**源码健身Fitness移动Web站**

**首页&我的Day03**

# 今日任务概览

**完成页面：首页 / 我的（静态布局 + 接口联调）**

首页 我的



附：

设计稿地址：[设计稿链接](https://modao.cc/app/629ff0afd82a66bebb7e0d87b726d618140faf20?simulator_type=device&sticky)

# 修炼技术点

* 第三方轮播插件Swiper.js的使 用
* Flex布局
* 使用JS动态渲染底部的Tab栏，Fixed布局
* 阿里IconFont（字体图标）的使用
* Ajax请求接口联调、PostMan的使用

# 首页开发

## Swiper.js的使用

英文官网（最新版）：<https://swiperjs.com/get-started/>

Swiper中文网-在线演示：<https://www.swiper.com.cn/demo/index.html>

Swiper - GitHub：<https://github.com/nolimits4web/swiper>

使用它，依然按照一般引入第三方插件的步骤：

1. 安装，引入CSS和JS（Installation）
2. 添加HTML布局（Add Swiper HTML Layout）
3. 添加Swiper容器的样式（Swiper CSS Style/Size）
4. 在页面JS中初始化配置Swiper轮播功能（Initialize Swiper）

HTML：

<!-- 轮播区域 -->

<div class="swiper-container">

<div class="swiper-wrapper">

<!-- Slides -->

<div class="swiper-slide index-swiper1"></div>

<div class="swiper-slide index-swiper2"></div>

<div class="swiper-slide index-swiper3"></div>

</div>

<!-- If we need pagination -->

<div class="swiper-pagination"></div>

</div>

CSS：

/\* 图片轮播 \*/

.swiper-container {

width: 100%;

height: 200px;

}

.index-swiper1 {

background-image: url(../image/index-swiper-bg1.jpg);

background-position: top center;

background-size: 500px;

}

.index-swiper2 {

...

}

.index-swiper3 {

...

}

JavaScript：

// 设置轮播 Swiper：Swiper API：https://swiperjs.com/api/#initialize

const swiper = new Swiper('.swiper-container', {

loop: true, // 允许图片循环

pagination: { // 显示轮播分页

el: '.swiper-pagination',

},

autoplay: true // 自动播放

});

## 入口区域使用Flex布局占比

如图效果，通过Flex布局设置子元素的flex占比，可以轻松实现：



HTML：

<!-- 今日排名 & 累积打卡 -->

<div class="row-container">

<div class="row-card row-card-ranking mr16">

<p>今日排名</p>

<p id="rankCount" class="rank-count">--</p>

</div>

<div class="row-card row-card-sum">

<p>累积打卡<span id="punchSum">--</span>天</p>

<div id="checkBtn" class="checkbtn">--</div>

</div>

</div>

Less：

.row-container { /\* 行容器 \*/

display: flex;

height: 150px;

}

.row-card {

...

}

.row-card-ranking { /\* 今日排名 \*/

flex: 2;

...

}

.rank-count {

...

}

.row-card-sum { /\* 累积打卡 \*/

flex: 3;

...

}

## 字体图标IconFont的使用

为何要使用字体图标？

1. 传统方式下，一个图标对应一个图片文件，而使用字体图标，所有图标整合到1个字体文件中，大大减少了HTTP请求资源的次数；
2. 位图的图标图片，在放大后会失真，没有办法1张图片应用于不同大小的图标上。而通过font-size设置字体大小没有这种限制，即便图标很大也不会占用多余的资源空间；
3. 通过设置文字的color可以改变图标颜色（而使用确定的图片则不能），不过限制是必须是单色图标；

