

脚本程序设计期末项目——ToDo List

李璐 1652811

1 功能简介

1.1 基本功能

- (1) 新增、删除、展现列表
- (2) 全部完成 / 取消
- (3) 删除已完成
- (4) 使用LocalStorage保存页面状态，刷新页面后可恢复

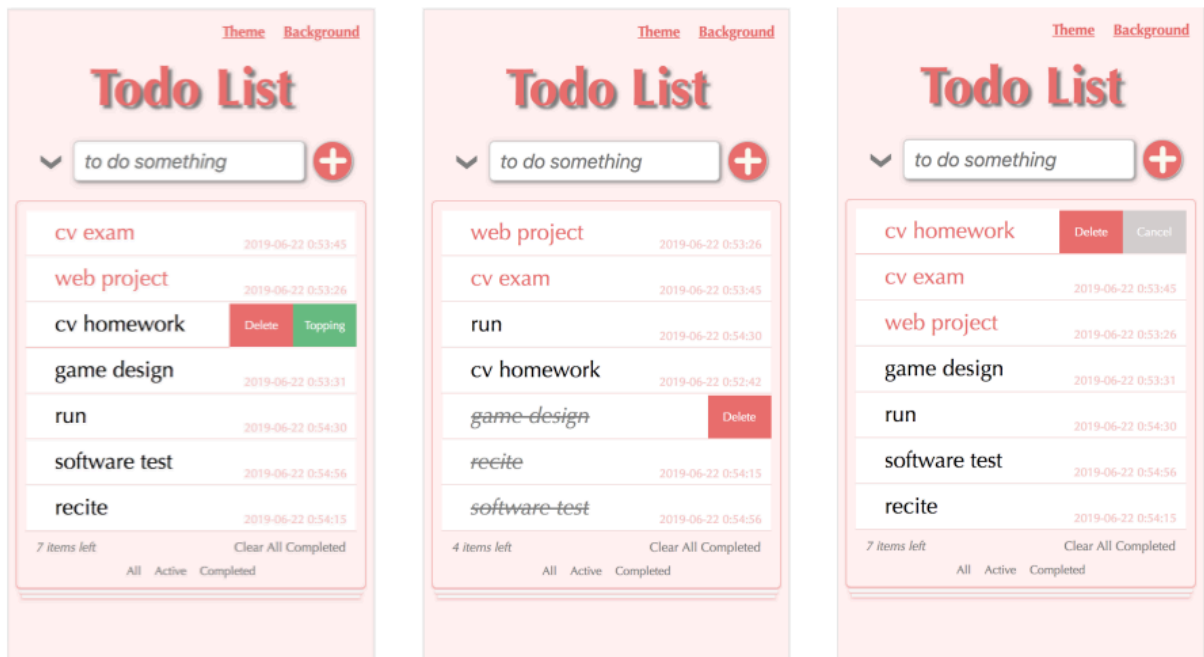
1.2 高级功能（亮点）

- (1) 过滤todo条目（All, Active, Complete）
- (2) 编辑单条todo
- (3) 置顶与取消置顶单条todo
- (4) todo条目列表内滚动
- (5) 更改主题
- (6) 自定义背景颜色
- (7) 左滑手势选择删除&置顶&取消置顶
- (8) 添加时间
- (9) 置顶、未完成、已完成条目的排序展示

2 部分高级功能简介与展示

2.1 置顶与取消置顶单条todo

左滑条目点击置顶后（图一），该条目会被展示在列表最顶端，并且改变字体颜色。当改条目被置顶后，左滑的按钮变为取消置顶（图三），点击取消置顶，条目将会变回正常状态，展示于所有置顶条目之下。若一项条目为已完成状态，则不可以进行置顶操作，只有删除按钮（图二）。



