

文档说明

2019-9-8 21:37:46

目录

文档说明 1

课程安排 2

基础知识 2

 HTML 教程 2

 Div-css 2

 练习 2

 JavaScript 讲解 2

 练习 3

 Chrome 3

 练习 3

Webstorm 3

 主题外观 3

 练习 4

 字体 4

 练习 6

 安装 vue 插件 6

 练习 7

Vue 7

 介绍 8

 Vue-视频 8

 渐进式 8

 官网 8

 工具 9

 自学，了解 9

 下载 9

 代码实操 9

练习	11
扩展	11
回顾	12
课后	12

课程安排

2019-9-8 20:40:51

- 邮箱: 2669553551@qq.com
- 创建一个 qq 群, 发文件
- 负责人
- 可以自带电脑

基础知识

HTML 教程

<https://www.w3school.com.cn/index.html>

Div-css

https://www.w3school.com.cn/tags/tag_div.asp

练习

JavaScript 讲解

<https://www.w3school.com.cn/js/index.asp>

练习

Chrome

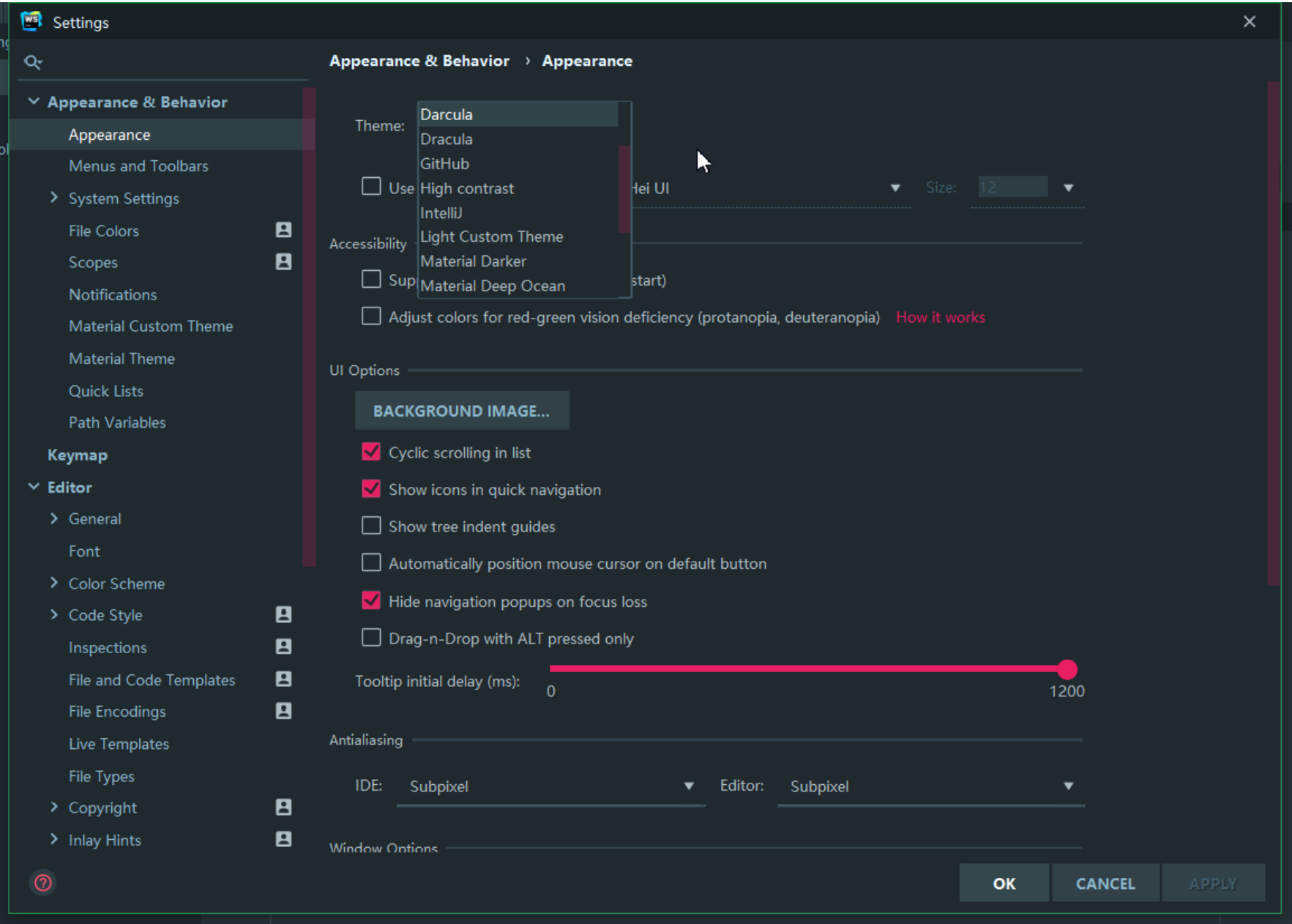
