“日向的时光机”个人网站的设计与开发

——详细设计

学号	姓名	负责内容	成绩
2018051074	**陈孟麟**	**网站设计**	

目 录

1 概述

本网站为个人网站，面向有前端设计基础的个人，有个人网站前端基础需求的爱好者，提供 UI 设计思路和网页基础框架。

本次网站设计只涉及前端基础部分，后端部分没有设计，仅仅只掌握了 HTML、CSS、JavaScript 基础部分，所以对于网站的不同内容、框架的复用毫无头绪，所以本网站仅适用于体现网站效果和 UI 设计，bug 和效率问题并未达标，而且几乎没有考虑不同浏览器的兼容问题，不推荐实际使用。

对于导航栏有些同理的二级和三级页面，因为是简单的复制工作，为了减少代码的冗余和避免做无用功，没有实现，在以后学习了后端知识和一些框架后，再考虑系统的做一个合格的网站也不迟。

2 设计规范说明

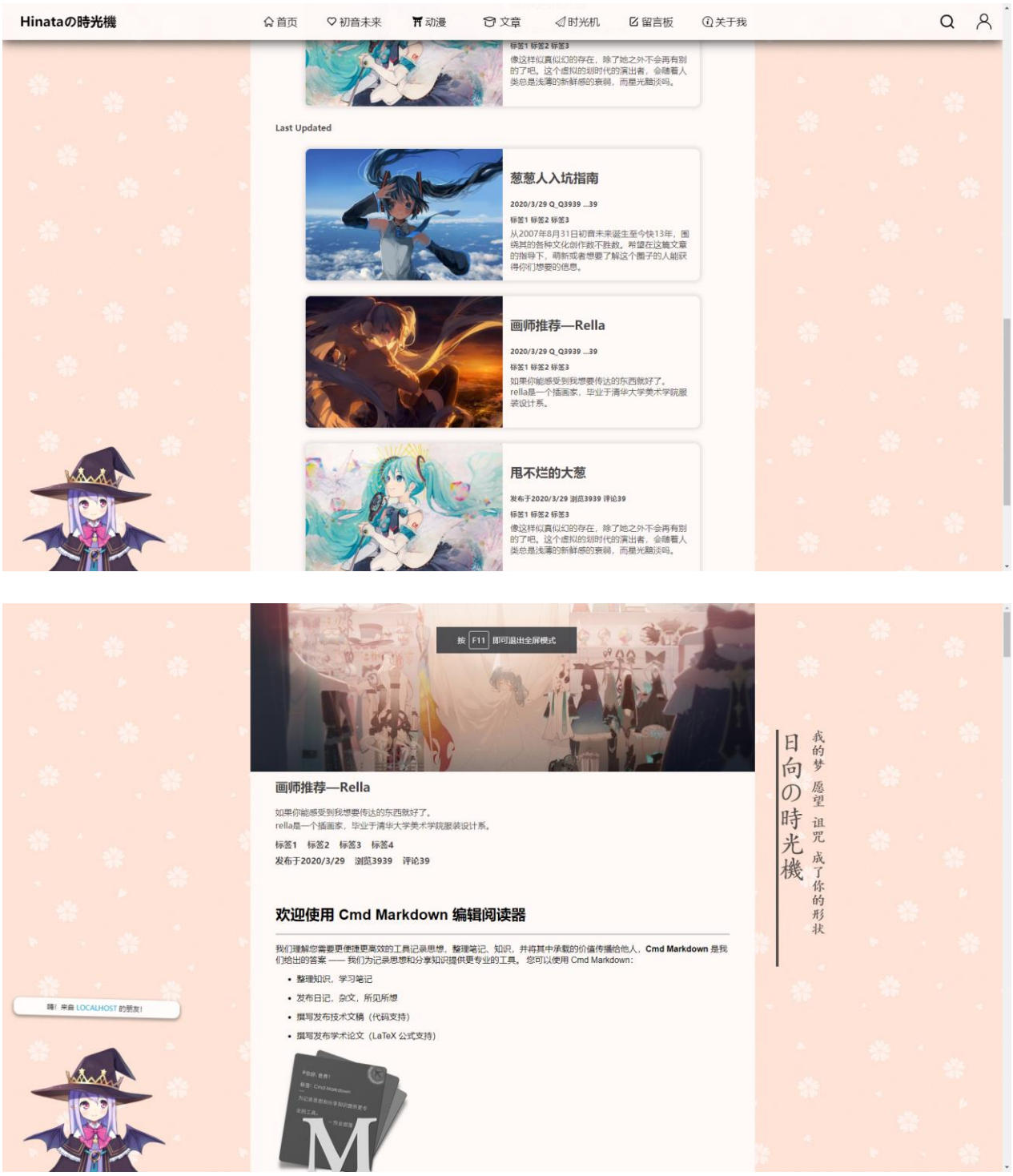
2.1 界面尺寸

由于本网站是一个个人网站，其中内容不会太多，主要体现在用户的文章阅读体验上。所以设计了文章核心部分宽度为960px，恰好为主流16：9即1920px：1080px显示器的一半，用户可以分屏观看文章，而与全屏时的体验差距不会太大。除此之外，考虑到为移动端做一个页面体系比较耗时，个人移动设备繁多，而且本网站和本人倾向于使用电脑，所以采用了一部分的响应式设计，使用了大量的%和vh、vw等单位设计元素尺寸，这样不管在移动端还是PC端的效果都比较不错（至少可以看了）。

