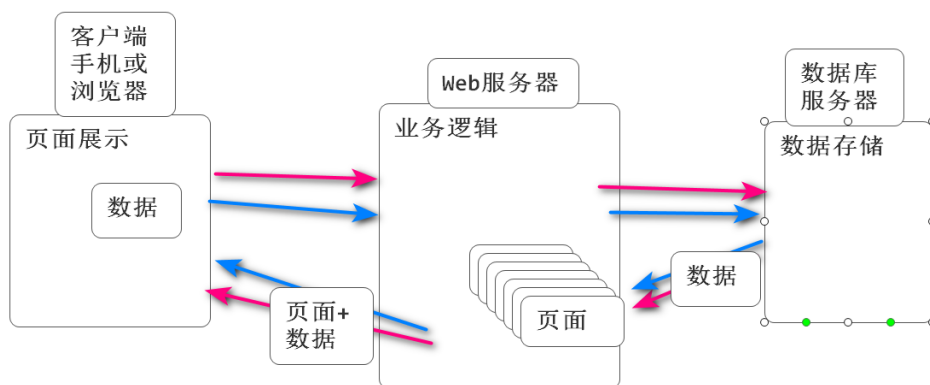


副本 2202Web前端笔记

自我介绍：刘国斌

第三阶段课程介绍：

- web前端
- 数据库
- SpringBoot



Web前端-HTML

- HTML作用：负责搭建页面结构和内容（盖房子）
- 学习HTML主要学习的就是有哪些标签

文本相关标签

- h1-h6 文本标题，特点：独占一行，自带上下间距，字体加粗
- p 段落标签，特点：独占一行，自带上下间距
- hr 水平分割线
- br 换行
- b 加粗
- i 斜体
- u 下划线
- s 删除线

列表标签

- 无序列表：ul和li 组合 unordered list list item 列表项

- 有序列表: ol和li 组合 ordered list
- 列表嵌套: 有序列表和无序列表可以任意无限嵌套 .

图片标签img

- src: 资源路径
 - 相对路径: 访问站内资源
 - 图片和页面在同级目录: 直接写图片名
 - 图片在页面的上级目录: ../图片名
 - 图片在页面的下级目录: 文件夹名/图片名
 - 绝对路径: 访问站外资源, 图片盗链, 有找不到图片的风险
- alt: 图片不能正常显示时显示的文本
- title: 图片标题
- width/height: 设置宽高 , 只设置宽度时高度会自动等比例缩放
 - 像素
 - 上级元素百分比

超链接a

- href: 资源路径, 作用类似图片标签的src
- a标签包裹文本为文本超链接, 包裹图片为图片超链接
- 页面内部跳转, 在目的地的元素里面添加id=xxx 然后在超链接里面添加href="#xxx"

表格table

- 相关标签: table tr表示行 td表示列 th表头 caption标题
- 相关属性: border边框 colspan跨列 rowspan跨行

表单form

- 作用: 获取用户输入的各种信息 并提交给服务器
- 学习form表单主要学习的就是有哪些控件

```

1 <form action="http://www.baidu.com">
2     <!--maxlength最大字符长度
3     value设置默认值
4     readonly只读-->
5     用户名:<input type="text" name="username" maxlength="5"
6         value="abc" readonly ><br>
7     密码:<input type="password" name="password"><br>
8     <!--value是单选框必须添加的属性 否则提交on
9     checked默认选中-->
10    性别:<input type="radio" name="gender" value="m" id="r1">
11    <label for="r1">男</label>
12    <input type="radio" name="gender" checked value="w">女<br>
13    兴趣爱好:<input type="checkbox" name="hobby" value="cy">抽烟
14    <input type="checkbox" name="hobby" checked value="hj">喝酒
15    <input type="checkbox" name="hobby" value="tt">烫头<br>

```

```

16     生日:<input type="date" name="birthday"><br>
17     靓照:<input type="file" name="pic"><br>
18     所在地:
19     <!--value设置提交的内容      selected默认选中-->
20     <select name="city">
21         <option value="bj">北京</option>
22         <option value="sh">上海</option>
23         <option value="gz" selected>广州</option>
24     </select><br>
25     <input type="submit" value="注册">
26 </form>

```

分区标签

- 作用: 可以理解为是一个容器,将多个有相关性的标签进行统一管理
- 块级分区标签div: 特点是独占一行
- 行内分区标签span: 特点是共占一行
- 页面区域如何划分? 至少分为三大区(头,体,脚) 每个大的区域再划分n个小的区域
- HTML5的标准中新增了一些语义更强的分区标签,为了提高代码的可读性. 这几个标签的作用和div一样都是块级分区标签
 - header头
 - footer脚
 - main主体
 - section区域
 - nav 导航

