

JavaScript 作业实验报告

目录

1.	中文数字时钟.....	2
1.1.	实验题目	2
1.2.	实验截图.....	2
1.3.	运行方法.....	2
1.4.	实验分析.....	2
1.5.	实验心得.....	2
2.	个性化网页.....	2
2.1.	实验题目	2
2.2.	实验截图.....	3
2.3.	运行方法.....	3
2.4.	实验分析.....	4
2.4.1.	状态栏中的实时时间	4
2.4.2.	JS 自动生成文档目录.....	4
2.4.3.	点击回到顶部.....	4
2.4.4.	点击下拉显示预览效果.....	4
2.5.	实验心得.....	4

1. 中文数字时钟

1.1. 实验题目

编写形如下图的中文数字时钟。

日期: 2013年4月25日 星期四
时间: 09时05分08秒

1.2. 实验截图



Figure 1 极客风格的中文数字时钟

1.3. 运行方法

进入 `src/clock` 目录，双击 `clock.html`，即可在浏览器看到运行结果。

注意：推荐使用 Chrome 内核浏览器打开，并且确保浏览器没有阻止 javascript。

1.1. 实验分析

在 js 中，通过创建一个 `Date` 类的对象，就可以获取到对象创建时候的时间。这样思路就很明确了，使用 `setTimeout` 函数，每隔 500ms 新建一个 `Date` 对象，然后将获取到的时间中的具体信息通过对应的 `get` 函数读取出来，然后再 HTML 中的特定位置显示出来即可。

下面有一些需要注意的点：

- `getFullYear` 获取到的值是减去 1900 后的结果，所以要将得到的值加上 1900
- `getDay` 获取到的星期是数字形式，可以通过创建数组，然后把得到的数字做为下标，读取数组即可得到对应的中文
- 当数字小于 10 时，读取到的数字如果直接用 `toString` 转化为字符串，长度只有一位，影响美观。可以通过创建辅助函数来解决。本实验中，创建了一个 `lpad(n, width)` 函数，该函数先将得到的数字转为字符串，然后判断如果长度小于要求的 `width`，就在字符串前面补 0 直到满足给定长度 `width`。

具体详见 `clock.html` 源码底部 `script` 部分

1.4. 实验心得

本实验要求实现一个中文数字时钟，非常有实际意义。不过，如果只是纯粹的实现，就显得有点单调，本人结合所学 CSS 知识，稍微调节了一下 UI，模拟传统的命令行终端配色，最终得到了这个极客风格的中文数字时钟，觉得有趣。

2. 个性化网页

2.1. 实验题目

手工编写个性化网页，尽可能多的利用已经讲过的 HTML 和 JavaScript 功能。（可以与上一章的平时作业 24 合做在一起）

2.2. 实验截图



Figure 2 个性化网页截图 1



Figure 3 个性化网页截图 2

2.3. 运行方法

进入 src/webpage 目录，双击 index.html 文件，即可在浏览器上看到运行效果。网页有如下简单功能：

- 单击预览按钮，可以看到相应的标签的运行结果，再次单击按钮，即可收起预览框
- 点击左部的目录标题，页面就会滚动到对应的位置，同时高亮显示被点击的标题大约 1 秒
- 当页面向下滚动一定距离时，会出现回到顶部按钮，点击就会回到顶部
- 顶部状态栏是固定住的，方便阅读者时刻关注当前时间

注意：推荐使用 Chrome 内核浏览器打开，并且确保浏览器没有阻止 javascript。同时，由于本页面调用了外部的 bootstrap 样式，所以请确保电脑处于联网状态。

2.4. 实验分析

本实验引用了 Bootstrap 样式做一下布局调整，同时使用了 jquery 库(基于 JavaScript 封装)来丰富动态效果。但整体还是尽可能的使用了原生的 JavaScript，下面就实验中的几个功能的实现进行详细分析

2.4.1. 状态栏中的实时时间

调用了中文数字时钟作业中的函数来实现，详见 1.4 部分的分析

2.4.2. JS 自动生成文档目录

在 HTML 中给特定的 div 块加上 intro-block 类，然后遍历所有的 intro-block，将其中的 h2 标签作为 1 级标题，h3 标签作为二级标题，同时给 h2 和 h3 加上自动生成的特定格式的 id，然后将生成的目录链接到生成的 id 所对应的位置即可。用到的原生 JavaScript 的函数主要有 `getElementsByClassName`，同时应用 jquery 库中的函数 `animate` 来实现动态滚动，滚动到特定位置的标题高亮是通过给标题加上 class 来实现的，该 class 样式使用了 CSS3，详见 main.css 中的 flash 类。

2.4.3. 点击回到顶部

回到顶部按钮使用了 Bootstrap 提供的字体。默认将该按钮隐藏，然后判断页面的滚动长度，如果超过 300px，就现实该按钮。主要用到了 jquery 库中的 `scrollTop`, `fadeIn`, `fadeOut` 等函数。

2.4.4. 点击下拉显示预览效果

在 CSS 中，将预览框的 `display` 属性设置为 `none`，同时用 js 动态在该预览框上面添加按钮，点击该按钮时，触发显示或隐藏事件。主要用到了 jquery 库中的 `slideToggle` 函数。

2.5. 实验心得

本页面基于上次作业的简单 HTML 页面进行修改得到，内容上没有改变，但通过加入 CSS 和 CSS3 样式，以及 JS，是页面变得更加美观，同时也极大得提升了用户体验。

本页面实现的灵感来自于 Bootstrap 教程页面，旨在为用户提供更好的入门 HTML 的材料。另外，由于页面中大部分核心内容都是通过 js 实现，比如自动生成目录，自动生成预览框隐显按钮等。所以，可以通过简单的在 HTML 页面中添加更多的介绍，就可以实现页面内容的扩展。

2016 年 11 月 11 日

[THE END]