2.2 界面风格

由于本人爱好和网站主题原因，界面风格偏日系的卡通和温暖，以及学术派的简洁。

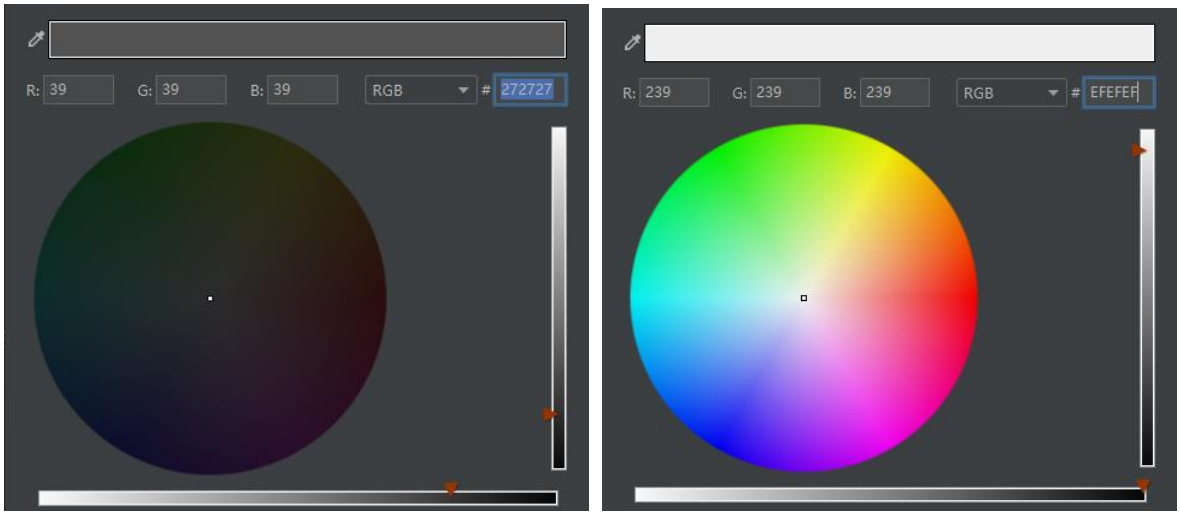
考虑到可能长时间观看一个文章，所以网站整体颜色偏清淡，文章背景也不是纯白色，字体也不是纯黑色，以及网站内容尽可能归类减少，尤其是在三级页面，给人良好的阅读体验。


Hinataの時光機
首页
初音未来
动漫
文章
时光机
留言板
关于我
Last Updated
葱葱人入坑指南
画师推荐—Rella
甩不烂的大葱
按 F11 即可退出全屏模式
画师推荐—Rella
如果你能感受到我想要传达的东西就好了。
rella是一个插画家，毕业于清华大学美术学院服装设计系。
欢迎使用 Cmd Markdown 编辑阅读器
日向の時光機
我的梦 愿望 诅咒 成了你的形状
嗨！来自 LOCALHOST 的朋友！

