

基础实验四

杨妍喆 2012013323

18010045724 yangyanzhe@126.com

运行环境及说明

测试环境为 32 位 Window 7 操作系统， Intel Core i5-3210M CPU，在 Chrome 最新版浏览器中打开。使用 nginx 作为本地服务器，详见 bin 文件夹。配置时双击 nginx.exe 完成配置，打开浏览器，输入 127.0.0.1 即可。

Part I 新闻图片轮播

1. 数据获取

- 实现要点：
- 1.1 通过 ajax，完成 http 的 Get 请求，打开 json 文件
 - 1.2 调用 JSON.parse()完成 json 的解析
 - 1.3 自行设计 json 并填充相应内容
 - 1.4 数据处理，通过 jquery 来修改 DOM 元素的属性值或内容，在前端展示

2. 异步刷新

实现要点：本次实验采用一次读入的方式，在 1.4 中即完成属性值或内容的修改。可以看到在 index.html 中，等标签中的内容为空，可以得出其为异步刷新。

3. 基本事件响应

实现要点：

3.1 单击触发：

通过.click()函数实现鼠标点击的基本响应，鼠标点击的响应分为两部分：单击“左移”和“右移”图标与单击缩略图。

单击“左移”、“右移”图标时只会单向移动；单击缩略图则完成跳转。

3.2 定时器触发：

通过 setInterval(function, intervalTime)函数来完成定时器的设置，并不断调用作为内部的第一个参数的函数

3.3 单击触发与定时器触发的结合

这里主要是针对单击缩略图的触发，因为左右移的触发只要调用单一的移动、调整函数即可，如果是缩略图的触发，需要先停止之前的定时器触发，完成跳转，再开启新的定时器触发。

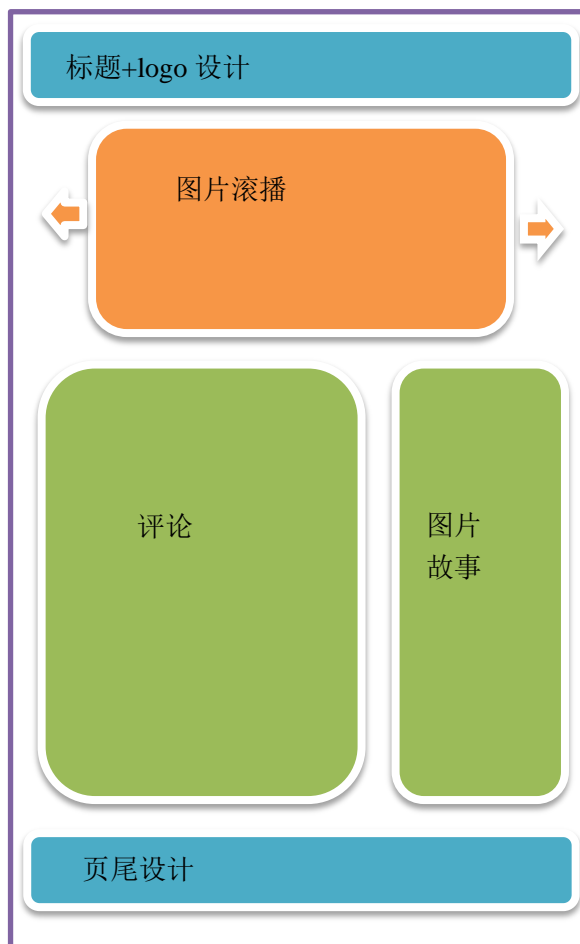
3.4 图片的动画

使用 jquery 的 animate()动画完成左右的迁移效果，图片横成一行排列，只是中间作为可见区，这样移动装着图片的 div，就可以看到类似移动切换的效果。

4. 基本 UI 设计

实现思路：

- 4.1. 遵从简约的 UI 风格，参考 google doodles 等网页，布局图大致如下



4.2. 尽量使用 reflective 布局，增强网页的可适性

4.3 由于左移到最后一个时，通常的设计思路为闪到右边，但是我认为这样的用户体验非常不好，快速的闪动时间长了后会产生晕眩，让人不愿意久看。因此改为左移到头后再右移，而下方的缩略图也大概采取少移动的策略，主要根据右侧的三个缩略图来判断策略。

