

1. 前端常用的开发工具及软件（见PPT）

2. 数据库项目中前端的职责以及分工

- 后台管理系统(B端)重视业务逻辑，视觉上简洁、美观即可，适合采用组件库。B端即Business，面向企业，一般在真实开发中，例如商城后台管理系统，会涉及到多个角色的用户且业务逻辑很复杂，而对于页面的要求不高，所以适合采用element-ui/ant-design等组件库进行开发，组件库已为开发者提供了成型的组件，使开发者更专注于业务逻辑的开发。
- 建议采取前后端分离模式（前后端分离模式的三大主流框架Vue/React/Angular），简单理解，前端与后端仅仅通过API文档进行交流，前后端可以并行开发，在前后端分离模式中，前端主要负责页面的设计、搭建、**通过调用各个接口/Mock的方式获取数据以填充自己的页面结构、以及权限验证。**
- 分工方面，建议前端大于后端(6 4/7 3/8 2)

3. HTML

- HTML，即超文本标记语言，何为超文本？即通过**超链接**方法将文本中的信息组织链接起来，这些文本链接起来就组成了万维网的概念：“WWW服务器通过超文本标记语言(HTML)把**信息**组织成为图文并茂的超文本，利用链接从一个站点跳到另个站点。”
- HTML的核心就是一个文档，而浏览器根据文档的内容来解析文档，HTML文档采用使用成对/自结束的语义化标签表示，如下代码所示，例如浏览器在解析文档时，遇到了成对的title标签，那就会将title对标签的内容渲染至该网页的标题上，即文档告诉了浏览器每一行的内容代表了什么意思。

```
<!DOCTYPE html> <!-- 文档声明用来告诉浏览器当前网页的版本，当前版本为HTML5-->
<html lang="en"> <!-- 属性采用名值对结构，lang属性代表网页的语言为英语-->
  <head>
    <meta charset="UTF-8"> <!-- 一些元数据部分，网页采用UTF-8编码-->
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <!-- 移动端适配-->
    <title>My First HTML</title> <!-- 浏览器遇title标签将其内容解析为网页标题-->
    <style></style> <!-- 此处一般放CSS，内部/外部样式表-->
  </head>
  <!--html仅仅关注语义，而不关注样式，不同浏览器帮忙添加了样式-->
  <body>
    <div><span>超链接/列表/表格/表单/图片/音视频/canvas画布...</span></div>
  </body>
</html>
```

- 进一步，有可能你会发现，例如h1 h2 h3等等很多标签内的内容，会被浏览器解析成特殊样式，但这只是浏览器在解析到这个标签时，为该标签加入了特殊的样式（可以理解为写了CSS加上去），所以**HTML仅仅关注语义，而不关注样式。**
- 因为该部分内容较简单，对于更多的语义化标签与HTML知识学习，推荐w3school: <https://www.w3school.com.cn/html/index.asp>网站进行学习。

4. CSS

- CSS，翻译为层叠样式表，层叠的意思是可以为一个标签添加多个样式，而这些样式会叠加（并不是单纯的叠加，也有可能覆盖）。

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>My First HTML+CSS</title>
    <style>
      main{/*先选中*/
        color: pink; /*再设置样式*/
        font-size: 120px;
      }
    </style>
  </head>
  <body>
    <div style="color:red; font-size:60px">我是一个段落</div>
    <main>我是C位</main>
    <div class="main-body" style="color:red; font-size:60px">我是一个段落
  </div>
  </body>
</html>
```

- CSS的设置主要是利用选择器选中元素，再用名值对的形式设置样式，对于CSS的引入，**最佳实践是使用外部样式表**（即将CSS单独写成一个文件并引入），这样方便各个文件之间进行重用，通过缓存机制，可以提升效率：（以下两种方式引入的区别见<https://www.cnblogs.com/passkey/p/10141553.html>）

```
<link rel="stylesheet" href="./reset.css">
@import url(./reset.css)
```

- CSS中一些更高级的话题（加粗的部分可以自行学习下，在当前项目开发中较常用）

复合/关系/属性/伪类/伪元素选择器

样式的继承与选择器的权重

盒模型(※)-标准盒模型/怪异盒模型/块、行内元素

定位(※)-position:relative/absolute/fixed/sticky

flex弹性盒(※※)-水平布局的利器

水平垂直居中的几种方法(※)-开发中常用,定/不定宽高

background背景(※)-设置背景颜色/图以及各种属性

更高级的话题：浮动/文档流相关/BFC/清除浮动/高度塌陷/外边距折叠/transition/animation/媒体查询/**CSS预处理器**(预处理器可以让你用更优雅的写法书写CSS，常见的有less, scss)

- 推荐的CSS学习教程：<https://www.bilibili.com/video/BV1XJ411X7Ud>（尚硅谷CSS，适合零基础入门）

5. JavaScript

- JavaScript主要由三部分构成：ECMAScript提供核心语言功能（即如何定义变量，如何定义函数等等....）、DOM文档对象模型，提供访问和操作网页内容的方法和接口、BOM浏览器对象模型，提供与浏览器交互的方法与接口。
- JavaScript是一个弱类型语言，某一个变量被定义类型,该变量可以根据环境变化自动进行转换。

