

## 调研与选型

### 一、前端框架

#### 1. 备选的3个框架分析

##### 1.1 Vue

##### 1.2 React

##### 1.3 Angular

#### 2. 最终选择 - vue

### 二、web端样式库

#### 1. 备选的8个Web端组件库简介

#### 2. 最终选择 - Ant Design

### 三、数据库

#### 1. 可选方案

#### 2. 纵向比较

##### 2.1 MySQL

##### 2.2 SQL Serve

##### 2.3 Oracle

#### 3. 最终选择 - MySQL

### 四、Python版本

#### 1. Python综述

#### 2. Python2与Python3的选择

### 五、编辑器

#### 1. 可选方案

#### 2. 最终选择 - Visual Studio Code

### 六、服务端框架

#### 1. 备选的2种框架分析

##### 1.1 Django

##### 1.2 Flask

#### 2. 二者比较

#### 3. 最终选择 - Django

### 七、在线编程语言

#### 1. 可选方案：

#### 2. 最终选择 - Coffeescript

### 八、编程游戏操作框架 - Scratch

#### 1. Scratch介绍

#### 2. Scratch版本选择

# 调研与选型

## 一、前端框架

### 1. 备选的3个框架分析

#### 1.1 Vue

Vue是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。

Vue的两大核心：数据改变驱动了视图的自动更新；视图组件化，可以把整个网页的拆分成一个个区块

适用于小型应用

**特点：**简洁、轻量、快速、数据驱动、模块友好、组件化Vue

## 1.2 React

react是Facebook 开发的框架，用于构建页面、javascript库。主要功能是对DOM操作，声明式设计，更快的开发出web应用系统。

适用于个性化需求、中型应用

**特点：**

1. 声明式设计：React采用声明范式，可以轻松描述应用。
2. 高效：React通过对DOM的模拟，最大限度地减少与DOM的交互。
3. 灵活：React可以与已知的库或框架很好地配合。

## 1.3 Angular

Angular 是一款来自谷歌的开源的 web 前端框架，诞生于 2009 年，由 Misko Hevery 等人创建，后为 Google 所收购。是一款优秀的前端 JS 框架，已经被用于 Google 的多款产品当中。

适用于大型超大型web应用开发

**特点：**模板功能强大丰富、比较完善的前端MVC框架、引入了Java的一些概念优点

## 2. 最终选择 - vue

最终选定Vue作为前端框架，原因如下：

1. 开发的项目为小型web应用，数据源稳定，对于运营的要求不高，但是对加载速度等都有很高的要求。
2. Proxy实现的响应式相比Angular的zone暴力代理和rxjs的复杂操作显得更加接地气，不需要额外地进行学习。对象式的声明在UI实现上速度更快。
3. vue有清晰的中文文档，易于学习上手。

## 二、web端样式库

---

### 1. 备选的8个Web端组件库简介

	简介
<b>Ant Design</b>	一个服务于企业级产品的设计体系，基于确定和自然的设计价值观和模块化的解决方案，让设计者专注于更好的用户体验
<b>Ant Design Pro</b>	一个企业级中后台前端/设计解决方案，秉承 Ant Design 的设计价值观，致力于在设计规范和基础组件的基础上，继续向上构建，提炼出典型模板/业务组件/配套设计资源，进一步提升企业级中后台产品设计研发过程中的用户和设计者的体验。
<b>Iceworks</b>	一套基于 React 的中后台应用解决方案，在阿里巴巴内部，已经有 270 多个来自几乎所有 BU 的项目在使用。Iceworks包含了一条从设计端到开发端的完整链路，帮助用户快速搭建属于自己的中后台应用。
<b>Element</b>	一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库。
<b>SUI</b>	一套基于bootstrap开发的前端组件库，同时它也是一套设计规范。通过SUI，可以非常方便的设计和实现精美的页面。
<b>iView</b>	一套基于 Vue.js 的开源 UI 组件库，主要服务于 PC 界面的中后台产品。
<b>Admui</b>	一个基于最新 Web 技术的企业级通用管理系统快速开发框架，可以帮助企业极大的提高工作效率，节省开发成本，提升品牌形象。可以 Admui 为基础，快速开发各种MIS系统，如CMS、OA、CRM、ERP、POS等。
<b>Zent</b>	有赞 PC 端 WebUI 规范的 React 实现，提供了一整套基础的 UI 组件以及一些常用的业务组件。