开发者模式，调试，控制台，注入，爬虫，网络（http）

练习

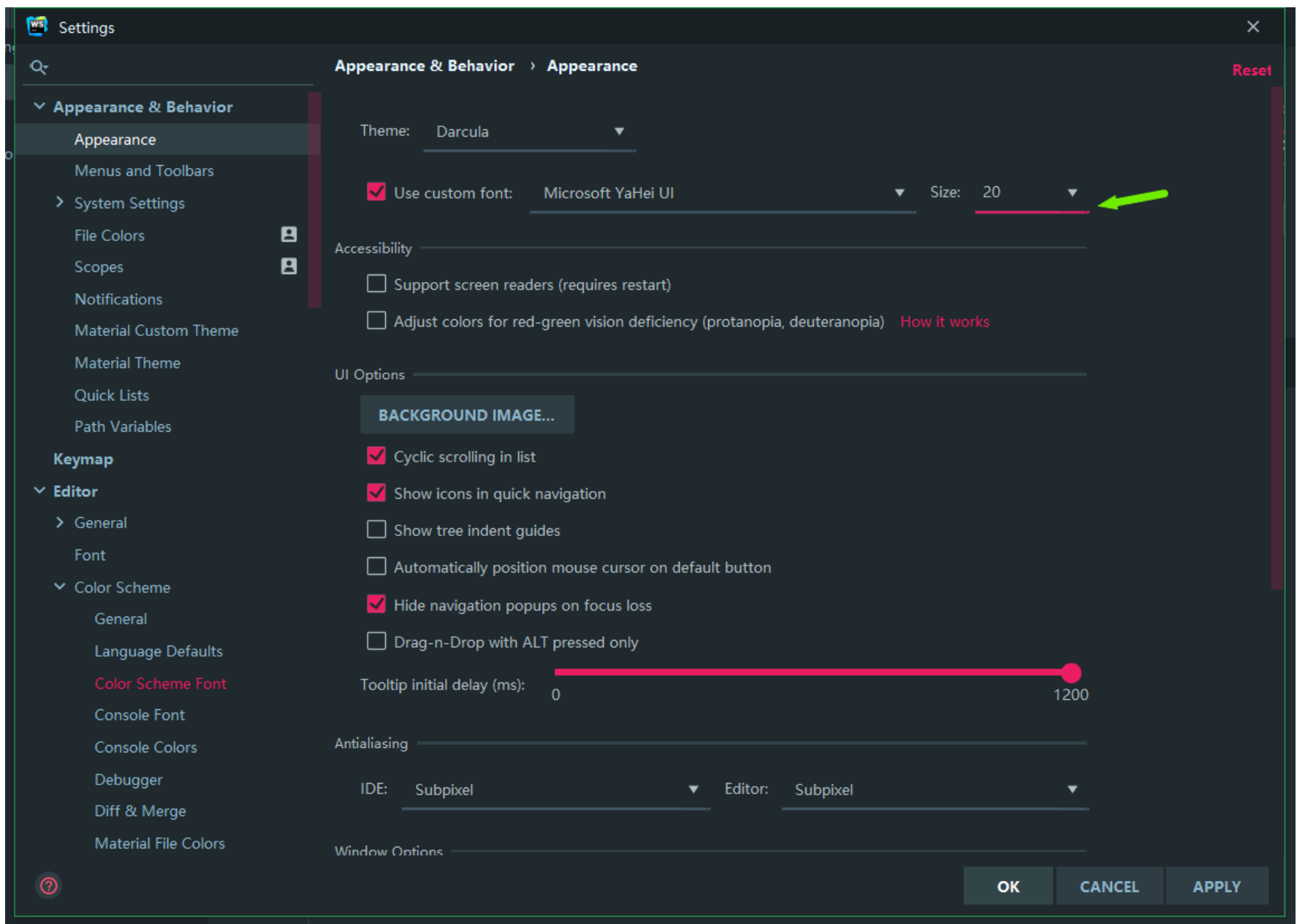
Webstorm

主题外观

选择不同的主题



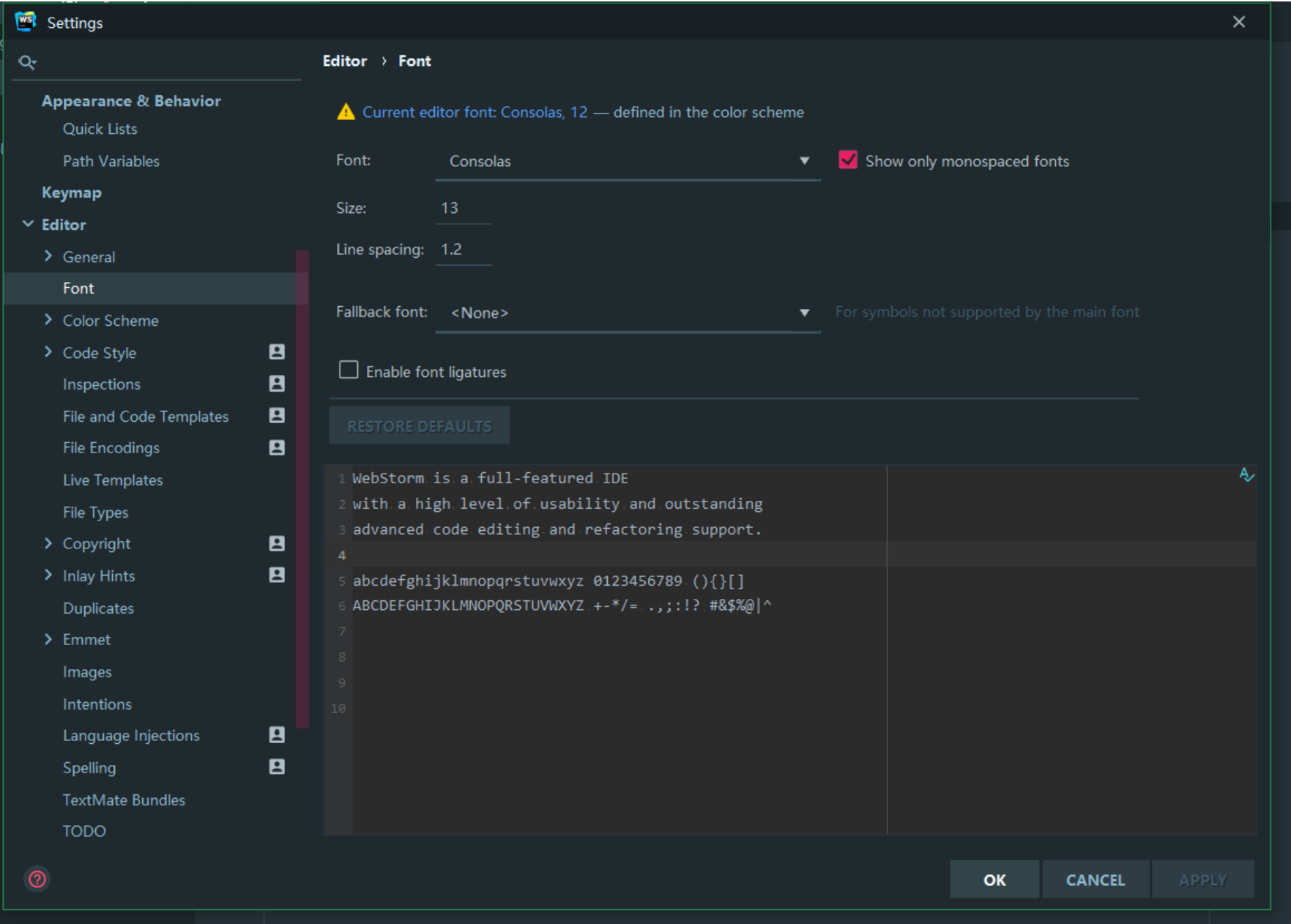
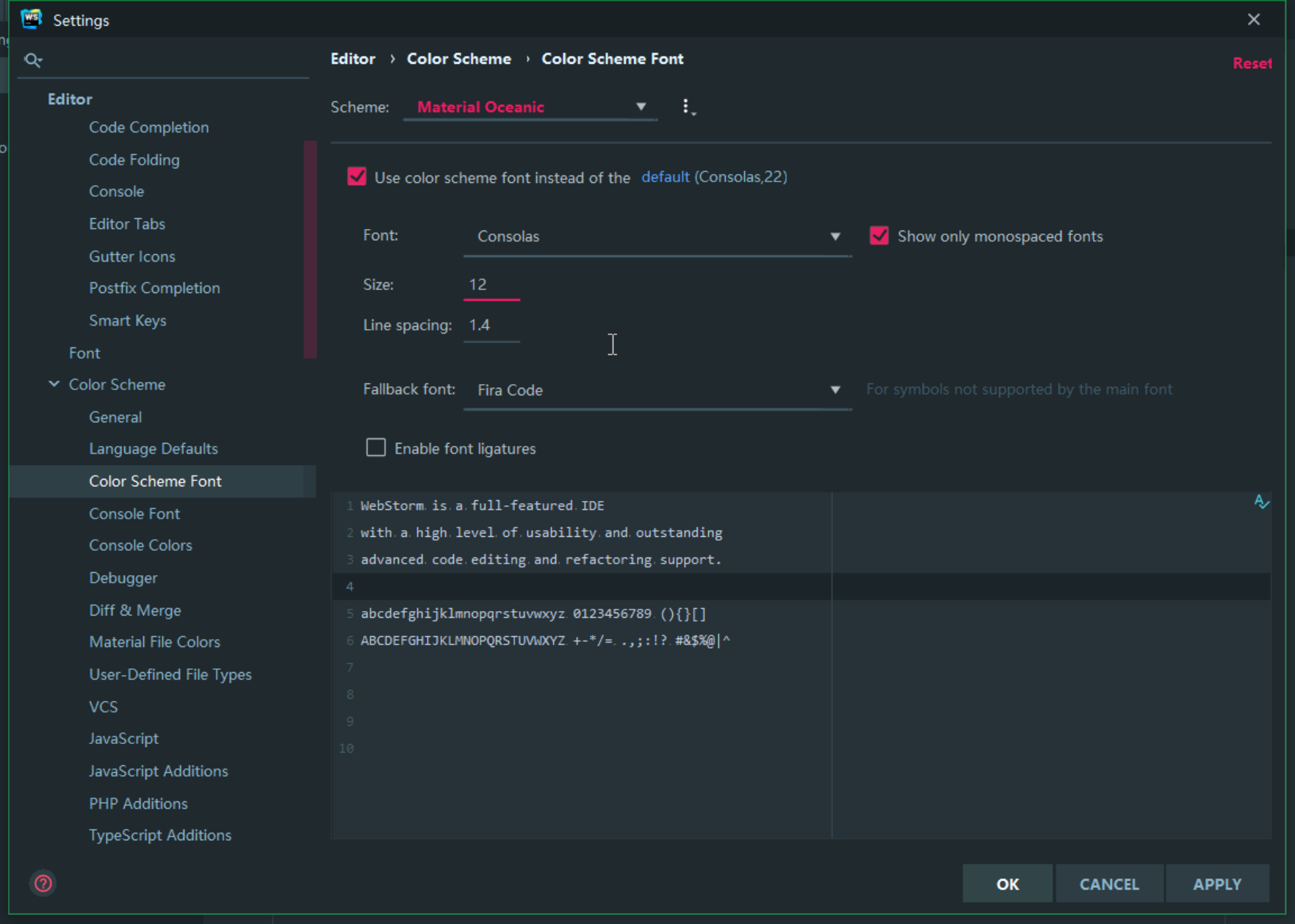
改变主题字体的大小