4.4 采用将标题设置为 opacity 为 0，当鼠标悬停时才会显示标题的方式，更加符合需求，当感兴趣的时候去查看新闻。

4.5 整体风格统一，简约自然

Part II 评论部分

1. 数据获取

1.1 类似轮播部分的数据获取，通过 ajax 来完成服务器请求，打开 json 文件，利用 JSON.parse()来完成解析；

1.2 图片轮播内容与评论内容为一个 json 文件下的不同对象。

2. 翻页功能

2.1 翻页功能完全自己实现，通过对 click 事件的监听，进行异步刷新，实现只有局部改变而非整个页面刷新的效果；

2.2 改变 cursor 即箭头的不同形状，来表示可以点击和不能点击。

2.3 在翻页页码给出不同的颜色和背景，显示出是否为当前页面；

3. 界面展示

3.1 评论元素完善：时间、姓名、头像、评论内容

3.2 遵从 part1 中的 UI 整体设计风格与原则

3.3 在页码处有箭头的变化设计和按钮的背景、文字颜色变化设计，用户友好

PartIII HTML5 要求

思路：

1. 在 document.ready() 中添加初始化的函数，即利用本地存储使用的 localRecord[] 中的信息来完成下一个页面的初始化；

2. 注意 localRecord[] 中储存的是 string 类型的变量，需要进行转换，成为 number 方便使用；

3. 在调用 rotate() (该函数用来完成 slide 的变化) 及 changeReview() (该函数用来完成评论的异步更新)、changeReviewCurrent() (该函数用来完成评论页码的显示调整) 时，同时完成 localRecord[] 的修改；

4. 设定只需保存两个内容：4.1 浏览的页面的下标 'imgID'

4.2 浏览的评论的页码 'pageID'

效果为再次打开页面或刷新页面还会恢复关闭前状态

实现要点：

1. 设置初始值(在这个要求之前，初始值一直为 0，因此要更改为 localRecord 的内容)

2. 完成初始化的函数 setReview()，在数据处理后调用该函数；

3. 添加修改后及时保存到 localRecord 中相应的对象下

PartIV CSS3 要求

思路：

1. 在 \$(document).mouseenter() 和 \$(document).mouseleave() 函数中修改 slide 所在的 div 的样式，且调用 clearInterval(***) 的方式来完成图片轮播的暂停。当鼠标离开后即轮播继续进行；

2. 刚开始的思路是鼠标一离开就调用 rotate() 函数直接开始轮播，但是这样会有一个后果就是如果鼠标恰好在边缘，就会看到图片过于灵敏地切换，用户体验很差。因此设置从鼠标离开后开始 setInterval()，也就是会有一个单位的时间后再开始轮播，这样就不会出现鼠标在 div 边缘，使得图片快速切换闪动的情况；

3. 在暂停图片轮播的同时，修改图片的样式，使得原来 opacity 设置为 0 的标题栏显示出来。满足大家了解新闻内容的需要。

实现要点：

\$(document).mouseenter() 函数与 \$(document).mouseleave() 的实现

Part V 成果展示 及亮点

☆☆☆☆☆

1. slides 中的缩略图

缩略图看上去简单，但是蕴含着较为复杂的逻辑，首先它需要根据 current 来设计不同的透

明度，其次，在到达边缘时，它需要移动来使后面的图片可以看到。为了用户友好，根据被遮挡的几张图片来移动，而不是每换一张图就移动一次。

效果不错。

☆☆☆☆☆

2. 标题栏的事件响应设计

通过设计根据滚轮及页面情况缩放的标题栏，使页面更有活力。CSS 样式参照 google doodle 的设计，但是 logo、标题及事件响应为自己完成

☆☆☆☆☆

3. 分页栏的简约美观



统一的淡蓝色风格，友好的交互，让我觉得分页栏是我的得意之处。

☆☆☆

4. 新闻故事

右侧的 div 当把鼠标移上去时，会出现淡黄色的背景，暂时还没有做分页面，如果有分页面，就可以在已有的[a](#)中加入 href，来完成链接。

Part VI 总结与反思

在这次的实验中，完成了所有的基本任务要求，并且在尝试中加入了自己的创新点和思考，通过学习和思考设计逻辑来完成一个用户友好的网页。

不足之处在于：一方面，其实应该是新闻对应评论的。着实是一个假新闻页面，没有具体的内容页。另一方面，并没有完全做到自适应。因为是中间才加上自适应，之前有一些 css 内容并没有更改完全，还有一些其实还是写死的像素值。希望自己可以写出自适应的页面。此外，last but not least，其实代码还是很长，写到后来已经发现改起来不那么方便了。感觉还是认识不清吧，不知道放在不同的文件中，编译会造成哪些问题，因此就将 js 代码放在一个文件中。希望自己再多尝试一下，与助教和老师交流。

这其实只是前端学习的一个开始，希望今后可以写出更加漂亮的代码，更加漂亮的界面。