常见的字体图标方案有Iconfont、font-awesome、fontello等，目前最主流的是IconFont

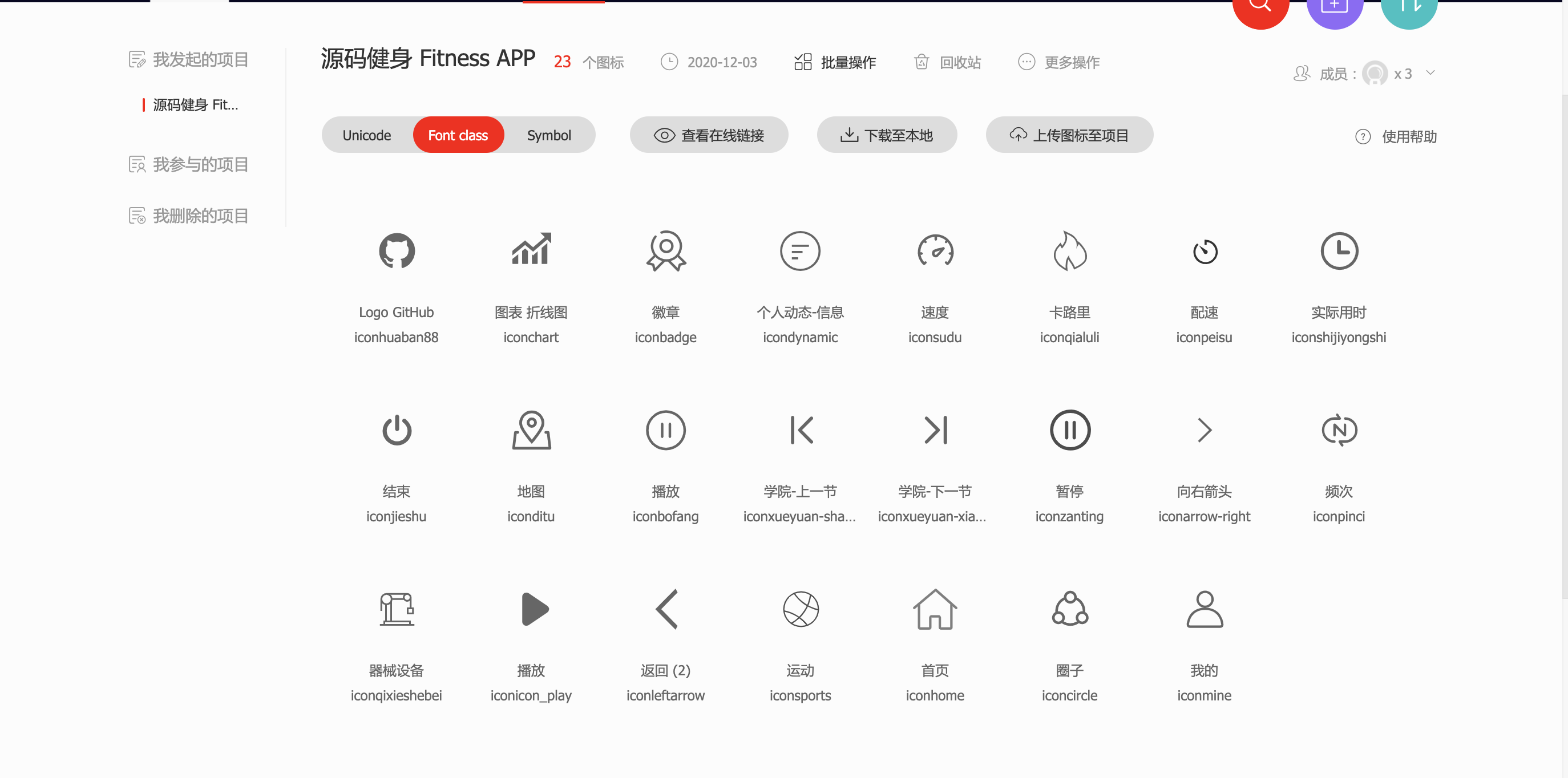
IconFont官网：<https://www.iconfont.cn/home/index>