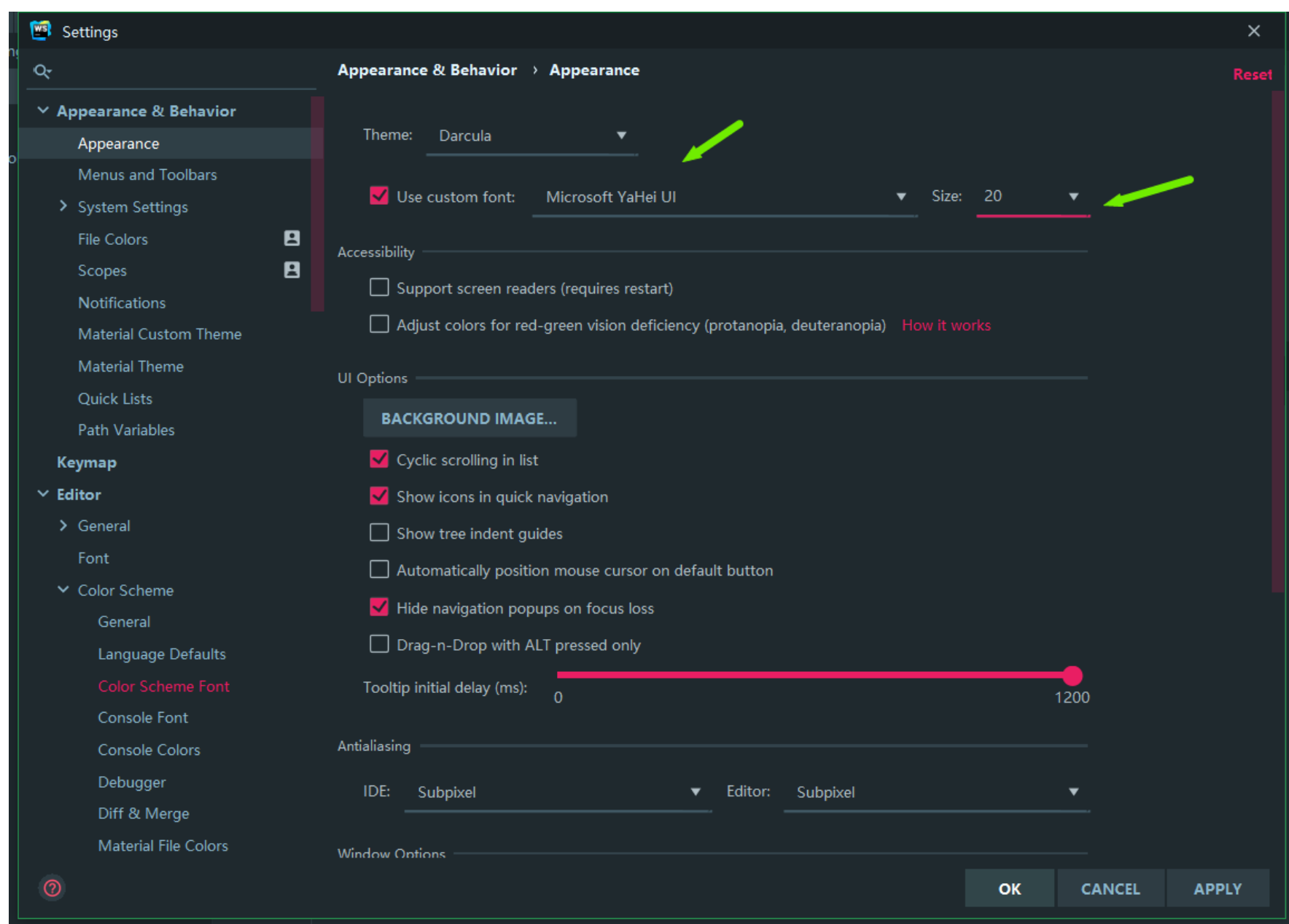
练习

字体

➤ 如果定义了主题

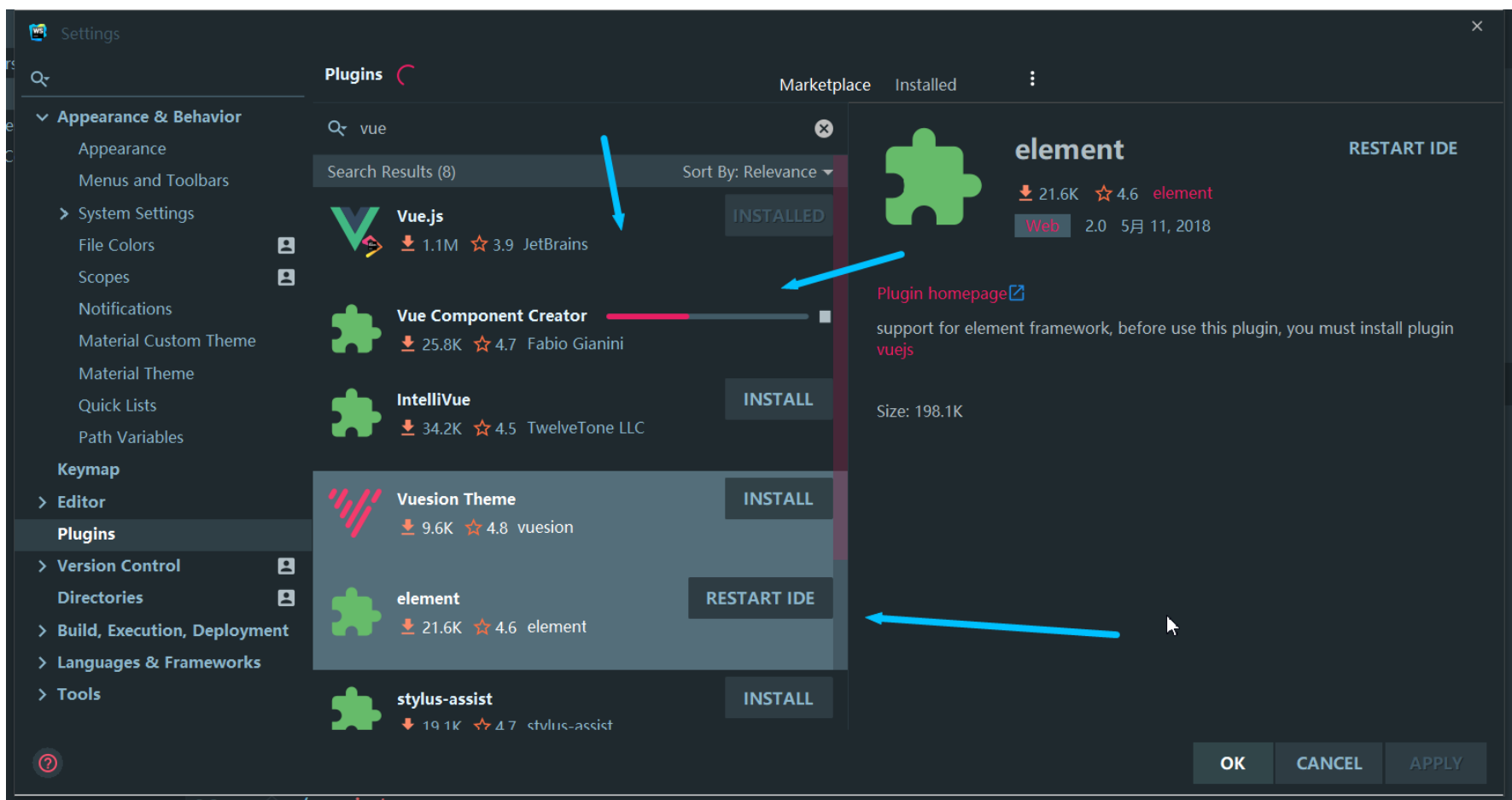


➤ 改变外观字体

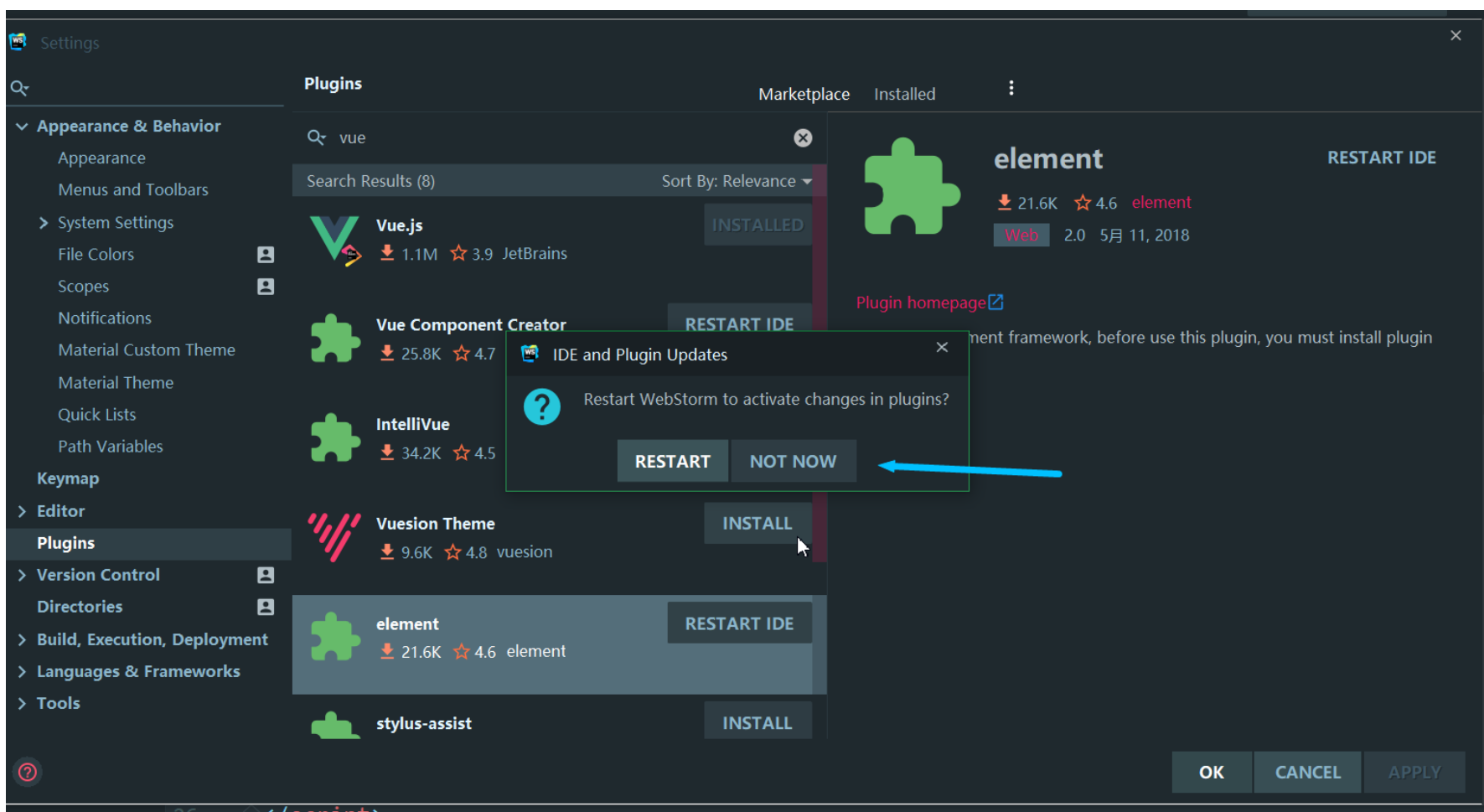


练习

安装 vue 插件



重启



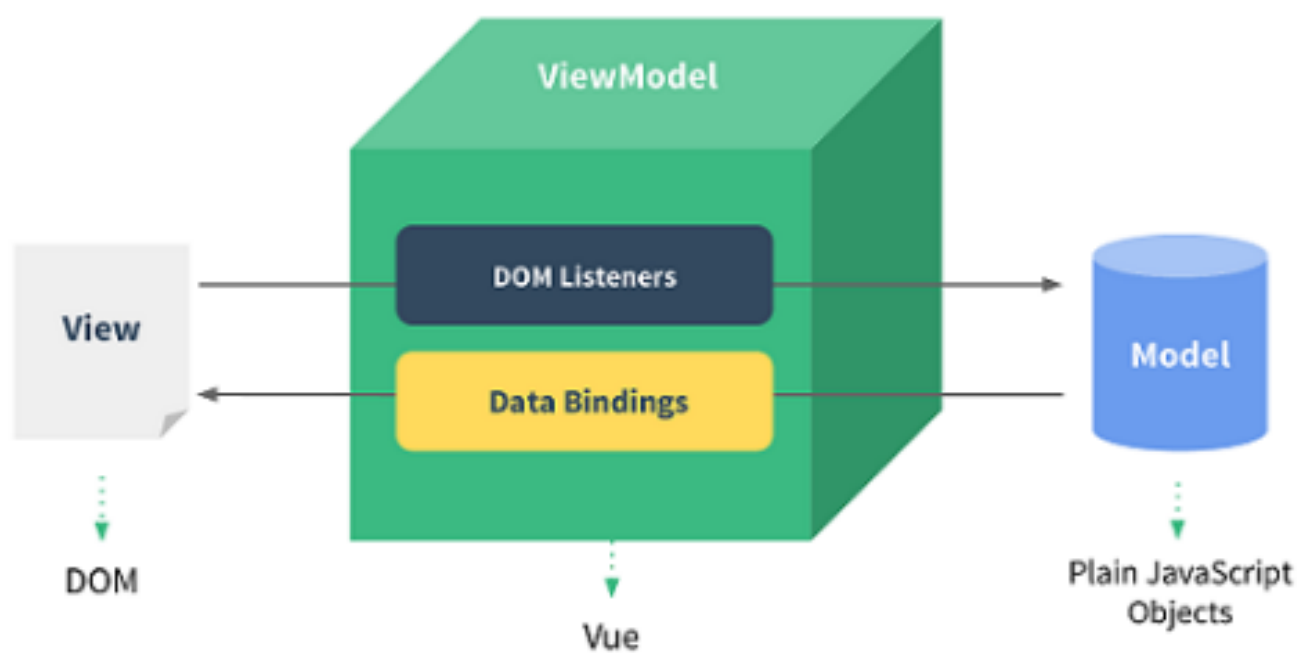
练习

Vue

2019-9-8 20:21:14

介绍

Vue 是一个数据驱动页面的一个框架，基于 MVVM 模式，M 指的是数据，V 值得是视图，VM 是视图模型，将数据绑定视图上（双向绑定），这个框架着重于 VM 部分



Vue-视频

渐进式

当我们需要与后台进行交互，我们就去加入一个 vue-resource 的插件。

当我们需要路由的话，我们就去加一个 vue-router 的插件。

当我们的项目足够复杂，就去加一个 vuex，来进行状态管理。

最后，可以使用构建工具来构建我们的系统。

从简单到复杂，却什么就补什么，这就是渐进式构建。

官网

➤ 英文官网: <https://vuejs.org/>

➤ 中文官网: <https://cn.vuejs.org/>

工具

教程: <https://cn.vuejs.org/v2/guide/>

百度翻译: <https://fanyi.baidu.com/?aldtype=16047#auto/zh>

自学, 了解

下载

➤ 开发环境

```
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
```

➤ 生产环境

```
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
```

代码实操

➤ 初级

```
<!DOCTYPE html>
<!-- 声明语言 -->
<html lang="en">
<!-- 头 -->
<head>
<!-- 声明字符编码 -->
  <meta charset="UTF-8">
<!-- 标题 显示位置 -->
  <title>vue</title>
</head>
<!-- 内容 -->
<body>

<div id="hello">
  {{username}}
</div>
```

```
</body>

<!--引入vue-->
<script src="https://cdn.jsdelivr.net/npm/vue"></script>

<script type="text/javascript">

    let vm=new Vue({
        el:"#hello",
        data:{
            username:"helllowwww"
        }
    })

</script>

</html>
```