```
var global = 3; //因为global的值为3, 所以global为Number类型
var global = '1'; //因为global的值为1, 所以global为String类型
//当前, JS共有如下的数据类型:
String/Number/Boolean/Null/Undefined/Object/Symbol/BigInt
```

- 在2015年后的ES6新标准后, 常采用let定义变量, const定义常量, 弥补了用var定义变量的缺点 (有什么缺点可以在学习完ES5基础后再了解), 但在实际使用中, 尽量使用let和const代替var来定义变量。

```
let global = 3; //定义变量
const GLOBAL = 3.1415926; //定义常量, 一般选取大写
```

- 定义函数的写法:

```
let f1 = function(a,b){
    return a+b;
}; //函数表达式
function f2(a,b){
    return a+b;
}; //函数声明
//ES6箭头函数:
let f3 = (a,b) => {
    return a+b;
}
f1();
f2();
f3();
```

- JavaScript中你可以需要知道的语言知识点 (在使用Vue等框架开发时会遇到) :
 - 箭头函数
 - 对象的基本操作,构造函数, ES6中如何定义对象(React现常用)
 - event对象 (监听事件时以参数传入可以获取节点的一些属性)
 - this到底指向谁? 参考文档: <https://www.cnblogs.com/ago-0912/p/10576053.html>, 同时你也可以去了解call, apply, bind函数, 其是用于改变函数内部的this指向时使用的。
 - map/forEach、filter、reduce 方法, 其会在操作数组/对象时常常会用到的高阶方法, 会大幅度提高你的开发效率。
 - 模板字符串
 - promise对象 (ES6中的对象, 解决了回调地狱的问题)
- DOM的增删改查
- BOM浏览器信息, 地址栏信息, 历史记录...
- JS学习教程推荐: 文字MDN (有些枯燥, 但举例全面准确) <https://developer.mozilla.org/zh-CN/>,
- 视频: <https://www.bilibili.com/video/BV1YW411T7GX> (超哥yyds!)

6. 为什么使用Vue框架? 使用原生JS/JQuery去写可不可以?

见demo01原生JS.html和demo02vue.html的对比, MVVM的定义见课件

“解决了用 DOM API 操作 UI 过于反人类的问题。”

- 不用写一大堆操作 DOM 的代码了, 可能提高了工作效率
- 操作 DOM 代码写的不好, 可能会对性能造成影响

- 这些框架让你写模块化的代码更加方便
- 框架流行以后，就有人会在这基础上造轮子，比如 UI 框架、组件通信，方便你更快的实现业务
- 这些框架都有虚拟 DOM 的机制，这层机制可以实现跨端

7. Vue VS React VS Angular如何选择？见课件，概括来说：如果现阶段JS基础不好，建议选Vue.

8. 使用Vue + Element UI开发：这里提供两个demo，其中一个已经下载至资料内，两个demo的仓库的地址为：

- <https://github.com/PanJiaChen/vue-admin-template>：一个极简的 vue admin 管理后台模板。它只包含了 Element UI组件库 & axios发送请求 & iconfont图标 & permission control权限控制 & lint代码格式控制，这些搭建后台管理系统必要的东西，可以读懂它的基础代码后，将一些样式修改成自己想要的，定义自己需要的路由，进而为每个路由分工书写页面。
- <https://github.com/PanJiaChen/vue-element-admin>：功能较复杂、完整的后台管理系统模板，可以作为拓展。
- demo配套了完善的**使用说明与教程**，对新手较为友好：使用说明<https://panjiachen.github.io/vue-element-admin-site/zh/>，教程<https://juejin.cn/post/6844903476661583880>
- Vue学习教程推荐：1、官方文档<https://cn.vuejs.org/index.html>
- 2、<https://www.bilibili.com/video/BV15741177Eh>（B站播放量最高教程，讲解详细，有些过时内容基本可忽略）

9. 组件化开发与组件库

组件化，你怎么写代码我不管，你的功能全在这你这儿实现（内聚性），不要让我还帮写你那块功能。另外，你的代码不要影响我的代码（低耦合性）。

即如果将一个页面中所有的逻辑全部放在一起，处理起来会变得非常复杂，且极其不好分工，所以我们将页面拆成小的组件，每个组件有自己的html+css+js，管理比较容易。

页面只不过是组件的容器，组合组件即可形成功能完整的界面

10. 认识Vue-CLI

vue-cli 是**vue.js的脚手架**，用于**自动生成vue.js+webpack**的项目模板，你可以理解类似于PPT模板，你在这些模板内填写内容，你也可以自己搭建脚手架，但你考虑的因素（热部署,单元测试等等...）是否全面是一个重要的问题。

11. 什么是Vue Router？前端路由与后端路由的区别是什么？（见PPT）

12. 什么是Vuex？用处是什么？（见PPT）

13. 项目中如何发送请求至后端？（代码见demo02）

ajax: Ajax技术网页应用能够快速地将增量更新呈现在**用户界面**上，而不需要重载（刷新）整个页面，这使得程序能够更快地回应用户的操作

axios: 用于浏览器和node.js的**基于Promise**（解决回调地狱问题）的HTTP客户端

14. 如何验证用户的登录状态及鉴权？(cookie, session, token的基本认知与使用)

- 对于前端来说，你的系统的一些页面的访问与操作只能在登录状态下进行，而且对于多角色的系统，不同角色能够访问页面的权限与**侧边路由（后台管理系统）**不同；
- 对于后端来说，需要验证每次访问接口的权限（验证是否过期）以及角色信息。

比较常见的两种机制：

1、 cookie+session

cookie是存在浏览器本地的信息，而session是存储在服务器的信息，第一次登录成功后，服务器将存储此次登录的用户信息（SESSIONID + USERID + ROLE），然后在这次登录的响应头中添加 SetCookie:SESSIONID:.....，浏览器接收到此信息时，会将其存储在cookie中。在接下来的接口访问中，会在请求中携带cookie，cookie中最重要的就是SESSIONID,服务器接到请求后，服务器将去存储用户信息的表中（一般采用Redis内存数据库）查询SessionID是否有效，以及相应的角色信息，用于判断该用户是否可以访问该接口。

但这种机制存在的问题主要在分布式的后台中，最大的弊端就是SESSION存储在服务器中，如果有A,B,C三台服务器构成了分布式的后端，第一次登录用户访问了A服务器，SESSION存储在A服务器，第二次用户的请求被分配到了B服务器，此时无法访问到上次存储在A服务器中的SESSION了，如果采用SESSION STICKY(让用户的请求粘在服务器上) 集中式存储SESSION，也会出现效率不高、单点故障等问题。

2、 cookie/localStorage+token

token主要是以时间换空间的思想，第一次登陆成功时，服务器会利用自身密钥与用户信息以及加密算法等信息共同计算出一段字符串：（x.y.z的格式）

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjMONTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0Ij0iOnRydWV9.TjVA95OrM7E2cBab30RMhRHDcEfxjoYZgeFONFh7HgQ
```