使用流程：

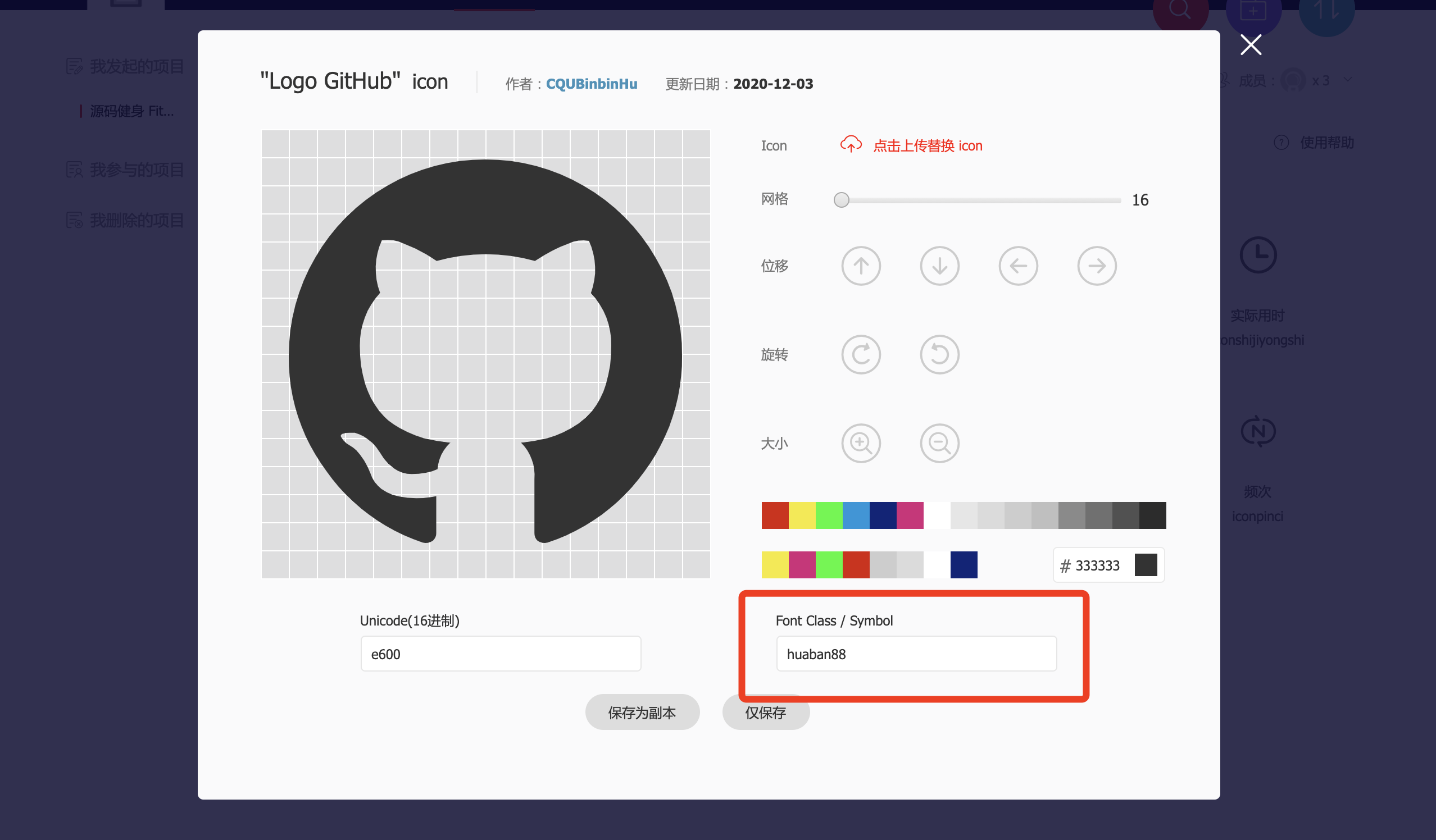
Step1：在Iconfont网站上搜图标



1. 添加图标至项目库：



1. 编辑图标，可以修改class的名字：



1. 下载后使用

## 使用JS动态添加Tab导航栏

**技术方案解析**-关于Tab导航栏的实现，要考虑以下几方面：

1. 使用Fixed布局固定到浏览器窗口底部，body加padding-bottom等高占位；
2. 移动端用户使用手指操作，而PC端使用鼠标操作，所以移动端的可交互区域一定要较大；
3. 使用字体图标Iconfont，方便地实现Tab页的选中/未选中状态；
4. 封装成JS函数，CSS布局提为公共样式，便于复用；
5. 通过函数传参，来区分页面；