图：界面主体风格

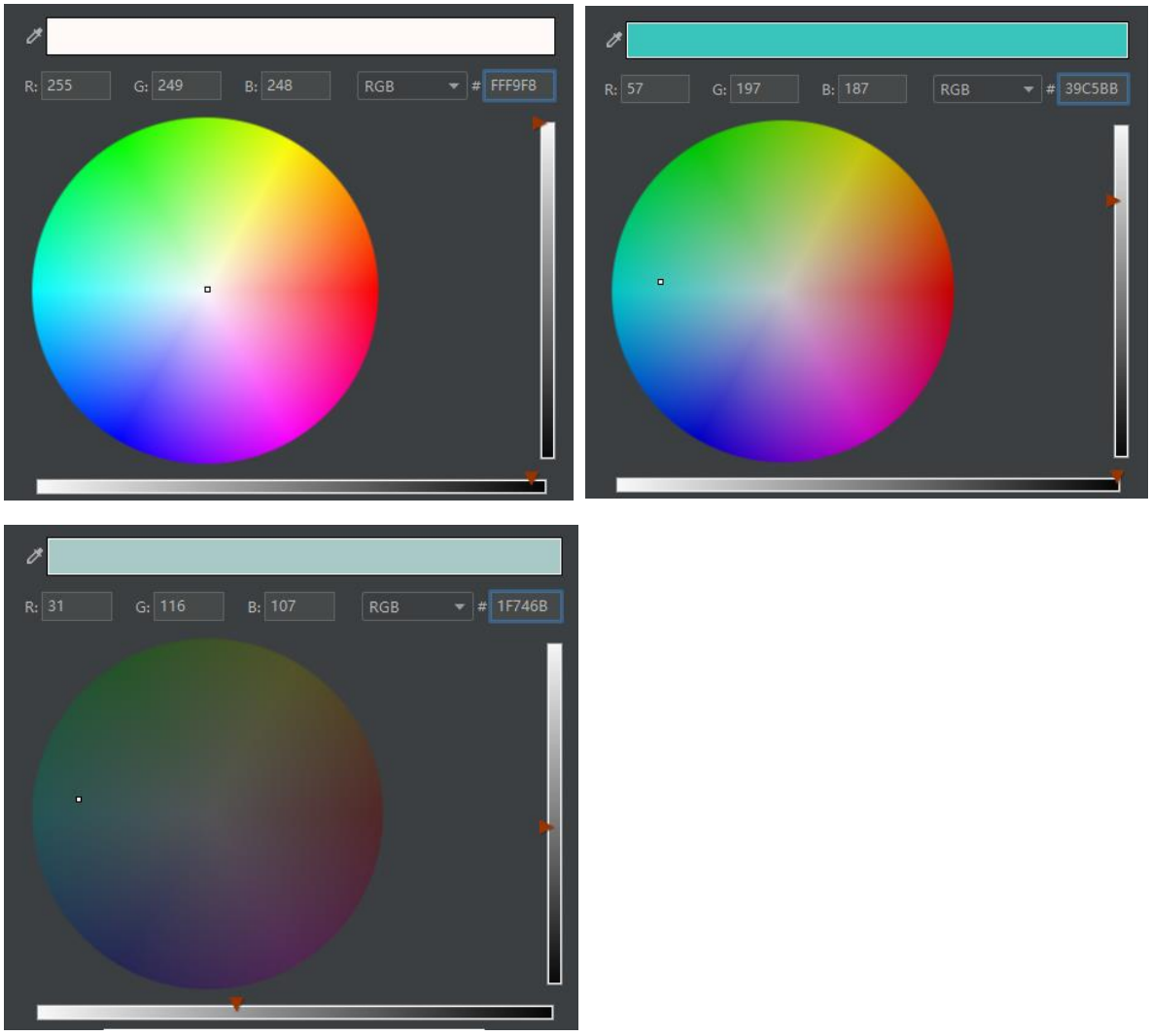
2.3 色彩方案

字体颜色：



图：字体色彩说明

背景颜色：



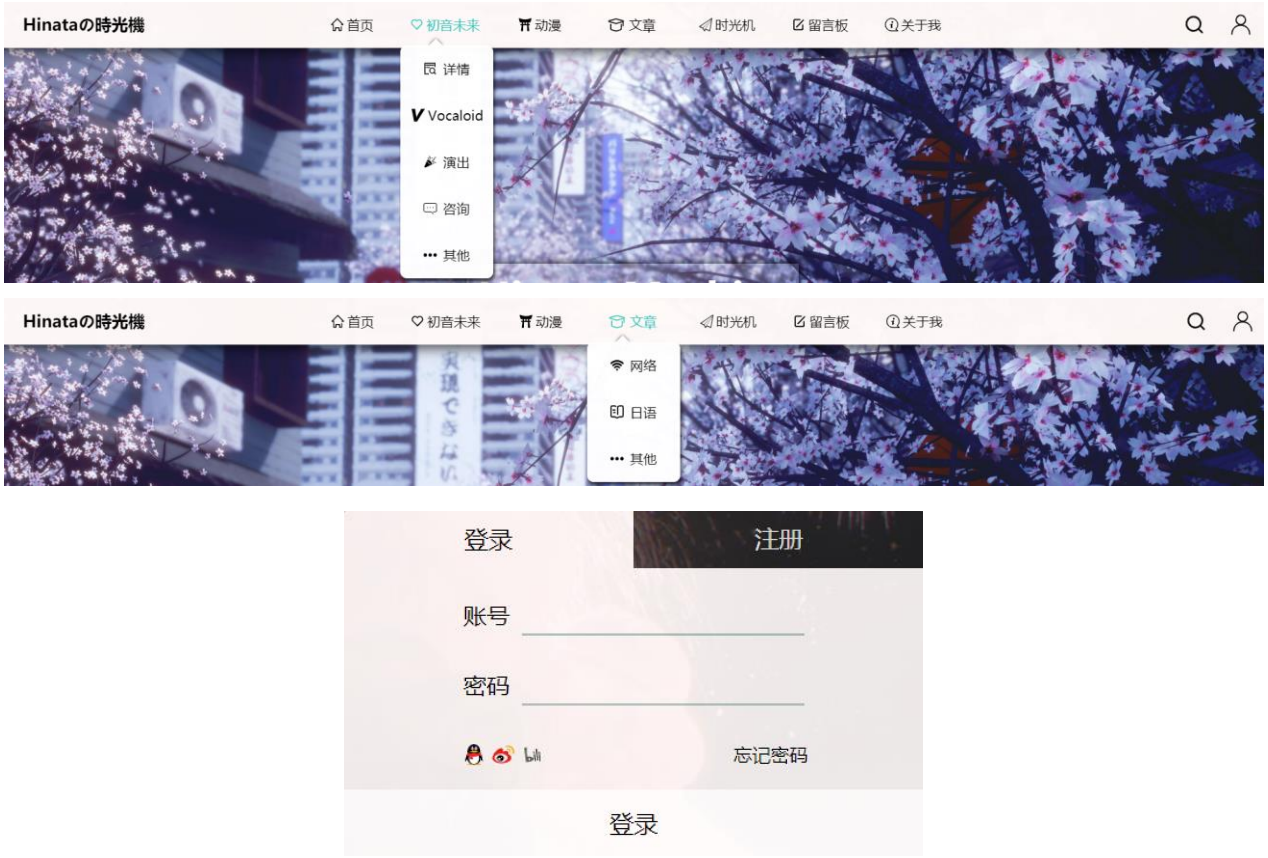
图：背景色彩说明

本网站的色彩比较简单，没有通过不同字体大小设置颜色，标题的重要性全靠字体大小体现，这样便于色彩管理，网站才不会花里胡哨费眼睛。

正如前文所说，考虑到可能长时间观看一个文章，所以网站整体颜色偏清淡，文章背景也不是纯白色，字体也不是纯黑色，以及网站内容尽可能归类减少，尤其是在三级页面，给人良好的阅读体验。

除了上面列出的颜色，还有纯黑色和纯白色的使用，不过都是根据情况加了不同的透明度，用在边框和字体的阴影和凸显字体的地方等，实现了补足的效果。

2.4 图标说明



图：图标库

图标因其具有明显示意、简洁造型、可塑性强的特点，被大量应用在界面的操作中。对于图标制作，目前市场上虽然仍使用导出图片的形式进行使用，但图形技术的革新使.svg格式的图标兴起，更多的前沿互联网公司都已经使用.svg内容开发项目。

.svg格式区别于.png、.jpg格式来说，真正意义上的实现了矢量。利用Adobe Illustrator绘图软件进行矢量制图，在.svg中可以保留所有的图形信息，通过一定的标准，可编译、描述二维矢量图形，这也造就了其尺寸更小、占用更小空间、压缩性更强、在任何的分辨率下，都可以高质量显示的几大优势。

本网站的图标全部由阿里图标库里的图标获得，全部为svg格式。

2.5 字体设计

字体：考虑到各系统的字体兼容，有以下的文字设计。

```
font-family: Lucida Console,SimHei,cursive;
```

最后一个是通用字体。除此之外，全部是默认字体。

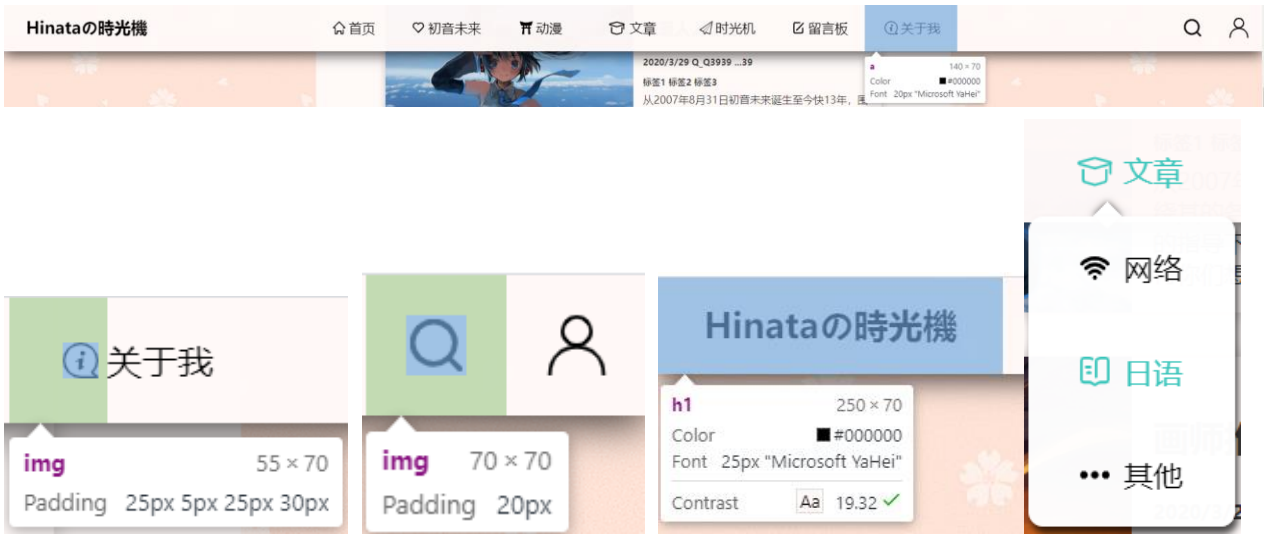
字体大小：由于网站设计灵活，字体大小根据实际情况变化很多，这里就不记录了。

不过值得一说的是关于文章的字体组织，是使用md的css文件统一组织，至少在文章正文中，字体是统一的。还有关于代码，其格式和颜色由代码高亮的prism的css和js文件统一管理。在后文中可以看到实例。