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9（第一个点之前的部分）使用base64编码，解码后的内容是：

```
{"alg": "HS256", "typ": "JWT"}
```

eyJzdWIiOiIxMjMONTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0Ij0iOnRydWV9(2-3点之间的部分)

```
{官方字段()+自定义字段"sub": "1234567890", "name": "John Doe", "admin": true}
iss (issuer): 签发人
exp (expiration time): 过期时间
sub (subject): 主题
aud (audience): 受众
nbf (Not Before): 生效时间
iat (Issued At): 签发时间
jti (JWT ID): 编号
```

即你会发现，x与y在一定程度上可以说是明文的，而真正起验证与保护作用的是z，即 $z = f(x, y, \text{服务器唯一密钥})$ ，任何一个部分发生篡改，这个等式不可能成立，保护了服务器。

例如我想将第二个部分的role信息修改为更高权限，role: 0，变为role:1,然后组合其他部分信息发送至服务器，服务器拿到token解析的过程如下：与重新生成签名时一样， $z' = f(x, y, \text{服务器唯一密钥})$ ，检查 $z' == z$ ？如果相等，则验证这个token没有被篡改，且拿到了里面的用户信息，进而去查找用户是否能访问该接口。

15. 跨域问题的出现及解决

- 例如你当前处于<http://47.120.45.21:8080> 获取了网页，你现在在网页中发送了一个请求，这份请求的访问地址（URL）是<http://47.120.45.21:3000> 或者 另一台服务器（这样的情况比较常见，因为前后端分离，所以通常前后端部署在不同的服务器上）<http://120.45.25.138:3000>，只要网页的URL与请求的URL中协议（http/https），域名，端口，有一个不同，**就说明这两个网页不是同一个域下的网页，那这份请求会发送出去，服务器也会响应对应的结果，但是浏览器会在请求返回时拦截，发生请求报错。**跨域不能请求的问题是浏览器处于安全机制的考虑而产生的。
- **主流的解决方案：**<https://panjiachen.github.io/vue-element-admin-site/zh/guide/advanced/cors.html>网站讲解的很精准与到位，现主流的解决就是后端配置CORS（后端加一个配置文件即可，即在响应头中加入一个字段，前端无工作量，推荐数据库课设项目中使用），还有就是前端配置代理（前端增加工作量，不建议）

16. 如何上传图片？我上传的图片应该在哪里存储？

建议图片存储在阿里云OSS云存储服务器上，这是一个适合专门存储图片、视频等数据的服务，在任何时间、任何地点、任何互联网设备上上传和下载数据。基于OSS，您可以搭建出各种多媒体分享网站、网盘、个人和企业数据备份等基于大规模数据的服务。

- 建议后端提供上传图片接口（后端调用阿里云OSS，以返回图片地址）

17. 项目应该如何打包以及部署？

- **首先注意一个问题：前端不一定非要部署，而后端一般必须要部署。**
- 前端部署教程：<https://www.cnblogs.com/mark21/p/14118693.html>