化

```
$.stellar({
    horizontalScrolling: false,
    responsive: true
});
```

- 参数解释

名称	说明
horizontalScrolling 和 verticalScrolling	该配置项用来设置视差效果的方向。horizontalScrolling设置水平方向，verticalScro设置垂直方向，为布尔值，默认为true
responsive	该配置项用来制定load或者resize时间触发时是否刷新页面，其值为布尔值，默认为false
hideDistantElements	该配置项用来设置移出视线的元素是否隐藏，其值为布尔值，若不想隐藏则设置为false
data-stellar-ratio="2"	定义了此元素针对页面滚动的速度比率，比如，0.5为页面滚动的50%，2为页面滚动的200%，所以数值越大，你可以看到页面元素滚动速度越快。
data-stellar-background-ratio	该配置项用在单个元素中，其值为一个正数，用来改变被设置元素的影响速度。例如 值为0.3时，则表示背景的滚动速度为正常滚动速度的0.3倍。如果值为小数时最好在样式表中设置

完成视觉滚动差页面开发

- 模块一: 导航(nav)
- 模块二: 宣传(banner)
- 模块三: 语音聊天(sound)
- 模块四: 文件传输(file)
- 模块五: 兴趣部落(interest)
- 模块六: 底部/footer)

具体需求参照示例

H5-dom扩展

获取元素

```
document.getElementsByClassName ('class');
//通过类名获取元素，以伪数组形式存在。

document.querySelector('selector');
//通过CSS选择器获取元素，符合匹配条件的第1个元素。

document.querySelectorAll('selector');
//通过CSS选择器获取元素，以伪数组形式存在。
```

类名操作

```
Node.classList.add('class');
//添加class

Node.classList.remove('class');
//移除class

Node.classList.toggle('class');
//切换class，有则移除，无则添加

Node.classList.contains('class');
//检测是否存在class
```

自定义属性

在HTML5中我们可以自定义属性，其格式如下data-*=""

```
<div id="demo" data-my-name="itcast" data-age="10">
<script>
/*
    Node.dataset是以对象形式存在的，当我们为同一个DOM节点指定了多个自定义属性时，
    Node.dataset则存储了所有的自定义属性的值。
*/
var demo = document.querySelector(反馈);
//获取
//注：当我们如下格式设置时，则需要以驼峰格式才能正确获取
var name = demo.dataset['myName'];
var age = demo.dataset['age'];
//设置
demo.dataset['name'] = 'web developer';
</script>
```

H5-新增API

全屏方法

HTML5规范允许用户自定义网页上任一元素全屏显示。

- Node.requestFullScreen() 开启全屏显示
- Node.cancelFullScreen() 关闭全屏显示
- 由于其兼容性原因，不同浏览器需要添加前缀如：
webkit内核浏览器：webkitRequestFullScreen、webkitCancelFullScreen，如chrome浏览器。
Gecko内核浏览器：mozRequestFullScreen、mozCancelFullScreen，如火狐浏览器。
- document.fullScreen检测当前是否处于全屏
不同浏览器需要添加前缀
document.webkitIsFullScreen、document.mozFullScreen

多媒体

自定义播放器

方法

方法	描述
addTextTrack()	向音频/视频添加新的文本轨道
canPlayType()	检测浏览器是否能播放指定的音频/视频类型
load()	重新加载音频/视频元素
play()	开始播放音频/视频
pause()	暂停当前播放的音频/视频

属性

属性	描述
audioTracks	返回表示可用音轨的 AudioTrackList 对象
autoplay	设置或返回是否在加载完成后随即播放音频/视频
buffered	返回表示音频/视频已缓冲部分的 TimeRanges 对象
controller	返回表示音频/视频当前媒体控制器的 MediaController 对象
controls	设置或返回音频/视频是否显示控件（比如播放/暂停等）
crossOrigin	设置或返回音频/视频的 CORS 设置
currentSrc	返回当前音频/视频的 URL
currentTime	设置或返回音频/视频中的当前播放位置（以秒计）
defaultMuted	设置或返回音频/视频默认是否静音
defaultPlaybackRate	设置或返回音频/视频的默认播放速度
duration	返回当前音频/视频的长度（以秒计）
ended	返回音频/视频的播放是否已结束
error	返回表示音频/视频错误状态的 MediaError 对象
loop	设置或返回音频/视频是否应在结束时重新播放
mediaGroup	设置或返回音频/视频所属的组合（用于连接多个音频/视频元素）
muted	设置或返回音频/视频是否静音
networkState	返回音频/视频的当前网络状态
paused	设置或返回音频/视频是否暂停
playbackRate	设置或返回音频/视频播放的速度
played	返回表示音频/视频已播放部分的 TimeRanges 对象
preload	设置或返回音频/视频是否应该在页面加载后进行加载
readyState	返回音频/视频当前的就绪状态
seekable	返回表示音频/视频可寻址部分的 TimeRanges 对象
seeking	返回用户是否正在音频/视频中进行查找
src	设置或返回音频/视频元素的当前来源
startDate	返回表示当前时间偏移的 Date 对象
textTracks	返回表示可用文本轨道的 TextTrackList 对象
videoTracks	返回表示可用视频轨道的 VideoTrackList 对象
volume	设置或返回音频/视频的音量