JavaScript：

// 在页面的底部添加 Tab 导航栏

// currentTabType 可以是 'index' / 'sports' / 'circle' / 'mine'

function addTabs(*currentTabType*) {

const body = document.querySelector('body');

// 1.创建 Tabs 元素

const tabs = document.createElement('div');

// 2.为元素设置属性

tabs.setAttribute('class', 'tabs');

// 3.拼接 HTML 字符串

const toSetHTML =

'<a href="./index.html" class="tabs-tab"> \n' +

'<div><span class="iconfont iconhome tabs-icon' + (currentTabType === 'index' ? ' tabs-crt' : '') + '"></span></div>\n' +

'<p class="tabs-title">首页</p>\n' +

'</a>\n' +

'<a href="./sports/run.html" class="tabs-tab">\n' +

'<div><span class="iconfont iconsports tabs-icon' + (currentTabType === 'sports' ? ' tabs-crt' : '') + '"></span></div>\n' +

'<p class="tabs-title">运动</p>\n' +

'</a>\n' +

'<a href="./moments/moments.html" class="tabs-tab">\n' +

'<div><span class="iconfont iconcircle tabs-icon' + (currentTabType === 'circle' ? ' tabs-crt' : '') + '"></span></div>\n' +

'<p class="tabs-title">圈子</p>\n' +

'</a>\n' +

'<a href="./mine.html" class="tabs-tab">\n' +

'<div><span class="iconfont iconmine tabs-icon' + (currentTabType === 'mine' ? ' tabs-crt' : '') + '"></span></div>\n' +

'<p class="tabs-title">我的</p>\n' +

'</a>\n';

// 4.填充元素的 HTML 内容

tabs.innerHTML = toSetHTML;

// 5.将元素挂载至 body

body.appendChild(tabs);

}

window.utils = {

addTabs: addTabs

}

## 获取LocalStorage中的userId

要想获取首页的个人信息，前提是有userId来发请求。登录成功后会将用户数据存入LocalStorage中，其中就有各页面发请求要用的userId。

需要注意的是，如果用户在没有本地存储数据的情况下直接访问首页，那么需要让其跳转到登录页登录。

JavaScript：

const userData = JSON.parse(localStorage.getItem('user'));

if (!userData) { // 若无用户信息，则返回登录

location.href = './login.html';

}

const userId = userData.userId;

