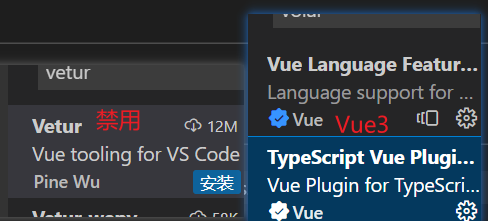
setup和reactive,ref

1. vue3

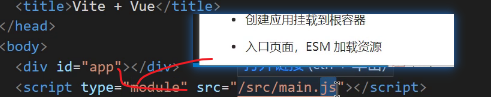
按需导入创建vue实例的函数

Import {createApp } from ‘vue’

导入全局样式文件，导入根组件，创建vue实例，将根组件挂载到vue实例中，将vue实例挂载到页面容器中。



Vue2要有一个根节点，vue3不用。



Composition api的使用，需要配置一个setup函数。在setup中定义的数据默认是没有响应式的（数据源发生改变，页面也跟着发生改变）。如果需要定义响应的对象，可以使用reactive函数，常用于复杂数据类型。如果是简单数据类型，就用ref进行处理。

1. setup函数是一个新的组件选项，作为组件中componsitionAPI的起点
2. 从生命周期角度看，setup会在beforeCreate钩子函数之前执行
3. setup中不能使用this,this指向undefined。实例都没有创建。
4. 在模板中需要使用的数据和函数需要在setup返回。

Vue2习惯写this,在vue3一定要改过来。因为已经没有this了。

在模板中template中要用到的数据和方法，需要在setup中返回。

Import {ref} from ‘vue’

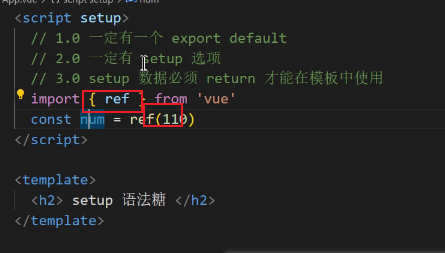
ref函数可以定义响应式数据

接收一个值，返回一个ref响应式对象，对象有唯一

在模板中,ref的value属性会自动解套，不需要额外的value。

ref既可以处理复杂数类型可以处理简单数据类型，需要.value属性进行操作。reactive必须是复杂数据类型。

一定有一个export default 一定有setup选项，setup数据必须return 才能在模板中使用。



计算属性无法赋值。给计算属性赋值的方法，但是可以通过set和ger方法给计算属性赋值

Vue2中的父传子：1、父组件准备好数据 2、父组件通过自定义属性，向子组件中传入数据

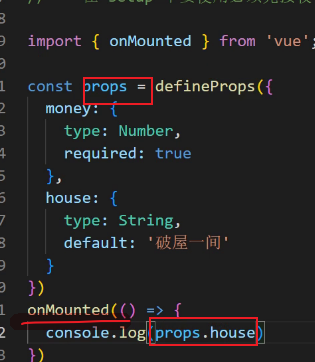
3、子组件通过关键字：props接收数据 4、子组件使用数据

Vue3中的父传子：1、父组件准备好数据 2、父组件通过自定义属性，向子组件传入数据

1. 子组件通过关键字：defineProps接收数据 4、子组件使用数据

组件的使用：直接导入就使用，不用注册。Props：变成defineProps，另外在使用之前必须先接收。

接收数据，通过方法事件干活。



VUE2子传父：1、在子组件中定义数据 2、在子组件中通过this.$emit(‘事件名’，参数)传递数据

在父组件实现事件接收数据。

VUE3子传父：1、在子组件中定义数据 2、在子组件中通过事件传递数据，defineEmit（）在父组件实现事件接收数据。

Vue3没有this。以前是this.$emit。Vue3是通过defineProps获取。

复习：vue3中操作dom

1. 给要操作的dom添加一个属性： ref=’xx’ 2、通过this.$refs.xx得到dom元素，并且进行操作

Vue3中操作dom

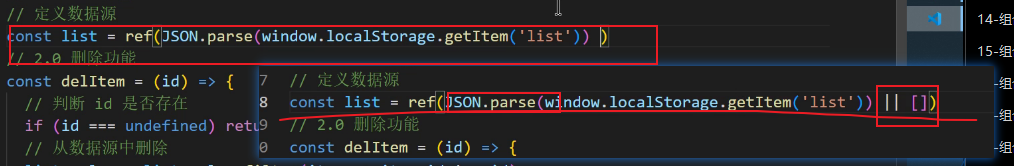
定义一个ref对象：const target=ref(null)

将名称通过ref属性设置给要操作的dom元素，通过target.value就可以获取到dom元素

依赖注入：一个组件向所有的后代组件进行传参。关键字：provide，inject 父组件子组件孙子组件

步骤：父组件:provide(key,value) 后代组件：const value=inject(key)