## 2. 最终选择 - Ant Design

决定暂时选择**Ant Design**作为项目开发前端样式库。因为Ant Design对Vue有专门的实现，即Ant Design of Vue。且Ant Design提供完善的设计指引、最佳实践、设计资源和设计工具，来帮助设计者快速产出高质量产品。还有Ant Design采用React封装了一套Ant Design的组件库，即Ant Design of React。

## 三、数据库

### 1. 可选方案

MySQL、Oracle、SQL Server

### 2. 纵向比较

#### 2.1 MySQL

MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。

MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，可以降低总体拥有成本。

MySQL可运行在多种平台，如Windows，Unix/Linux；它是专为WEB 数据库设计的，特点是响应速度特别快，主要面向中小企业。

它是真正的多用户多任务的数据库系统，他占用系统资源很少但功能很强大。

#### 2.2 SQL Serve

SQL Serve使用很方便，支持命令和图形化管理，主要结合asp语言开发；SQL Serve是需要购买的部门级数据库。

**缺点：**SQL Serve只能运行在windows平台，体积比较庞大，占用许多系统资源，且难处理日益增多用户数和数据卷伸缩性有限。

## 2.3 Oracle

Oracle能在所有主流平台上运行。完全支持所有的工业标准，采用完全开放策略，可以使客户选择最适合的解决方案。对开发商全力支持。性能最高，保持开放平台下的TPC-D和TPC-C的世界记录。多层次网络计算，支持多种工业标准，可以用ODBC，JDBC，OCI等网络客户连接。长时间的开发经验，完全向下兼容。得到广泛的应用。

**缺点：**Oracle相对需要高额的费用，占用了内存里面太多的内存空间。对硬件的要求很高，管理维护相对麻烦。

## 3. 最终选择 - MySQL

最终选择MySQL，原因如下：

1. MySQL使用成本低，而SQL Server和Oracle成本较为昂贵
2. MySQL跨平台支持，可运行在多个平台，对Linux系统友好

# 四、Python版本

## 1. Python综述

Python由Guido van Rossum于1989年底发明，第一个公开发行业版发行于1991年。Python 是由诸多其他语言发展而来的，其中包括ABC、Modula-3、C、C++、Algol-68、SmallTalk、Unix shell和其他的脚本语言等等。Python是一种解释型、面向对象、动态数据类型的高级程序设计语言，它具有卓越的通用性、高效性、可扩展性、平台移植性和安全性。近几年Python在数据挖掘，人工智能等领域较为流行。

## 2. Python2与Python3的选择

因为早期的Python版本在基础方面设计存在着一些不足之处。2008年的时候Guido van Rossum重新开发了Python 3.0（被称为Python 3000，或简称Py3k），Python3在设计的时候很好地解决了这些遗留问题，并且在性能上也有一定的提升。

然而，Python3带来的最大的问题就是不完全向后兼容，当时向后兼容的版本是Python2.6。Python开发团队无法一下子就让所有项目和类库都转到Python3.0上面。所以，两个版本就进入了长期并行开发和维护的状态。目前，Python2.7被确定为最后一个Python2.x版本，Python3是未来的趋势。

因此，我们决定使用**Python3**来进行相关的开发。

# 五、编辑器

## 1. 可选方案

VScode、PyCharm、Sublime Text、Atom等。

## 2. 最终选择 - Visual Studio Code

我们决定使用Microsoft的**Visual Studio Code**，主要原因如下：

### 1. 轻量级

VScode是一款轻量级的编辑器，安装包非常小，而且启动速度非常快。虽然这对实际的项目没什么大的帮助，但是可以在无形间提高我们的开发体验。

### 2. 丰富的插件系统

VScode有着非常丰富的插件系统，无论你是编辑HTML、CSS、JS、TS、Vue、React等前端代码，还是JAVA、Python等后端代码。我们都可以找到相对应的插件，让我们如虎添翼，更快速地完成项目。

### 3. 代码跟踪功能

我们是一个团队，项目中往往都是协作开发，绝大多数情况下要使用Git来管理我们的代码，这个时候VScode会跟踪我们的代码，非常明显地为我们标注：你更改了那些文件，文件中你修改可第几行的代码，让我们对自己编写的代码一目了然。

## 六、服务端框架

---

### 1. 备选的2种框架分析

#### 1.1 Django

Django是一个由Python编写的具有完整建站能力的开源Web框架。使用Django，只要很少的代码，Python的程序开发人员就可以轻松地完成一个正式网站所需要的大部分内容，并进一步开发出全功能的Web服务。

Django本身基于MVC模型，即Model（模型）+View（视图）+ Controller（控制器）设计模式，因此天然具有MVC的出色基因：开发快捷、部署方便、可重用性高、维护成本低等。Python加Django是快速开发、设计、部署网站的最佳组合。