3 各模块设计介绍

注意：本部分介绍逻辑为 效果图/过程图 + 图名 +(介绍)+ （代码） 的流程。

3.1 导航栏



图：导航

导航栏分三部分组成，左、中、右，如图所示。

```
13      <nav id="nav"><div id="header_nav">
14          <div id="header_left"><h1>Hinataの時光機</h1></div>
15          <div id="header_right"...>
20          <div id="header_center"...>
47      </div></nav>
```

图：导航栏 html 框架

其中细节部分由前面的图可知，这里不详细展开。这里再讨论我根据不同页面施加在其上的 js 动画（过渡，隐藏，显示）操作。

图：导航栏默认隐藏

由图可知，刚进首页时，导航栏是隐藏的，这里是用的 opacity = 0；鼠标移入时，会从 0 过渡到 1，实现缓慢浮现效果。

图：导航栏浮现的瞬间

图：导航栏完全浮现

图：鼠标移入字体和图标变色

其中图标变色是用的网上大佬用 filter 方法矩阵算的值，其它方法在我这里因未知原因失效。

```
153     #header_center2>ul>li:hover img{
154         filter: invert(69%) sepia(78%) saturate(392%) hue-rotate(121deg) brightness(87%) contrast(83%);
155     }
```

图：鼠标移入图标变色代码

图：鼠标移入显示下拉菜单瞬间

这里下拉菜单也加入了动画，添加一个计时器改变菜单的高度和透明度。当然，在鼠标移出，即隐藏菜单或导航时也有对应的过渡动画。

```
/*鼠标移入显示下拉菜单动画*/
var arrow = document.getElementsByClassName("arrow");
var nav2 = document.getElementById("header_center2");
var nav2ul = nav2.children[2];
var nav2ulc = nav2.children[2].children.length;
var nav4 = document.getElementById("header_center4");
var nav4ul = nav4.children[2];
var nav4ulc = nav4.children[2].children.length;
nav2.addEventListener('mouseenter',function () {
    var x = 0;
    var h = 0;
    var mt = -15;
    var at = 48;
    nav2ul.style.display = 'block';
    arrow[0].style.display = 'block';
    clearInterval(nav2ul.timer);
    nav2ul.timer = setInterval(function(){
        x = x + 0.1;
        h = h + 70*nav2ulc/10;
        mt++;
        at++;
        nav2ul.style.height = h +'px';
        arrow[0].style.opacity = x - 0.4;
        nav2ul.style.opacity = x;
        nav2ul.style.marginTop = mt + 1 + 'px';
        arrow[0].style.top = at + 1 + 'px';
        if(nav2ul.style.opacity>=1) {
            nav2ul.style.opacity = '1';
            arrow[0].style.opacity = '1';
            clearInterval(nav2ul.timer);
        }
    },40);
});
```

```
/*鼠标移入显示导航栏*/
var bignav = document.getElementById("nav");//
function t(obj,ff,x){
    clearInterval(obj.timerr);
    obj.timerr = setInterval(function(){
        if(ff === 0){
            x = x + 0.1;
            bignav.style.opacity = x;
            if(bignav.style.opacity>=1) {
                bignav.style.opacity = '1';
                clearInterval(obj.timerr);
            }
        }
        if(ff === 1) {
            x = x - 0.1;
            bignav.style.opacity = x;
            if(bignav.style.opacity <= 0) {
                bignav.style.opacity = '0';
                clearInterval(obj.timerr);
            }
        }
    },50);
}
```

图：导航栏和下拉菜单过渡动画对应部分实现代码

图：在页面滚动一点距离后导航栏固定显示

```
151        if(window.pageYOffset >= document.documentElement.clientHeight && f === 0)
```

图：在页面滚动一点距离后导航栏固定显示判断代码

图：在二级页面和三级页面导航栏默认隐藏

图：在二级页面和三级页面鼠标移入后显示导航栏的瞬间

图：在二级页面和三级页面鼠标移入后完全显示导航栏

在文章页面，我想用户专注于内容，减少不必要的负担，于是导航栏也是隐藏了的，不过鉴于其依然用首页的方法直接浮动显示会遮挡页面的介绍图，不美观，于是想到将其从顶部顶开下面元素的做法，这种推动感可能比较有趣。

实现思路，在页面顶部放一个和导航栏一样大小的空 div，浮动且隐藏，鼠标移入其中时则设置计时器改变导航栏元素的高度（之前为 0），实现推动的效果，为什么不用 mousemove 的坐标，第一其耗费资源，第二我发现其实现的思路简单，但是在 js 里面会有逻辑冲突，做不出来。思路很好，可能也很糟糕。具体到细节的时候很让人抓狂，不过还算顺利做出来。于是 Bug 就来了，快速移过导航栏时，由于事件监视器可能有频率问题，在移入后检测不到移出，于是导航栏就卡在那里，要再次移入移出才能消失，可能还有导航栏闪动的问题。于是我加各种参数限制，各种骚方法，设置延迟显示等等，问题最终似乎还有，不过触发概率已经比较小了。这是本项目最难的地方。

```
/*导航栏动画*/
var nav = document.querySelector('#page_nav');
var nav_back = document.querySelector('#nav_back');
var navf = 0;
var navtime = null;
nav_back.addEventListener('mouseenter',function () {
    clearTimeout(navtime);
    navtime = setTimeout(function () {
        if(navf == 0){
            var x = 0;
            var o = 0;
            nav_back.style.display = 'none';
            clearInterval(nav.timer);
            nav.timer = setInterval(function(){
                x = x + 10;
                o = o + 0.3;
                nav.style.opacity = o;
                nav.style.height = x + 'px';
                if(nav.style.height == 70 + 'px') {
                    nav.style.opacity = '1';
                    nav.style.overflow = 'visible';
                    clearInterval(nav.timer);
                }
            },30);
        }
    },300);
    navf = 0;
});
```

```
nav.addEventListener('mouseleave',function () {
    var x = 70;
    var o = 1;
    clearInterval(nav.timer);
    nav.style.overflow = 'hidden';
    nav.timer = setInterval(function(){
        x = x - 10;
        o = o - 0.3;
        nav.style.opacity = o;
        nav.style.height = x + 'px';
        if(nav.style.height == 0 + 'px') {
            nav.style.opacity = '0';
            navf = 0;
            nav_back.style.display = 'block';
            clearInterval(nav.timer);
        }
    },30);
});
nav_back.addEventListener('mouseout',function () {
    navf = 1;
});
```