事件

事件	描述
abort	当音频/视频的加载已放弃时
canplay	当浏览器可以播放音频/视频时
canplaythrough	当浏览器可在不因缓冲而停顿的情况下进行播放时
durationchange	当音频/视频的时长已更改时
emptied	当目前的播放列表为空时
ended	当目前的播放列表已结束时
error	当在音频/视频加载期间发生错误时
loadeddata	当浏览器已加载音频/视频的当前帧时
loadedmetadata	当浏览器已加载音频/视频的元数据时
loadstart	当浏览器开始查找音频/视频时
pause	当音频/视频已暂停时
play	当音频/视频已开始或不再暂停时
playing	当音频/视频在已因缓冲而暂停或停止后已就绪时
progress	当浏览器正在下载音频/视频时
ratechange	当音频/视频的播放速度已更改时
seeked	当用户已移动/跳跃到音频/视频中的新位置时
seeking	当用户开始移动/跳跃到音频/视频中的新位置时
stalled	当浏览器尝试获取媒体数据，但数据不可用时
suspend	当浏览器刻意不获取媒体数据时
timeupdate	当目前的播放位置已更改时
volumechange	当音量已更改时
waiting	当视频由于需要缓冲下一帧而停止

地理定位

在HTML规范中，增加了获取用户地理信息的API，这样使得我们可以基于用户位置开发互联网应用，即基于位置服务 (Location Base Service)

- 获取当前地理信息

```
navigator.geolocation.getCurrentPosition(successCallback, errorCallback)
```

- 重复获取当前地理信息

```
navigator.geolocation.watchPosition(successCallback, errorCallback)
```

- 当成功获取地理信息后，会调用succssCallback，并返回一个包含位置信息的对象position。
position.coords.latitude纬度
position.coords.longitude经度
position.coords.accuracy精度
position.coords.altitude海拔高度
- 当获取地理信息失败后，会调用errorCallback，并返回错误信息error
- 在现实开发中，通过调用第三方API（如百度地图）来实现地理定位信息，这些API都是基于用户当前位置的，并将用位置位置（经/纬度）当做参数传递，就可以实现相应的功能。

本地存储

随着互联网的快速发展，基于网页的应用越来越普遍，同时也变的越来越复杂，为了满足各种各样的需求，会经常性在本地存储大量的数据，HTML5规范提出了相关解决方案。

- 特性
 - 设置、读取方便
 - 容量较大，sessionStorage约5M、localStorage约20M
 - 只能存储字符串，可以将对象JSON.stringify() 编码后存储
- window.sessionStorage
 - 生命周期为关闭浏览器窗口
 - 在同一个窗口(页面)下数据可以共享
- window.localStorage
 - 永久生效，除非手动删除（服务器方式访问然后清除缓存）
 - 可以多窗口（页面）共享
- 方法
 - setItem(key, value) 设置存储内容
 - getItem(key) 读取存储内容
 - removeItem(key) 删除键值为key的存储内容
 - clear() 清空所有存储内容

历史管理

提供window.history，对象我们可以管理历史记录，可用于单页面应用，Single Page Application，可以无刷新改变网页内容。

- pushState(data, title, url) 追加一条历史记录
 - data用于存储自定义数据，通常设为null
 - + title网页标题，基本上没有被支持，一般设为空
 - + url 以当前域为基础增加一条历史记录，不可跨域设置
- replaceState(data, title, url) 与pushState()基本相同，不同之处在于replaceState()，只是替换当前url，不会增加/减少历史记录。
- onpopstate事件，当前进或后退时则触发

离线应用

HTML5中我们可以轻松的构建一个离线（无网络状态）应用，只需要创建一个cache manifest文件。

- 优势
 - 1、可配置需要缓存的资源
 - 2、网络无连接应用仍可用
 - 3、本地读取缓存资源，提升访问速度，增强用户体验

- 4、减少请求，缓解服务器负担
- 缓存清单
 - 一个普通文本文件，其中列出了浏览器应缓存以供离线访问的资源，推荐使用.appcache为后缀名
 - 例如我们创建了一个名为demo.appcache的文件，然后在需要应用缓存在页面的根元素(html)添加属性manifest="demo.appcache"，路径要保证正确。
- manifest文件格式
 - 1、顶行写CACHE MANIFEST
 - 2、CACHE: 换行 指定我们需要缓存的静态资源，如.css、image、js等
 - 3、NETWORK: 换行 指定需要在线访问的资源，可使用通配符
 - 4、FALLBACK: 换行 当被缓存的文件找不到时的备用资源
- 其它
 - 1、CACHE: 可以省略，这种情况下将需要缓存的资源写在CACHE MANIFEST
 - 2、可以指定多个CACHE: NETWORK: FALLBACK:，无顺序限制
 - 3、#表示注释，只有当demo.appcache文件内容发生改变时或者手动清除缓存后，才会重新缓存。
 - 4、chrome 可以通过chrome://appcache-internals/工具和离线（offline）模式来调试管理应用缓存

文件读取

HTML5新增内建对象，可以读取本地文件内容。

网络状态

- 我们可以通过window.onLine来检测，用户当前的网络状况，返回一个布尔值
 - window.online用户网络连接时被调用
 - window.offline用户网络断开时被调用