CSS 层叠样式表

- Cascading Style Sheet
- 作用: 美化页面(装修)

如何在HTML页面中添加CSS样式代码

- 三种引入方式:
 - 内联样式: 在标签的style属性中添加样式代码, 弊端:不能复用
 - 内部样式: 在head标签里面添加一个style标签, 在标签体内写样式代码, 可以实现复用但是只能本页面复用
 - 外部样式: 在单独的css样式文件中写样式代码, 在html页面中通过link标签引入, 可以实现多页面复用, 可以将html代码和css样式代码分离

选择器

- 作用: 通过选择器找到页面中的元素之后再添加样式

1. 标签名选择器: 选取页面中所有同名标签

- 格式: 标签名{样式代码}
- 2. id选择器: 当需要选择页面中某一个元素时使用
 - 格式: #id{样式代码}
- 3. 类选择器: 当需要选择多个不相关的元素时,给多个元素添加相同的class属性,然后通过类选择器进行选择
 - 格式: .class{样式代码}
- 4. 分组选择器: 分组选择器可以将多个选择器合并成一个
 - 格式: h4,#id,.class{样式代码}
- 5. 属性选择器: 通过元素的属性选择元素
 - 格式: 标签名[属性名="属性值"]{样式代码}
- 6. 任意元素选择器: 选择页面中所有标签
 - 格式: *{样式代码}

选择器练习:

1. 把张学友改成绿色
2. 把刘德华和悟空改成蓝色
3. 修改取经3人和刘备的背景为黄色
4. 修改密码框的背景为红色
5. 给所有元素添加红色的边框

工程的git地址:

<https://gitee.com/teduliu/cgb2202.git>

复制整行快捷键 ctrl+D

注释快捷键 ctrl+shift+/
/

快速进入下一行 shift+回车

day02

选择器续

1. 子孙后代选择器: 通过元素之间的层级关系选择元素
 - 格式: body div div p{样式代码} 匹配body里面的div里面的div里面的所有p标签(包括后代)
2. 子元素选择器: 通过元素之间的层级关系选择元素
 - 格式: body>div>div>p{样式代码} 匹配body里面的div里面的div里面的所有p子元素(不包含后代)

3. 伪类选择器: 选择的是元素的状态, 元素有哪些状态? 包括: 未访问状态, 访问过状态, 悬停状态, 点击状态
- 格式: `a:link/visited/hover/active:{样式代码}`

选择器练习:

1. 关羽绿色
2. 张飞和苹果黄色
3. 文本框和所有水果背景红色
4. p2字体粉色
5. p2和p3背景黄色
6. 优酷未访问绿色, 访问过红色
悬停黄色, 点击粉色

颜色赋值

三原色: 红绿蓝 , red green blue rgb , 每个颜色的取值范围0-255

颜色赋值的几种方式:

- 颜色单词: 常见颜色单词都可以使用
- 6位16进制赋值: #ff0000
- 3位16进制赋值: #f00
- 3位10进制赋值: rgb(255,0,0)
- 4位10进制赋值: rgba(255,0,0,0-1) a=alpha代表透明度

背景图片

- background-image:url("路径") 设置背景图片
- background-size:100px 200px 设置背景图片尺寸
- background-repeat:no-repeat; 禁止重复
- background-position: 横向 纵向; 设置背景图片的位置, 两种方式: 1像素 2百分比

文本和字体相关样式

- text-align:left/right/center; 文本水平对齐方式
- line-height:20px; 设置行高, 多行文本时可以控制行间距, 单行文本时可以控制垂直居中(因为文本默认是在当前所在行内居中)
- text-decoration:overline上划线/underline下划线/line-through删除线/none去掉文本修饰
- text-shadow:颜色 x偏移值 y偏移值 浓度; 阴影
- font-size:20px; 设置字体大小
- font-weight:bold加粗/normal去掉加粗
- font-style:italic; 设置斜体
- font-family: xxx,xxx,xxx; 设置字体

- font:20px xxx,xxx,xxx; 这只字体大小+字体

文本和字体相关练习

1. 刘德华 宽度100 高度30
绿色背景 红色字体
横向和纵向居中
2. 苹果和香蕉 字体大小25px 斜体
3. 冰箱去掉加粗 蓝色阴影
方向是左下 浓度3
4. 洗衣机添加下划线
5. 百度去掉下划线字体加粗
字体大小20px