图：在二级页面和三级页面鼠标移入后导航栏推动页面显示效果代码

3.2 网站页面分类

由于只会原生 js 的基础，不知道怎么复用框架，所以目前是一个页面一个 html 文件。
本网站有一级页面（首页），二级页面，三级页面构成。

- **一级页面：**

图：网站首页

由一个背景图片，设置了宽高为 100%，能覆盖满任何屏幕，和左下角的看板娘，中间的网站标语组成，右下角的播放按钮点击后能播放站主放置的视频。往下滑动，展示站长推荐（想让观众）看到的文章视图，再下面则是最近更新的文章的视图，用于展示，文章视图的内容都是重复的，重点不在这。最后是页脚。关于文章视图，是一篇具体文章的名片之类的东西，链接着一篇具体的文章，其上由文章标题，发布日期，评论等具体信息，后面的二级页面的文章视图是同一个东西。

图：首页视频播放中，可暂停。

图：站长推荐的文章视图

图：鼠标移入，图片会放大，阴影加深

```
59      <div class="web_body_sign"><h4 style="padding-left:5%;padding-top:30px;color:rgba(0,0,0,0.7);">Exhibition</h4></div>
60      <div class="face"><a href="miku_others.html">
61          <div class="face_img" id="face1_1"></div>
62          <div class="face_text">
63              <h2>葱葱人入坑指南</h2>
64              <h5>2020/3/29 Q_Q3939 ...39</h5>
65              <h5>标签1 标签2 标签3</h5>
66              <p>从2007年8月31日初音未来诞生至今快13年，围绕其的各种文化创作数不胜数。希望在这篇文章的指导下，萌新或者想要了解这个圈子的人能获得你们想要的信息。</p>
67          </div></a></div>
68      <div class="face"><a href="miku_Realla.html"...></div>
76      <div class="face"><a href="miku_details.html"...></div>
84      <div class="web_body_sign"><h4 style="padding-left:5%;padding-top:30px;color:rgba(0,0,0,0.7);">Last Updated</h4></div>
85      <div class="face"><a href="miku_others.html"...></div>
93      <div class="face"><a href="miku_Realla.html"...></div>
101     <div class="face"><a href="miku_details.html"...></div>
```

图：文章视图的 html 结构

图：最近更新的文章视图

图：简洁的页脚

- **二级页面：**

二级页面是便于网站的分类，是为了导航到一些具体文章的集合页面。不过在导航栏中不全是二级页面，导航栏中有一些是我认为比较重要的文章也是包含在导航栏中，比如“初音未来”本身是链接到一个二级页面的，这个页面的主题为初音未来，有关于初音未来的文章在这个二级页面中以文章视图的样式被展示、被链接。而导航栏中“初音未来”的下拉菜单中，详情和其他我链接的是三级页面即一篇具体的文章，而其他几项我链接的是二级页面。

在二级页面中，有头部的轮播图，可以链接对应图片的文章，轮播图是在我实验项目中直接套用的，接下来则是二级页面对于这个页面主题的介绍信息，后面则是文章视图和首页一样，不过这些文章视图对应的都是一个主题的文章。

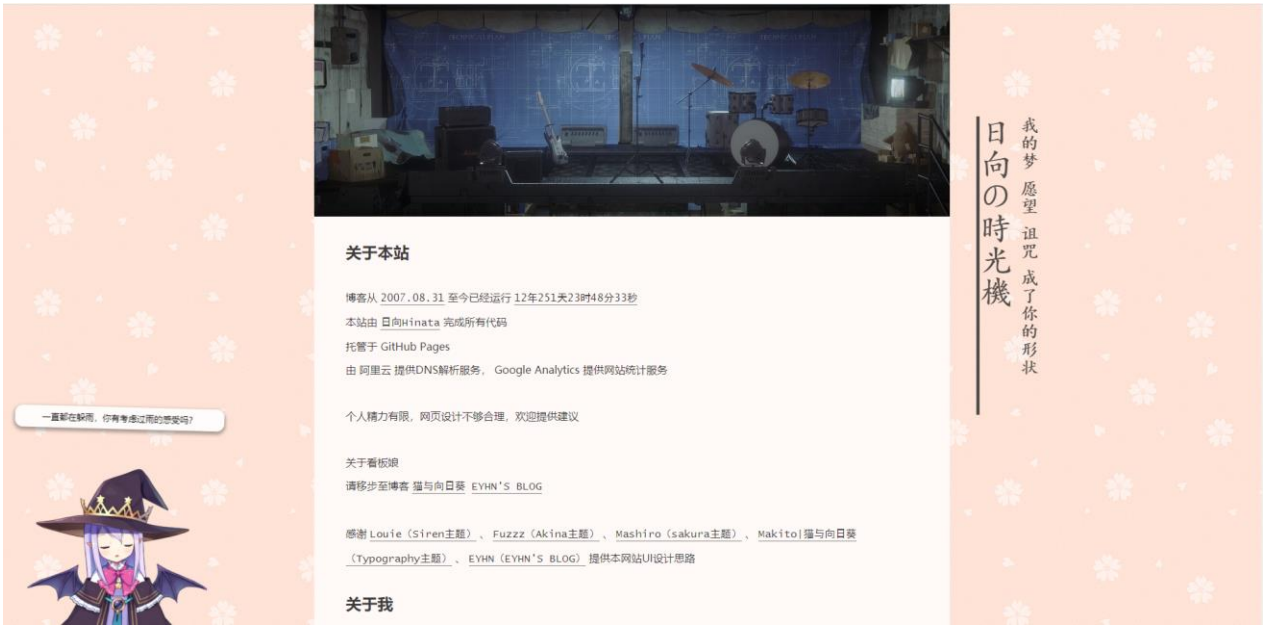
图：二级页面“初音未来”

```
55   <div id="move_pic"...>
78   <div id="web_body_head" style="..."...>
86   <div class="face"><a href="miku_others.html"...></div>
94   <div class="face"><a href="miku_Realla.html"...></div>
102  <div class="face"><a href="miku_details.html"...></div>
```