# 我的页面开发

技术点解析：

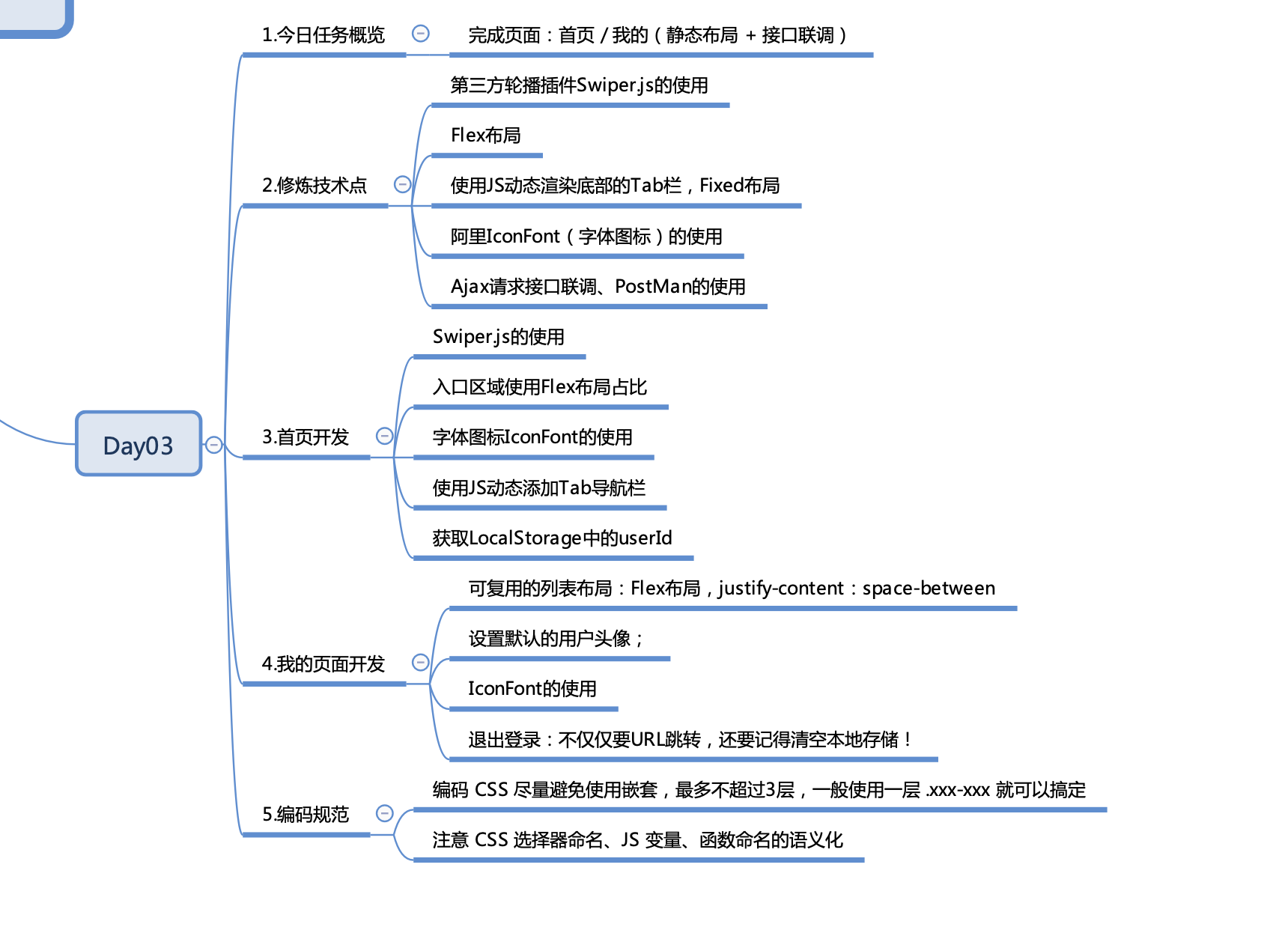
1. 可复用的列表布局：Flex布局，justify-content：space-between;
2. 设置默认的用户头像；
3. IconFont的使用
4. 退出登录：不仅仅要URL跳转，还要记得清空本地存储！

# 今日研发任务CheckList

* 首页：
  + 补全轮播图功能：自动滚动、循环滚动、分页器、背景图适配
  + 入口区域的Flex布局
  + 使用JS动态添加Tab栏
  + 获取userId并请求用户数据，回填到DOM
* 我的：
  + 设置默认的用户头像
  + IconFont的实践
  + 布局、请求数据与回填
  + 退出登录逻辑
* 预习weui

PS：使用Git进行代码版本控制，commit格式：git commit -m “feat: 完成注册页背景图适配”

# 今日小结



## 编码最佳实践

* 使用第三方库、或是比较不常用的API，可以注释API文档地址；
* 对于图标，优先使用字体图标技术，而不是一张张引图片
* 写CSS选择器，尽量减少层级嵌套，使用一层class已可以解决绝大多数场景；

## 编码意识

* 跨页间存储、通讯，考虑使用本地存储技术；
* 注意CSS选择器、JS变量、函数的命名