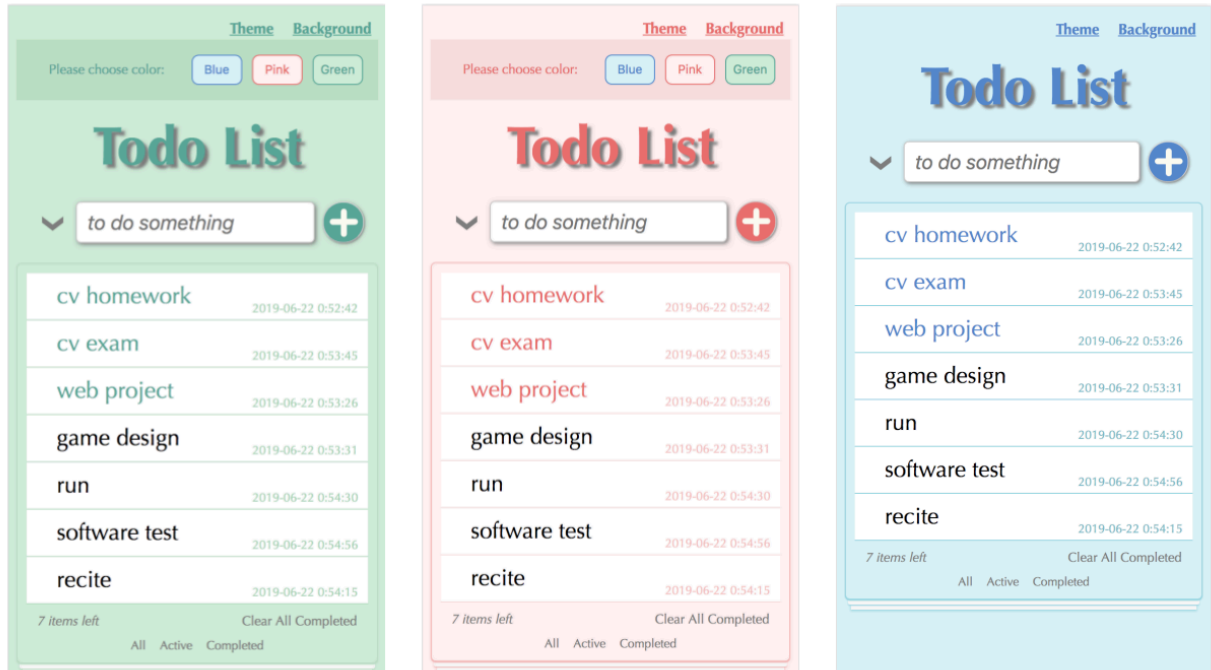
2.2 左滑手势

在对应条目上左滑将会展示出删除 / （取消）置顶按钮。这里主要使用的是hammer.js库，通过swipeleft识别左滑手势。展示图如2.1。

```
Hammer_item.on("swipeleft", function () {  
    ...  
});
```

2.3 更改主题

点击页面上方的Theme按钮将会展示三种不同颜色的主题，点击某一颜色按钮即可改变对应主题。主题状态将会被保存，刷新页面后依然会刷新前的主题。



这里我们用到alternate属性，用来替换css文件。link标签中的rel属性值为stylesheet，就是当前页面所加载显示的css样式文件。而alternate stylesheet就是用来做替换用的，写在head中时不会被显示出来，但是会加载到浏览器中。html部分如下：

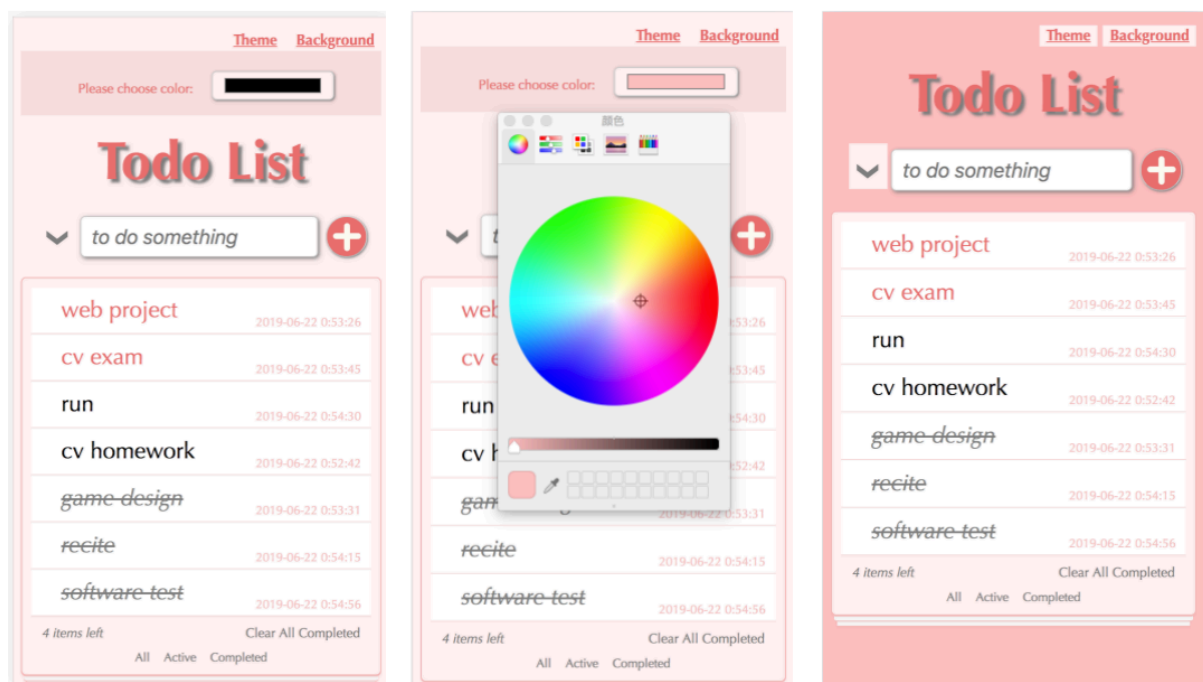
```
<head>
  <meta content="width=device-width,initial-scale=1,user-
scalable=no,maximum-scale=1" name="viewport" />
  <link rel="stylesheet" type="text/css" href="todolist-
pink.css" title="pink">
  <link rel="alternate stylesheet" type="text/css" href="todolist-
blue.css" title="blue">
  <link rel="alternate stylesheet" type="text/css" href="todolist-
green.css" title="green">
  <script src="hammer.min.js"></script>
</head>
```