图：二级页面的框架的一部分

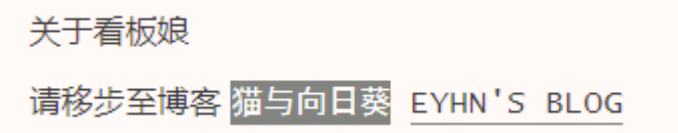
- **关于页面：**

这个页面属于二级页面，但是与前面不同，有其自己的格式。主要介绍网站的信息，版权情况，作者的信息以及赞助请求。

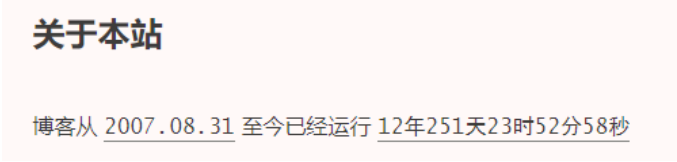


图：关于页面

这个页面内容一般很少修改，我就直接将内容写在 html 文件中了，有链接（有下划线）的文字鼠标移入会形成反差色的效果。



图：鼠标移入形成反差



图：网站记时

这里的时间我用 js 封装了一个时间函数，用计时器每隔 1s 填充时间内容，实现记时。

```
61          <p>
62              博客从<span>2007.08.31</span>至今已经运行<span id="year"></span><br/>
63              本站由<a href="#me"><span>日向Hinata</span></a>完成所有代码<br/>
```

```
<script>
    var year = document.querySelector("#year");
    var input_time = +new Date('2007-8-31 00:00:00');
    function count_time() {
        var now_time = +new Date();
        var times = (now_time - input_time) / 1000;
        var y = parseInt(times / 60 / 60 / 24 / 365);
        y = y < 10 ? '0' + y : y;
        var d = parseInt(times / 60 / 60 / 24 % 365);
        d = d < 10 ? '0' + d : d;
        var h = parseInt(times / 60 / 60 % 24);
        h = h < 10 ? '0' + h : h;
        var m = parseInt(times / 60 % 60);
        m = m < 10 ? '0' + m : m;
        var s = parseInt(times % 60);
        s = s < 10 ? '0' + s : s;
        year.innerHTML =  y + '年' + d + '天' + h + '时' + m + '分' + s + '秒';
    }
    setInterval(count_time, 1);
</script>
```

图：时间封装函数

- **登录/注册页面**

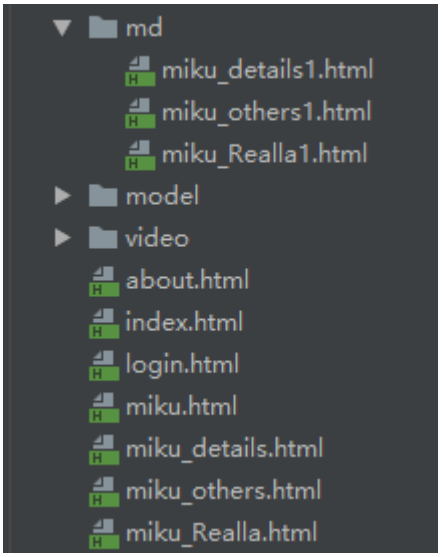
这里的点击切换操作是通过点击事件，改变相应元素的隐藏、展示属性实现的。

图：登录/注册页面

图：点击切换操作

3.3 文章内容

不知道正规的发布文章的操作，我这里是用的多建立一个 html 文件，其实就是通过 markdown 编辑器编辑后的文章转换为的 html 文件，然后在这个 html 文件中引入对应的 md 的 css 文件。再将这个 html 用 iframe 方法引入网页的三级页面中。实际中不会这么操作，iframe 也是不推荐使用的，会有高度问题，正常情况下应该是 ajax 调用 md 文件，js 渲染 md 文件为网页中的文章。



图：md 文件夹中为包含文章的 html 文件，一个文章对应一个

```
<link rel="stylesheet" href="../css/swiss.css"/>
```

Md 编辑器导出的 html 文件中引入喜欢的 css 文件，规范格式。

```
<div id="markdown_text">
    <iframe scrolling="yes" id="main" name="main" frameborder="0" src="md/miku_Realla1.html" style="..."></iframe>
</div>
<script>
    // 计算页面的实际高度，iframe自适应会用到
    function calcPageHeight(doc) {
        var cHeight = Math.max(doc.body.clientHeight, doc.documentElement.clientHeight);
        var sHeight = Math.max(doc.body.scrollHeight, doc.documentElement.scrollHeight);
        return Math.max(cHeight, sHeight);
    }
    //根据ID获取iframe对象
    var ifr = document.getElementById('main');
    ifr.onload = function() {
        //解决打开高度太高的页面后再打开高度较小页面滚动条不收缩
        ifr.style.height='0px';
        var iDoc = ifr.contentDocument || ifr.document;
        var height = calcPageHeight(iDoc);
        if(height < 850){
            height = 850;
        }
        ifr.style.height = height + 'px'
    }
</script>
```

图：引入 md 的 html 文件

用 iframe 引入 html 到三级页面中，需要用 js 获取引入页面的原始高度，以此来设置 iframe 框架的高度。

图：成功引入 md 的 html 文件

关于代码高亮，需要在原始生成的 html 文件中引入 prism 的 css 和 js 文件。

```
<link href="../css/prism.css" rel="stylesheet" />
<script src="../js/prism.js" ></script>
```

代码标签加入类名，显示行数和代码高亮

```
<pre class="line-numbers"><code class='language-C++' lang='C++'>代码
</code></pre>
```

```
47
48  // - - - - - 栈的基本操作的函数原型说明
49  Status InitStack(Stack& S);
50  Status StackEmpty(Stack S);
51  Status Push(Stack& S, SElemType e);
52  Status Pop(Stack& S, SElemType& e);
53  // - - - - - 图的关键路径基本操作函数原型说明
54  Status CreateALGraph(ALGraph& G);
55  Status PrintALGraph(ALGraph G);
56  Status TopologicalOrder(ALGraph G, Stack& T);
57  Status CriticalPath(ALGraph G, Stack T);
58  void FindInDegree(ALGraph G, int indegree[MAX_VERTEX_NUM]);
59  // - - - - - 画图的函数原型说明
60  void DrawGraph(ALGraph G);
61  void lineArrow(int x1, int y1, int x2, int y2);
62
63  int main() {
64      ALGraph G;
65      printf("/******求图的关键路径******/\n\n\n");
66      Stack T;    //图的一个拓扑序列
67      CreateALGraph(G);
68      PrintALGraph(G);
69      if (TopologicalOrder(G, T) == ERROR) {
70          printf("该有向网有回路，无法求取关键路径！\n");
71          return 0;
72      }
```

