symbol

1. 写法：let c:symblo=Symbol();
2. unique symbol ：因为Symbol 值不存在字面量，所以：unique symbol 表示单个的、某个具体的 Symbol 值。

具体写法：const d :unique symblo=Symbol(); 注意用的是const声明，const命令为变量赋值 Symbol 值时，变量类型默认就是unique symbol，用unique symbol 声明之后的每个变量的值都是不一样的

1. unique symbol 类型可以赋值给 symbol 类型，反过来不行

函数

声明参数的类型，如果不声明就判断为any

写法：

function name(params:type) {

// 函数体

}

function hello(

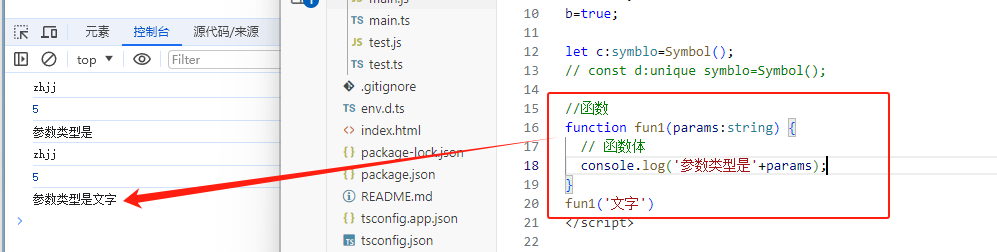
txt:string

):void {

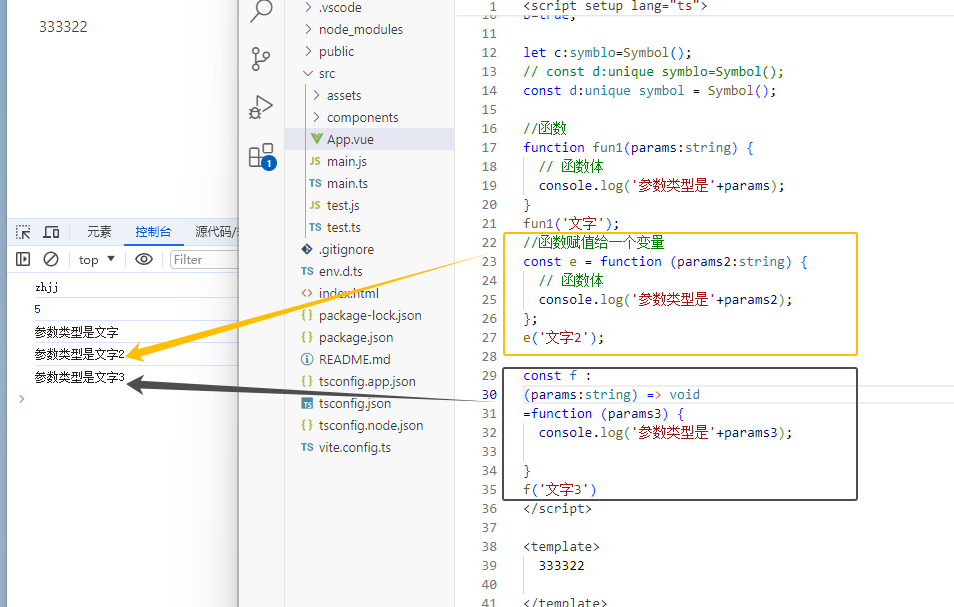
console.log('hello ' + txt);

}

需要给出参数txt的类型（string），以及返回值的类型（void），后者写在参数列表的圆括号后面。void类型表示没有返回值。



函数赋值给变量



函数的参数声明可以少于命名的参数，但是不能超出规定好的参数的个数。参数个数少的函数可以赋值给参数个数多的函数，反之不行

使用typeof运算符从一个值获取类型，用于其他的函数的参数也可以

任何函数都属于Function 类型

箭头函数

const repeat =(

str:string,

times:number

):string => str.repeat(times)

变量repeat被赋值为一个箭头函数，类型声明（ str:string,

times:number）写在箭头函数的定义里面。其中，参数的类型（string,number）写在参数名（str， times）后面，返回值类型（string）写在参数列表的圆括号后面。

如果函数的某个参数可以省略，则在参数名后面加问号表示。

function f(params?:number) {

return x;

};

可选参数要跟在确定参数的后面，如果参数a有可能为空，就只能显式注明类型包括undefined，以及如果设置了参数的默认值，那就表示这个参数是可选的，如果不传入该参数，它就会等于默认值。

void 类型

void 类型表示函数没有返回值。

function f():void {

console.log('22');

}

但是可以返回undefined或null（在函数体里写 return undefined，return null）

never 类型

函数没有返回值，理解成函数不会正常结束执行

函数重载

根据参数的类型，走不同的流程

构造函数

使用new关键字

const d =new Date();

D 是 Date的对象示例，用new 调用， Date()表示的是构造函数