然后写一个js函数进行触发、替换。首先找到DOM中所有的link标签，然后找到所有link中rel属性值包括style的，也就是包括stylesheet和alternate stylesheet的。将符合条件的link的disabled的属性设为true，都改为禁用，然后判断link标签中的title属性，找到需要替换的css文件，最后将该link的disabled改为启用。

```
function setStyleSheet(title){
    var link_list = document.getElementsByTagName("link");
    if ( link_list ){
        for ( var i=0;i<link_list.length;i++ ){
            if ( link_list[i].getAttribute("rel").indexOf("style") != -1 ){
                link_list[i].disabled = true;
                if ( link_list[i].getAttribute("title") === title){
                    link_list[i].disabled = false;
                }
            }
        }
    }
}
}
```

2.4 自定义更改背景颜色

点击页面上方的Background按钮即可点击颜色框选择颜色，确定后即可改变背景颜色。但在手机端没有选择颜色的input，我们需要手动输入颜色值。



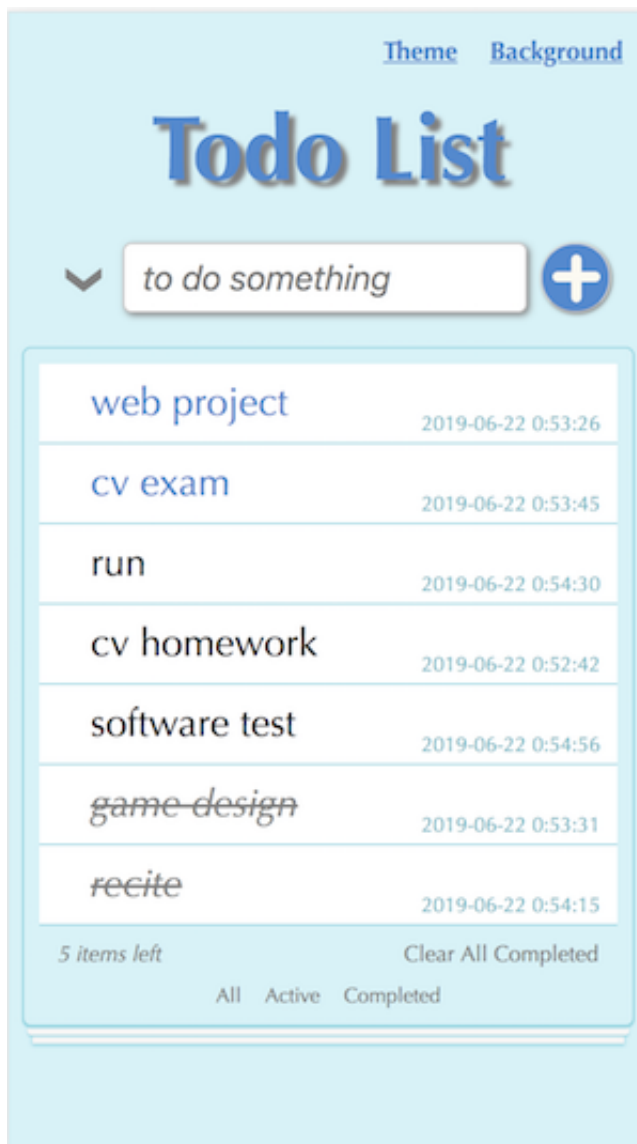
实现方法较为简单，即将type为color的input的value赋给body的background-color。

```
<input type="color" id="bg">
```

```
$( 'bg' ).addEventListener( 'change', function() {
    document.body.style.background = this.value;
} );
```

2.5展示条目的排序

列表将以置顶条目、未完成条目与已完成条目的顺序进行展示，当一个条目改变状态时，会实时改变位置到其相应的位置上。如下图：蓝色为置顶条目、黑色为未完成条目、灰色带划线为已完成条目。



在model.data.items中加入previlage字段以记录该元素是否被置顶，在model.data中加入topping字段记录目前列表中被置顶的条目数。

```
window.model = {
  data: {
    items: [
      // {msg:'',
      // completed: false,
      // previlage:'0'}
    ],
    msg: '',
    filter: 'ALL',
    theme: 'pink',
    topping: '0',
    time: ''
  },
}
```

```
};
```

在新建条目时，以及update()函数中，讲未完成的条目插在置顶条目的后方，通过计算当前条目总和 - 置顶条目得到插入位置。

```
data.items.splice(index, 1);  
data.items.splice(data.items.length - data.topping, 0, entry);
```

已完成的条目则被直接插入到数组最前（因为生成条目时使用的是insertBefore，所以在数组前面的条目在列表中将在较后面的位置。

```
data.items.splice(index, 1);  
data.items.unshift(entry);
```

置顶条目同理，直接插入到数组最后。

```
data.items.push(entry);  
data.items.splice(index, 1);  
entry.priority = 1;
```