元素的显示方式display

- block: 块级元素的默认值, 特点: 独占一行 可以修改宽高, 包括: h1-h6, p, div
- inline: 行内元素的默认值, 特点: 共占一行 不可以修改宽高, 包括: span, b,i,s,u,超链接a
- inline-block: 行内块元素默认值, 特点: 共占一行 并且可以修改宽高, 包括: img,input
- none: 隐藏元素
- 行内元素不能直接修改宽高, 如必须修改则先将元素的显示方式改成 block或inline-block

盒子模型

- 盒子模型用来控制元素的显示效果包括: 元素内容content+外边距margin+边框border+内边距padding
- 元素内容content: 控制元素的显示尺寸
- 外边距margin: 控制元素的显示位置
- 边框border: 控制边框效果
- 内边距padding: 控制元素内容的位置

盒子模型之内容content

- 包括: width和height
- 赋值方式有两种:
 - 像素
 - 上级元素的百分比
- 行内元素不能直接修改宽高

盒子模型之外边距margin

- 作用: 控制元素的显示位置

- 赋值方式:
 - margin-left/right/top/bottom:10px; 单独某个方向赋值
 - margin:10px; 四个方向赋值
 - margin:10px 20px; 上下10 左右20
 - margin:10px 20px 30px 40px; 上右下左 顺时针赋值
- 行内元素上下外边距无效
- 上下相邻彼此添加外边距 取最大值
- 左右相邻彼此添加外边距 两者相加
- 粘连问题: 当元素的上边缘和上级元素的上边缘重叠时,给元素添加上外边距会出现粘连问题,给上级元素添加overflow:hidden解决

盒子模型之边框border

- 赋值方式:
 - border:1px solid red; 给四个方向添加边框
 - border-left/right/top/bottom:1px solid red; 单独给某个方向添加边框
- border-radius:10px; 值越大越圆 当值超过宽高的一半时为正圆(前提是正方形)

盒子模型之内边距padding

- 作用: 控制元素内容的位置
- 赋值方式: 和外边距类似
 - padding-left/right/top/bottom:10px; 单独某个方向赋值
 - padding:10px; 四个方向赋值
 - padding:10px 20px; 上下和 左右赋值
 - padding:10px 20px 30px 40px; 上右下左顺时针赋值
- 给元素添加内边距会影响元素的显示宽高

CSS的三大特性

- 继承: 元素可以继承上级元素文本和字体相关的样式,部分标签自带的效果不受继承影响, 比如超链接字体颜色
- 层叠: 多个选择器可能选择到同一个元素,如果添加的样式不同则全部层叠有效,如果作用的样式相同 则由优先级决定哪个生效
- 优先级: 指CSS中的选择器具有优先级, 作用范围越小优先级越高, !important>id>class>标签名>继承(属于间接选中)

day03

元素的定位方式

- 五种定位方式:
 - 静态定位: position:static;

- 相对定位: position:relative;
- 绝对定位: position:absolute;
- 固定定位: position: fixed;
- 浮动定位: float:left/right

静态定位(文档流定位)

- 格式: position:static; (默认的定位方式)
- 特点: 元素以左上为基准, 块级元素从上往下排列,行内元素从左向右依次排列, 默认情况下 无法实现元素的层叠效果
- 如何控制元素的位置?

通过外边距控制元素的位置

相对定位

- 格式: position:relative;
- 特点: 元素不脱离文档流(仍然占着原来的位置) ,可以实现元素的层叠效果
- 如何控制元素的位置?

通过left/right/top/bottom 控制元素的显示位置,参照物是初始位置

- 应用场景: 当希望移动某一个元素其它元素不受影响时使用相对定位, 相对定位可以实现元素的层叠

绝对定位

- 格式: position:absolute;
- 特点: 元素脱离文档流(不占原来的位置),可以实现元素层叠
- 如何控制元素的位置?

通过left/right/top/bottom控制位置,参照物为窗口(默认)或某一个上级元素(需要在上级元素中添加position:relative作为参照物)

- 应用场景: 当需要往页面中添加一个元素并且不影响其它元素的显示位置,并且可以实现层叠效果

固定定位

- 格式: position:fixed;
- 特点: 脱离文档流
- 如何控制元素的位置?