图：漂亮的代码高亮和行数生成了

3.4 侧边文字和一键返回滚动条

```
24  }Stack;
25
26  // - - - - - 图的邻接表存储表示
27  #define MAX_VERTEX_NUM 20          //图的顶点存储空间初始分量
28  typedef struct ArcNode {
29      int adjvex;                    //该弧所指向的顶点的位置
30      struct ArcNode* nextarc;       //指向下一条弧的指针
31      InfoType* info;                //该弧相关信息的指针
32      int flag = 0;
33  };
34  typedef struct VNode {
35      VertexType data;               //顶点信息
36      ArcNode* firstarc;             //指向第一条依附该顶点的弧的指针
37      int x;                         //顶点x坐标，用户定义，用于打印
38      int y;                         //顶点y坐标，用户定义，用于打印
39  }VNode, AdjList[MAX_VERTEX_NUM];
40  typedef struct {
41      AdjList vertices;
42      int vexnum, arcnum;            //图的当前顶点数和弧数
43      int kind;                      //图的种类标志
44  }ALGraph;
45
46  int ve[MAX_VERTEX_NUM];            //全局变量，表示事件的最早发生时间
47
48  // - - - - - 栈的基本操作的函数原型说明
49  Status InitStack(Stack& S);
50  Status StackEmpty(Stack S);
51  Status Push(Stack& S, SElemType e);
52  Status Pop(Stack& S, SElemType& e);
53  // - - - - - 图的关键路径基本操作函数原型说明
54  Status CreateALGraph(ALGraph& G);
55  Status PrintALGraph(ALGraph G);
56  Status TopologicalOrder(ALGraph G, Stack& T);
57  Status CriticalPath(ALGraph G, Stack T);
58  void FindInDegree(ALGraph G, int indegree[MAX_VERTEX_NUM]);
59  // - - - - - 画图的函数原型说明
60  void DrawGraph(ALGraph G);
61  void lineArrow(int x1, int y1, int x2, int y2);
```