**特点：**功能完善、要素齐全、完善的文档、强大的数据库访问组件、灵活的URL映射、丰富的Template模板语言、自带免费的后台管理系统、完整的错误信息提示。

#### 1.2 Flask

Flask是一个轻量级的可定制框架，使用Python语言编写，较其他同类型框架更为灵活、轻便、安全且容易上手。它可以很好地结合MVC模式进行开发，开发人员分工合作，小型团队在短时间内就可以完成功能丰富的中小型网站或Web服务的实现。另外，Flask还有很强的定制性，用户可以根据自己的需求来添加相应的功能，在保持核心功能简单的同时实现功能的丰富与扩展，其强大的插件库可以让用户实现个性化的网站定制，开发出功能强大的网站。

**特点：**轻巧、简介、扩展性强、灵活的Jinja2模板引擎提高了前端代码的复用率

### 2. 二者比较

	Django	Flask
整体设计	Django提供一站式的解决方案，从模板、ORM、Session、Authentication等等都分配好了，app, admin后台生成, 使用非常方便.	Flask只提供了一些核心功能，非常简洁优雅。它是一个微框架，其他的由扩展提供，但它的blueprint使它也能够很方便的进行水平扩展。
路由设计	Django的路由设计是采用集中处理的方法，利用正则匹配	更多的是使用装饰器的形式，这个有优点也有缺点，优点是读源码时看到函数就知道怎么用的，缺点是一旦源码比较长，你要查路由就不太方便了，但这也促使你去思考如何更合理的安排代码
配置	Django的配置主要还是靠settings.py来做，当然为了Development和Production环境分离，还有一些方法来处理配置。	Flask的配置很灵活，有多种方法配置，不同环境的配置也非常方便。

### 3. 最终选择 - Django

经过讨论，最终选定**Django**作为我们的服务端框架，原因如下

1. 其中Django的权限管理部分，与 我们要实现的用户群体管理极为相似。
2. Django和Flask，均支持较多扩展，使用简单，但Flask需要手动安装，较为繁琐。Django自带一些模块，能够避免繁琐的安装过程。
3. 在之前的学习和Django实验中，我们成员都对Django有了初步的了解，可以节省学习成本。

## 七、在线编程语言

### 1. 可选方案：

Coffeescript、Javascript..

### 2. 最终选择 - Coffeescript

经过讨论，最终选定**Coffeescript**作为我们的在线编程语言，原因如下

1. 相对于Javascript硬绑了C/Java语法而言，Coffeescript采用了类似Ruby/Python的语法，因此可以用更少代码量去实现传统JavaScript的操作，并且是可使用又易懂的。通过官网(<http://coffeescript.org/>)中给的样例代码来看，Coffeescript的代码量显然更小；
2. 选型时调研了几款编程教育游戏，有CodeMonkey、CodeCombat等游戏都是支持Coffeescript的，原因也是Coffeescript语法更加友好；其中CodeCombat还支持Javascript编程，对比过后感觉Coffeescript代码量小，利于入门者理解编程的精髓。
3. CoffeeScript代码——对应地编译到Javascript，不会在编译过程中进行解释。而且已有的JavaScript类库可以无缝地和CoffeeScript搭配使用，反之亦然。编译后的代码是可读的，且经过美化，能在所有JavaScript环境中运行，并且应该和对应手写的JavaScript一样快或者更快。

## 八、编程游戏操作框架 - Scratch

这一部分指定了使用**Scratch**作为用户操作控制框架。

## 1. Scratch介绍

Scratch是美国麻省理工学院开发的一个可视化的开源编程游戏框架，采用BSD-3-Clause开源协议，开发的前端框架为React。目前市场上大部分网页编程游戏的操作界面基本都是基于该开源框架，在Github上其核心仓库scratch-gui也获得了2.3k的star与1.6k次fork。

该框架主要面向儿童与青少年，所以其游戏方法也较为简单。游戏提供各种行动模块，诸如“移动（）步”、“右转（）度”、“循环（）次”等，用户可以通过类似搭积木的游戏方法通过拖动自由组合这些语句模块从而使目标执行相应的动作，再根据运行的反馈并修改，从而最终完成游戏目标通关游戏。

## 2. Scratch版本选择

此次实验我们选择最新的**Scratch3.0**版本作为基础框架。相对于2.0版本，3.0版本不再使用即将停止维护的flash，而是选择HTML5来编写，它支持跨平台使用，运行也不需要额外的插件。同时相比于2.0版本还增加了诸如字符串包含判断、移至最下层积木等功能。