通过left/right/top/bottom相对于窗口做偏移

- 应用场景: 当需要将元素固定在窗口的某个位置时使用

浮动定位

- 格式: float:left/right;
- 特点: 脱离文档流, 元素从当前所在行向左或向右浮动,当撞到上级元素边缘或其它浮动元素时停止.

- 浮动元素一行装不下时会自动折行, 折行时有可能被卡住
- 当元素的所有子元素全部浮动时,自动识别的高度会为0,后面元素会顶上来导致显示异常,通过给元素添加overflow:hidden可以解决此问题
- 应用场景: 当需要将纵向排列的元素改成横向排列时使用.

溢出设置overflow

- visible超出部分显示(默认)
- hidden超出部分隐藏
- scroll 超出部分滚动显示

行内元素垂直对齐方式vertical-align

- top 上对齐
- middle中间对齐
- bottom下对齐
- baseline基线对齐(默认)

元素的显示层级z-index

当两个元素非静态定位时可能存在层叠的问题 通过z-index样式控制显示层级, 值越大显示越靠前

day04

JavaScript

- 作用: 负责给页面添加动态效果
- 语言特点:
 - 属于脚本语言(不需要编译直接由浏览器解析执行)
 - 基于面向对象
 - 属于弱类型语言

java: String name="tom"; int age = 18; int x; x="abc";报错
 JS: let name="tom"; let age = 18; let x; x="abc"; age="xyz"
 - 剪安全性强: JS语言只能访问浏览器内部的数据,浏览器以外电脑上的数据禁止访问.
 - 交互性强: 因为JS语言是嵌入到html页面中最终执行在客户端的语言 可以和用户直接进行交互, 而像Java语言是运行在服务器的语言和用户交互需要借助于网络,所有交互性JS语言会更强一些

如何在HTML页面中添加JS语言

- 三种引入方式:
 - 内联: 在标签的事件属性中添加js代码,当事件触发时执行.
 - 事件: 是系统提供的一系列时间点.

- 点击事件: 当点击元素时触发的时间点
- 内部: 在html页面中的任意位置(推荐写在/body上面)添加script标签, 标签体内写js代码
- 外部: 在单独的js文件中写js代码, 然后在html页面中通过script标签的src属性引入到html页面

变量, 数据类型, 运算符, 各种语句, 方法, 面向对象

变量

- JS属于弱类型语言

java: String name="tom"; int age = 18; int x; x="abc";报错

JS: let name="tom"; let age = 18; let x; x="abc"; age="xyz"

- let和var关键字的区别, 作用域有区别
- 使用let声明的变量, 作用域和Java语言的作用域类似
- 使用var声明的变量, 不管在什么位置声明 都相当于是全局变量

```

1  java:
2  for(int i=0;i<10;i++){
3      int y = i+1;
4  }
5  int z = i+y; 报错 i和y超出了自身的作用域
6  JS:
7  for(let i=0;i<10;i++){
8      let y = i+1;
9  }
10 let z = i+y; 不报错,但是访问不到i和y 因为超出了作用域
11 for(var i=0;i<10;i++){
12     var y = i+1;
13 }
14 var z = i+y; 不仅不报错还能够访问到i和y的值

```

数据类型

- JavaScript中只有对象类型
- 常见的对象类型包括:
 - 字符串: string 可以用单引号或双引号修饰
 - 数值: number 相当于Java中所有数值类型的总和
 - 布尔值: boolean true和false
 - 未定义: undefined 当变量只声明不赋值时属于未定义
- 获取变量类型的方法 typeof 变量;

运算符

- 算数运算符: + - * / % , JS中的除法会根据结果自动转换整数或小数

- Java: `int x =5 ; int y = 2; x/y = 2;`
- JS : `let x =5; let y = 2; x/y =2.5; x=6 x/y=3;`
- 关系运算符: `> < >= <= != ==和===`
 - `==`: 先统一两个变量的类型 再比较值
`"666"==666 true`
 - `===`:先比较类型,类型相同后再比较值
`"666"===666 false`
- 逻辑运算符: `&& || !`
- 赋值运算符: `= += -= *= /= %= ++ --`
- 三目运算符: `条件?值1:值2`

各种语句:

- `if else`
- `for`
- `while`
- `do while`
- `switch case`

方法

- java: `public 返回值类型 方法名(参数列表){方法体}`
- JS: `function 方法名(参数列表){方法体}`
- 常见的四种方法:
 - 无参无返回值
 - 有参无返回值
 - 无参有返回值
 - 有参有返回值
- 三种声明方法的格式:
 - `function 方法名(参数列表){方法体}`
 - `let 方法名 = function (参数列表){方法体}`
 - `let 方法名 = new Function("参数1","参数2","方法体");`

