ECMAScript

# Let和const

## 1.let命令

用来声明变量，用法类似于var，但是所声明的变量只在let命令所在的代码块内生效。

## 2.块级作用域

Let的出现实际上是为javascript新增了块级作用域，使用let声明的变量只能在当前代码块内生效；函数也可以在代码块内声明，但范围仅限于当前代码块；使用do表达式获取块级作用域内部最后执行的表达式的值；

Let x = dp{

Let t = f();

t \* t + 1;

}

上面的代码中，变量x会得到整个块级作用域的返回值（t \* t + 1）;

## 3.const命令

const命令声明一个只读的常量。一旦被声明，常量的值就不能被改变。

本质：const实际上保证的是变量指向的那个内存地址不得改动，这就类似与java使用final定义常量；

ES6声明变量的六种方法：var命令和function命令，let和const、import和class；

## 4.顶层对象的属性

顶层对象，在浏览器环境指的是windows对象，在node指的是global对象。ES6中var命令与function命令声明的全局变量依旧是顶层对象的属性，let、const与class声明的全局变量不属于顶层对象属性。

## 5.global对象

在不同的环境中很难使用统一的方式获取到顶层对象，所以ES6引入global作为顶层对象。

# 变量的解构赋值

变量的解构赋值有五种形式：

1. 数组的解构赋值
2. 对象的解构赋值
3. 字符串的解构赋值
4. 数组和布尔值的解构赋值
5. 函数参数的解构赋值

在解构赋值中最好不要使用圆括号；

用途：

1. 交换变量的值
2. 从函数返回多个值
3. 函数参数的定义
4. 提取json数据
5. 函数参数的默认值
6. 遍历map结构
7. 输入模块的指定方法

# Symbol

Javascript数据类型：undefined、null、布尔值（boolean）、字符串（String）、数值（number）、对象（Object）

# Promise 对象

意义：promise是一个容器，保存着某个未来才会结束的事件的结果。

理解：相当于将异步操作的结果存放在promise中。

特点：

1. 对象的状态不受外界的影响。Promise对象代表一个异步操作，有三种状态：pending（进行中）、fulfilled(（已成功）和rejected（已失败）；
2. 一旦状态改变就不会再变，任何时候都可以得到这个结果。

基本用法：

const promise = new Promise(function(resolve, rejected){

//...some code 理解：这里的some code是异步操作

If(/\*异步操作成功\*/){

resolve(value);

} else {

reject(value)

}

})