**ES6**

目录

[一、var let const 1](#_Toc492326517)

[1.作用域 1](#_Toc492326518)

[2.重复声明 1](#_Toc492326519)

[3.let const区别 2](#_Toc492326520)

[4.变量声明提升 2](#_Toc492326521)

[二、箭头函数 2](#_Toc492326522)

[1.匿名函数 2](#_Toc492326523)

[2.隐式返回 2](#_Toc492326524)

[3.命名函数 3](#_Toc492326525)

[4.不绑定this 3](#_Toc492326526)

[5.箭头函数不适用场景 3](#_Toc492326527)

[三、函数 3](#_Toc492326528)

[1.函数参数默认值 3](#_Toc492326529)

[四、字符串 4](#_Toc492326530)

[1.模板字符串 4](#_Toc492326531)

[2.标签模板字符串处理用户输入 4](#_Toc492326532)

[3.新增字符串函数 5](#_Toc492326533)

[五、数组/对象 6](#_Toc492326534)

[1.对象解构 6](#_Toc492326535)

[2.数组解构 7](#_Toc492326536)

[3.遍历数组 7](#_Toc492326537)

[六、promise 7](#_Toc492326538)

# 一、var let const

## 1.作用域

var有函数作用域：函数外不能访问函数内定义的变量

let和const有块级作用域：块{}外不能访问块内定义的变量

## 2.重复声明

var允许重复声明

let和const不允许重复声明，重复声明将报错

## 3.let const区别

let允许改变变量值

const允许改变对象变量属性，但不允许改变变量指向

如const obj = {a:”apple”,b:”orange”};

obj = {a:”apple”}; //报错

obj.a = “orange”; //可以执行

## 4.变量声明提升

var变量声明会提前到作用域开始的位置

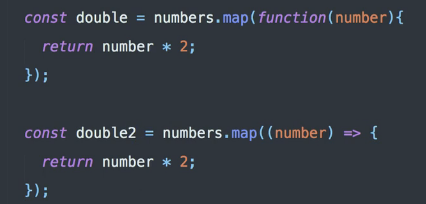
let和const在作用域开始到声明之间为暂时性死区，在暂时性死区引用let/const变量会报错

# 二、箭头函数

## 1.匿名函数

没有参数一定要加()，单个参数可以省略()，多个参数在()内以,分隔，

(argument[0],argument[1],...)=>{};



## 2.隐式返回

删除{}，=>后面的是要返回的值



## 3.命名函数

函数名 = (参数) => {};



## 4.不绑定this

对象的方法的this指向这个对象

但是对象方法内的独立函数（不作为对象方法调用）的this指向window

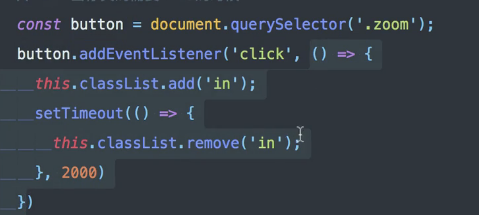
箭头函数内的this继承其父作用域中的this值，并且在定义时就确定下来了，不会因调用方法改变而改变

## 5.箭头函数不适用场景

（1）构造函数，this指向问题

（2）外部对对象方法函数的创建，this指向问题

（3）元素事件的回调函数（元素事件回调函数的this指向这个元素对象），this指向问题

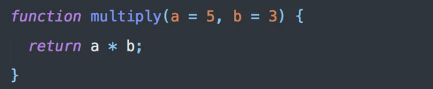


（4）箭头函数中没有arguments对象

# 三、函数

## 1.函数参数默认值

如果没有传入参数，那么参数有一个默认值



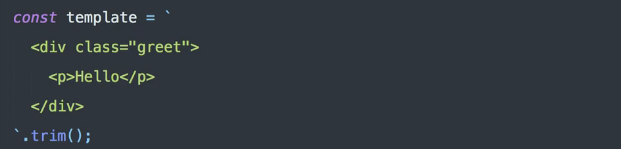
# 四、字符串

## 1.模板字符串

反引号，`${}`，{}内可以是变量、表达式、函数



html模板：





## 2.标签模板字符串处理用户输入

sanitize净化html文本，防止XSS攻击



## 3.新增字符串函数

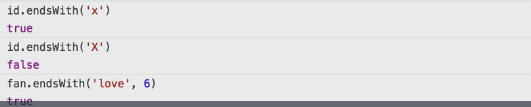
startswith（匹配字符串，匹配起始索引位），大小写敏感





endswith（匹配字符串，最后位的后一位索引位）





includes（匹配字符串，开始匹配的索引号）



repeat（重复次数）



# 五、数组/对象

## 1.对象解构

声明 {变量1，变量2} = 对象

执行原理：

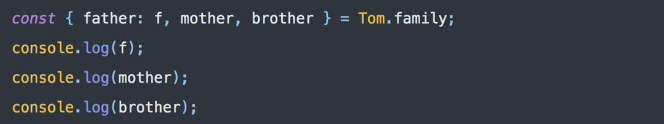
声明变量1,2，

寻找对象中与变量1,2同名的属性

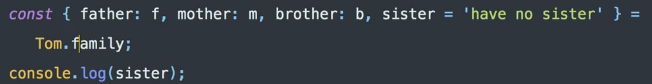
将该属性值分别赋给变量1,2



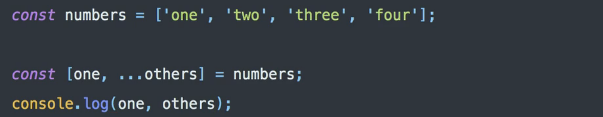
将变量重命名，这样father这个变量是没有被声明过的



设置默认值，当对象没有这个属性的时候，变量为默认值

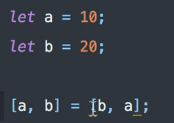


## 2.数组解构





应用场景：交换值

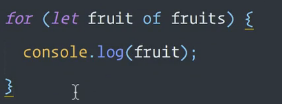


## 3.遍历数组

for of 循环

遍历数组值，不是枚举

循环内可以break/continue



# 六、promise

promise是对象，Promise是构造函数

使一个Ajax请求执行完毕后再执行下一个Ajax请求