和页面相关的方法

1. 通过选择器获取页面中的元素对象

`let 元素对象 = document.querySelector("选择器")`

2. 获取和修改元素的文本内容

`元素对象.innerText = "xxx";` 修改文本内容

`元素对象.innerText` 获取文本内容

3. 获取和修改input控件的值

`控件对象.value="xxx";` 修改

`控件对象.value` 获取

NaN

- Not a Number: 不是一个数
- isNaN(x) 判断变量是否是NaN

JavaScript对象分类

- 内置对象:包括string,number,boolean等
- BOM: Browser Object Model, 浏览器对象模型, 包括和浏览器相关的内容
- DOM: Document Object Model, 文档对象模型,包括和页面标签相关的内容

BOM浏览器对象模型

- window: 此对象里面的属性和方法称为全局的属性和方法,访问时可以省略掉window.
- window中常见的方法:
 - alert("xxx")弹出提示框
 - confirm("xxx") 弹出确认框
 - prompt("xxx") 弹出文本框
 - isNaN(x) 判断变量是否是NaN
 - parseInt()和parseFloat() 把字符串转成整数或小数
 - console.log() 控制台输出
 - let timer = setInterval(方法,时间间隔) 开启定时器
 - clearInterval(timer) 停止定时器
 - setTimeout(方法,时间间隔) 开启只执行一次的定时器
- window对象中常见的属性
 - location位置
 - location.href 获取和修改浏览器的请求地址
 - location.reload() 刷新页面
 - history历史
 - history.length 获取历史页面数量
 - history.back() 返回上一页面
 - history.forward() 前往下一页面
 - history.go(n) n=1是前往下1页面 n=-1 返回上一页面 n=2 前往下2个页面 n=0代表刷新

day05

DOM文档对象模型

- 包含和页面元素相关的内容

1. 通过选择器获取页面中的元素对象

let 元素对象 = document.querySelector("选择器")

2. 获取和修改元素的文本内容

元素对象.innerText = "xxx"; 修改文本内容

元素对象.innerText 获取文本内容

3. 获取和修改input控件的值

控件对象.value="xxx"; 修改

控件对象.value 获取

4. 创建元素对象

let 元素对象 = document.createElement("标签名");

5. 添加元素对象到某个元素里面

document.body.appendChild(元素对象);

父元素.appendChild(元素对象);

父元素.append(元素对象,元素对象,元素对象);

前端MVC设计模式

- MVC设计模式,其实就是将前端实现某个业务的所有代码划分为三部分
- Model: 模型, 指数据模型,这个数据一般来自于服务器
- View: 视图, 指页面标签内容
- Controller:控制器, 指将数据展示到视图中的过程代码
- 前端MVC设计模式弊端: 需要在Controller控制器部分频繁进行DOM(遍历查找元素的代码)操作, 会影响执行效率.

M,V,VM设计模式

- Model: 模型, 指数据模型,这个数据一般来自于服务器
- View: 视图, 指页面标签内容
- ViewModel:视图模型, 将页面中可能发生改变的元素和某个变量在内存中进行绑定,当变量的值发生改变时会从内存中直接找到对应的元素让其改变.

Vue

- Vue框架是目前最流行的前端框架
- Vue对象相当于MVVM设计模式中的VM, 负责将页面中可能发生改变的元素和变量进行绑定, 绑定完之后会不断监听变量的改变, 当变量的值发生改变时会自动找到对应的元素并改变其显示内容.
- 相关指令:
 - {{变量}}:插值, 让此处的文本内容和变量进行绑定

- v-text="变量", 让元素的文本内容和变量进行绑定
- v-html="变量", 让元素的html内容和变量进行绑定
- v-bind:属性名="变量", 让元素的xxx属性和变量进行绑定 可以省略掉v-bind
- v-model="变量", 双向绑定, 当变量的值发生改变页面会跟着改变, 页面的内容改变时变量也会跟着改变,只有使用form表单中的控件时才会涉及到双向绑定.
- v-on:事件名="方法", 事件绑定 简写@事件名="方法", 调用的方法必须声明在Vue里面的methods里面

day06

Vue指令(续)

- v-for="(变量,i) in 数组"; 遍历数组 同时重复生成当前标签,数量和数组中对象的数量一致
- v-if="变量", 变量值为true时显示元素 ,为false时 删除元素
- v-else 和v-if结合使用 取反
- v-show="变量", 变量值为true时显示元素 ,为false时 隐藏元素(需要频繁切换显示状态时使用)