以前是父传子，子传孙，现在vue3可以父传孙



Num.tolowerCase() 将字符串转为小写。

编译：js代码在书写编辑期间不会对类型进行检查也就是不会报错。执行：运行期间才会进行检查报错。

Vue3源码TS重写，react和ts完美配合，Angular默认支持TS，大中型前端项目首选。

我们使用的所有数据变量对象函数都是要有类型的，定义数据前，数据需要指定明确的类型。

写ts流程：.ts文件，然后通过工具，转为js，最后在浏览器或者node.js里面运行。

类型注解：作用：用来给变量设置类型

一旦设置了类型，将来再次赋值时，只能设置对应类型的值，不能是其他类型，否则会报错。

js原有类型：数字字符串null undefined 复杂数据类型对象数组函数

ts新增类型：联合类型，类型别名（自定义类型），字面量类型 void any 接口 泛型

Type类型别名 type MyType() type保存某一个类型

interface接口类型 extends接口类型-继承 父接口和子接口

类型别名没有继承的能力，但是可以通过交叉类型达到继承的目的。&

接口可以动态新增属性&方法，type一旦定义不能修改。



联合类型|



类型推论：简写，系统自动判别

字面量类型。---多选可选 结合联合类型|使用

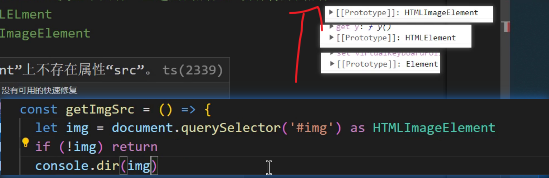
any类型作用是用来逃避TS类型检查，不到万不得已不要使用any

类型断言：有时候你比TS更加希望明确一个值的类型，用类型断言

类型断言在dom操作中比较多。

在ts中给所有的dom元素都设置了类型。dom元素的基本类型就是Element，类型中只有一些通用的属性，没有特殊的属性。强行转换类型。

Console.dir



泛型就是数据类型的占位。占位符 要传递一个方法<T> ;参数和返回值。

参数的占位。

泛型别名：嵌套使用，类型规则

设置默认值withDefaults

在ts中number与Number有没有区别

number类型

Number类型（一个Number对象）

Vue3中写大写，在ts中写小写。

403服务器有这张图片不给访问。有资源不给访问。

dom的类型 ts---HTMLTmageElement

可选链操作符：？



非常确定存在就用！

定义返回数据的类型

自定义类型

项目中安装的第三方库里面都是打包后的JS代码，但是我们使用的时候却有对应的TS 类型提示，在第三方库中的JS代码都有对应的TS类型声明文件。

类型声明文件 .d.ts

内置的类型声明文件。 官方提供的



Ctrl+鼠标左键

js操作，dom操作

使用第三方组件库没有类型声明文件，所以需要下载，下载完显示没有，那就重新启动编辑器。

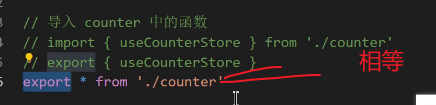
自定义类型声明文件。

Devtools

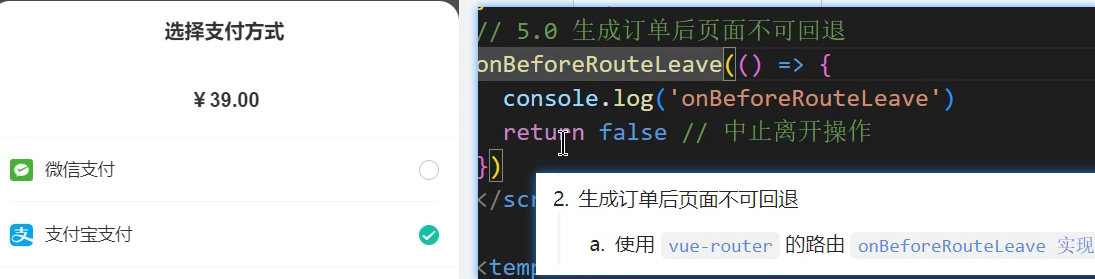
Vue3不分同步和异步。

解构store里面数据，数据是没有响应式的，使用storeToRefs解决仓库状态丢失响应式的问题。

Pinia-plugin-persistedstate







以使用开头---先使用再导入这样子快些。

Socket.io-client 基于websocket的第三方库。

&交叉类型、

轮询：每过一秒或者一百毫秒去发一个请求问服务器订单状态变了没有，性能损耗。

服务器订单发送改变，会向客户端发送请求。使用websocket监听订单状态的变化

第三方包时间包管理 dayjs

图片加载完毕之后再滚动到底部。