➤ 高级

```
<!DOCTYPE html>
<!--声明语言-->
<html lang="en">
<!--头-->
<head>
<!-- 声明字符编码 -->
    <meta charset="UTF-8">
<!-- 标题 显示位置 -->
    <title>vue</title>
</head>
<!--内容-->
<body>

<!--视图 view-->
<div id="hello">

<!--    v-model 双向数据绑定 -->
    <input type="text" v-model="username" />
    <br/>
    hello, {{username}}
</div>

<!--视图 view 模板页面 end -->

</body>

<!--引入vue-->
<script src="https://cdn.jsdelivr.net/npm/vue"></script>

<script type="text/javascript">
```

```
<!-- 视图模型 vm viewmodel -->
  let vm=new Vue({
    // element 元素选择器 # id
    el:"#hello",
    // 数据对象 模型
    data:{
      username:"world"
    }
  })
</script>
</html>
```

➤ 源代码

<https://github.com/username2099/vue-course.git>

练习

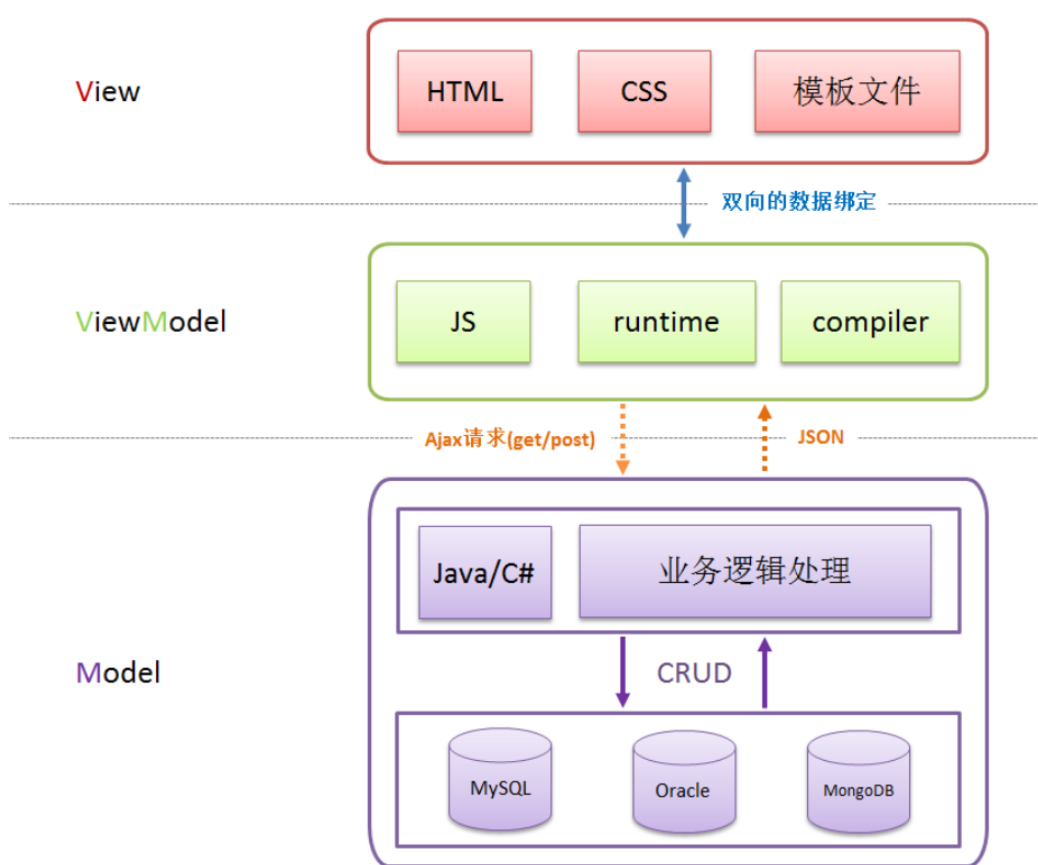
扩展

Mvvm

在 MVVM 中 View 和 Model 是不可以直接进行通信的，它们之间存在这 ViewModel 这个中介充当着观察者的角色。

当用户操作 View，ViewModel 感知到变化，然后通知 Model 发生相应改变，反之亦然。

ViewModel 向上与视图层 View 进行双向数据绑定，向下与 Model 通过接口请求进行数据交互，起到承上启下的作用。



回顾

- 主要讲了什么

课后

2019-9-8 21:26:45

- 了解 vue 的特点
- 复习课本 MVVM