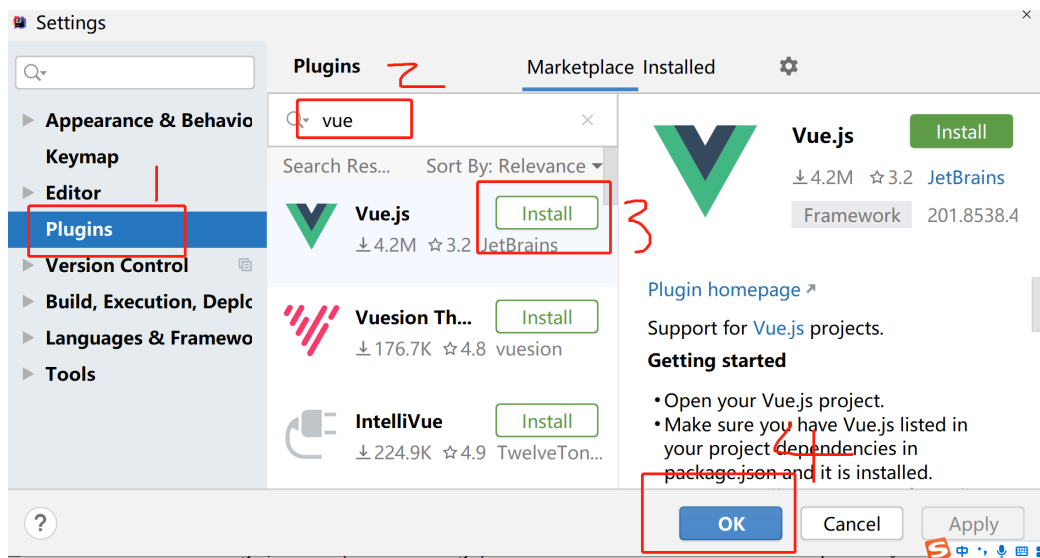
ElementUI

- 官网地址: <https://element.eleme.cn>

<http://localhost:63342/2202CGB/Web/day06/eui/css/fonts/element-icons.woff>

Idea中安装Vue插件

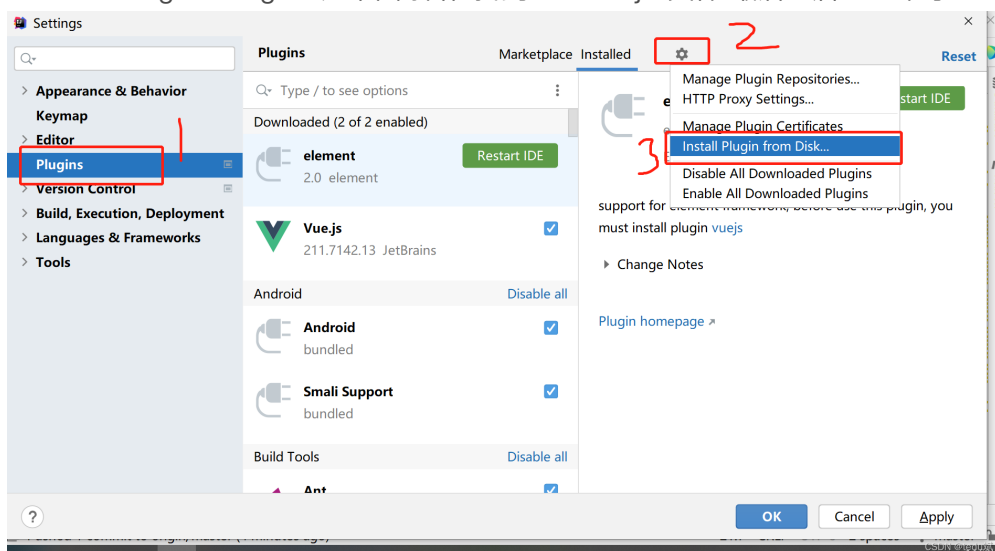
File->Settings->Plugins 搜索Vue



day07

在idea中安装elementUI代码提示插件

- 先从老师工程中得到element.jar 文件,保存到磁盘中的某个位置(容易找到的位置)
- File->Settings->Plugins 如下图 操作 找到element.jar文件 最后重启idea即可



数据库笔记地址:

<https://shimo.im/docs/NJkbEwl9LouJZ5qR/>