图：侧边文字和一键返回滚动条

- **侧边文字**

主要是为了体现网站的名称和标语及美观补白。
在网站被加载时，整个内容页有从上往下滑动，侧边文字有从右往左滑动的视觉落差效果，这在博客网站中十分常见。

图：网页刚开始加载瞬间

图：网页加载完成，元素归位

图：鼠标移入侧边文字，有文字反色差，缓动效果

- **一键返回滚动条**

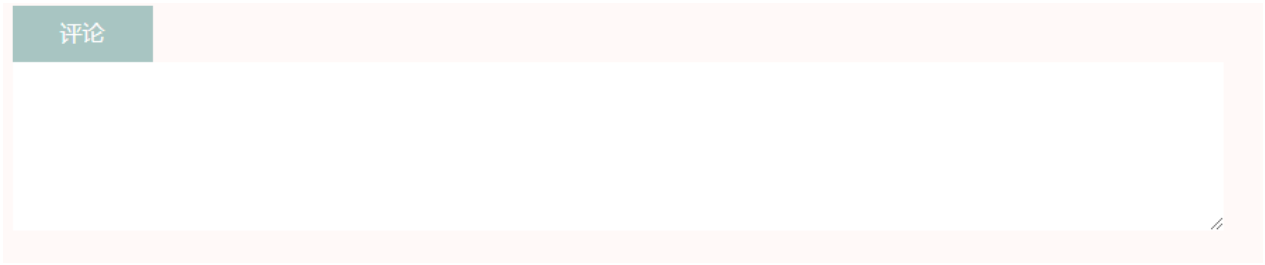
在浏览网页时，可以通过点击返回条回到网页顶部，不过感觉有点丑，以后再改。

图：滚动条的滚动瞬间

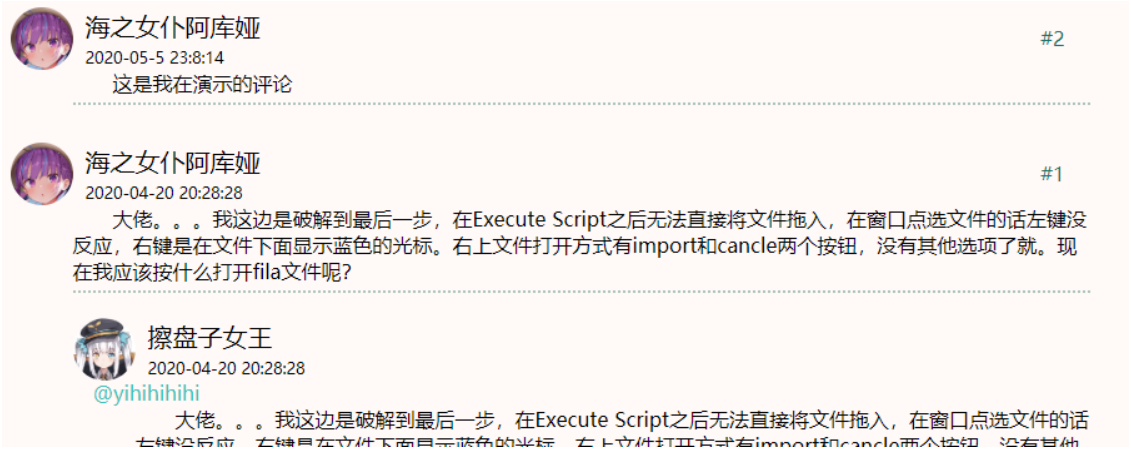
3.5 评论

图：评论模块

在三级页面的底部有评论模块，用 js 的节点复制功能实现了简陋的评论功能（模拟）。

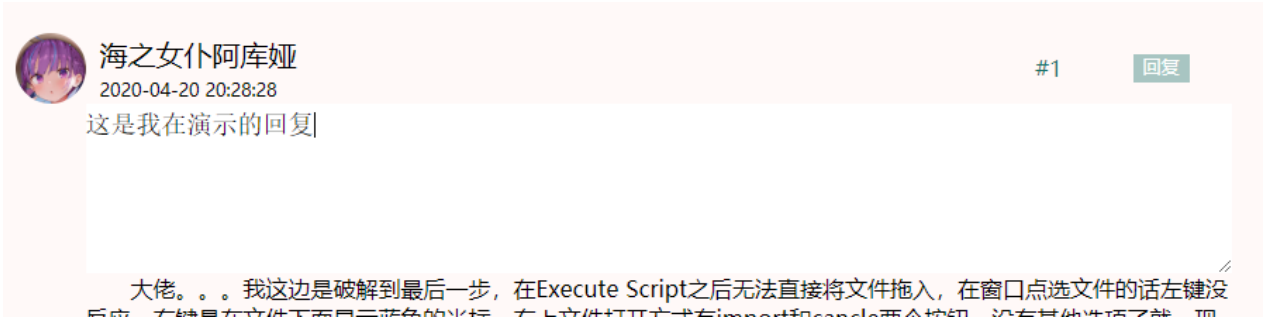


图：点击评论后弹出输入框。

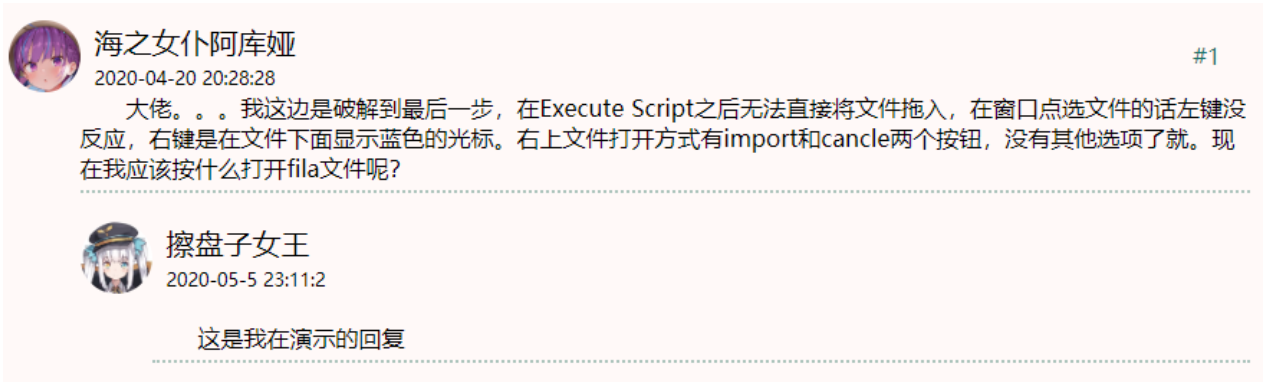


图：评论添加

对比之前的评论，评论已经添加成功，楼层数也是对的为 2



图：回复，鼠标移入一条评论，会展示回复按钮



图：回复成功

图：回复回复，鼠标移入一条回复，会展示回复按钮

图：回复回复成功

这里回复回复需要添加一个被回复人的信息，如图中的@***所示。

实现思路很简单，点击评论/回复，输入信息，js 复制一个原先我设置好的回复或评论模板（模板是隐藏了的），往复制的模板中添入信息，删除其作为模板的 id 或 class 属性，添加到相应位置即可。

```
/*回复回复交互*/
var reply_flag = 0;
var reply_textarea_2;
var reply_button = document.querySelectorAll(".reply_button");
for(i = 0;i < reply_button.length; i++) {
    reply_button[i].addEventListener('click', function () {
        var reply_ul = this.parentNode.parentNode.parentNode;
        if (reply_flag === 0) {
            reply_textarea_2 = comment_textarea.cloneNode();
            reply_textarea_2.style.width = "calc(100% - 50px)";
            reply_textarea_2.style.marginLeft = "50px";
            reply_textarea_2.style.display = 'block';
            this.parentNode.parentNode.insertBefore(reply_textarea_2, this.parentNode.parentNode.children[1]);
            reply_flag = 1;
        } else {
            if (reply_textarea_2.value) {
                var reply_model = document.querySelector("#reply_model");
                var reply_plus = reply_model.cloneNode(true);
                reply_plus.style.display = 'block';
                reply_plus.children[2].innerHTML = reply_textarea_2.value;
                reply_textarea_2.parentNode.removeChild(reply_textarea_2);
                reply_plus.children[0].children[0].children[1].children[1].innerHTML = timef();
                reply_plus.children[1].innerHTML = "@" + this.parentNode.children[0].children[1].children[0].innerHTML;
                reply_ul.insertBefore(reply_plus, reply_ul.children[0]);
                reply_plus.removeAttribute("id");
                reply_flag = 0;
            }
            reply_textarea_2.style.display = 'none';
        }
    });
}
```

图：这里展示最复杂的回复回复的 js 代码

3.6 live2d 看板娘

图：初次进入的欢迎语句与鼠标点击的互动

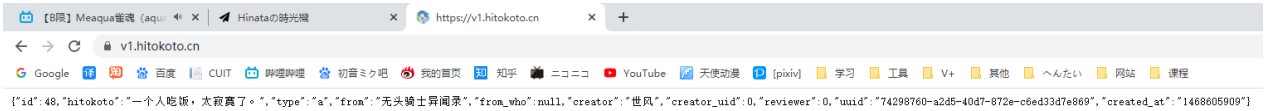
图：不同时间段不同的行为（现在 23 点，正在睡觉）以及视线跟随鼠标行为

关于看板娘，这部分的内在逻辑我也不懂，完全是看着好玩来完善丰富页面套用到的（尽管如此过程也不容易），关于其在网页中的应用，涉及到了 ajax 和 jQuery 等我的知识盲区，更深的地方则涉及到了对 live2dviewer 的 API 的调用与封装（应该是吧），在这里我只修改一部分 js 和 css 代码供我使用。

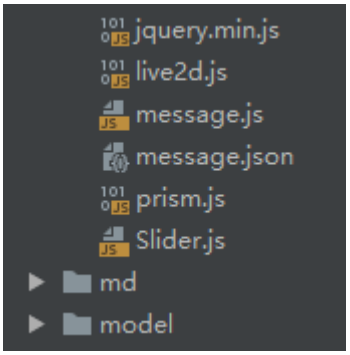
图：阅读行为（骚话）

```
106     function showHitokoto(){
107         $.getJSON('https://v1.hitokoto.cn/',function(result){
108             showMessage(result.hitokoto, 5000);
109         });
110     }
```

图：阅读的内容由这个程序设定



图：这个网址会不间断随机更新语言，live2d 看板娘会获取并展示出来



图：live2d 文件组织

其中前四个 js 文件是 live2d 用到的，model 文件夹是模型文件夹，可替换为其他模型。其中 live2d.js 是封装的 api（吧，看不懂），message.json 是管理用户鼠标点击/移入等需要展现什么对话内容的交互数据的，message.js 则是处理其他自动的动态内容的。

```
<div id="live">
    <div id="landlord">
        <div class="message" style="opacity:0"></div>
        <canvas id="live2d" width="280" height="250" class="live2d"></canvas>
    </div>
    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript">
        var message_Path = '';
        var home_Path = 'http://localhost:63342/';
    </script>
    <script type="text/javascript" src="js/live2d.js"></script>
    <script type="text/javascript" src="js/message.js"></script>
    <script type="text/javascript">
        loadlive2d("live2d", "model/pio/model.json");
    </script>
</div>
```

图：在 html 中调用看板娘

到这里，可以自己添加/修改内容。