

ECMAScript 6入门

——贺文榜

ECMAScript 6 简介

1. ES6的目标：是使得JavaScript语言可以用来编写大型项目的复杂应用程序，成为企业级的开发语言！
2. 为什么要学ES6:提高代码的逼格，方便找工作，适应未来！
3. ES6的编译 大部分我们都在chrome浏览器上测试 部分功能我们用webpack 添加对ES6的支持来做测试(<https://github.com/hwb2167sq/zfwebpack>)按照这里的步骤做就好

ES6 目录

- 数据类型的扩张(基本数据类型, 引用数据类型)
- 语法的扩张
- 语句的扩张
- 原型方法的扩张

数据类型的扩张

- Symbol原始数据类型

let 命令和const命令

- let 和const 不会进行预解释(变量提升)
- let 和const不能再同一个作用域内重复声明相同的变量
- const 重新赋值会失败
- let 和const命令的出现也给ES6带来了全新的作用域{}块级作用域

变量的解构赋值(只能用于数组或对象)

数组的解构赋值:

1. 解构赋值的概念:ES6允许按照一定的模式,从数组和对象中提取值,对变量进行赋值,这就被称为解构(Destructuring)
2. rest参数 :把多余的值放到一个数组里面(主要用于获取多余的参数,数组的取值)例如:`var [...myRest]=[1,2,34,]` 最终出来的是个数组
3. 扩张运算符 :将一个数组转换为用逗号分割的参数序列(该运算主要用于函数调用,数组的赋值) `var ary=[1,2];var spread=[12,...ary,34]` 最终出来是逗号分割的参数序列(这是不属于解构赋值的)
4. 数组的解构赋值可以指定默认值 `var [age=18,name="zf"]=[]`

注意点:1. 如果解构不成功,变量的值就等于undefined; `var [foo]=[];`

2.如果对null或者undefined解构就会报错;`var [foo]=undefined;`

数组的解构赋值就到这

变量的解构赋值(只能用于数组或对象)

- 对象解构赋值

1. 对象的属性没有次序，变量必须与属性同名，才能取到正确的值
2. 如果要给变量名起(别名:就是变量名和属性名不一致了); `var {gender:sex}={gender:'男'};`
3. 如果将一个已经声明的变量进行解构赋值必须这样写:`var x; ({x}={x:1})`
4. 对象的解构赋值也可以指定默认值: `var {name="hwb"}={}`

字符串的扩张

- `includes();` // 包含字符串
- `startsWith();` // 是否在开头包含
- `endsWith();` // 是否在结尾包含
- `repeat();`
- 模板字符串 ``hwb${name}``

数值的扩张

- `Number.isInteger()`; ['IntId3ə(r)] // 判断是否是一个数 整数
- `Number.isFinite()` ['faɪnaɪt] // 判断是否是一个数 除去NaN
- -2的53次方 到2的53次方 `Number.MAX_SAFE_INTEGER`
- Math对象的扩张
- `Math.trunc()`; 去掉小数部分

数组的扩张

- `Array.from()`; 类数组转化数组
- `Array.of()`; 创建数组
- `find()` 数组实例的方法找到第一满足条件的数组元素
- `findIndex()`;

对象的扩张

- `Object.is()`;用来比较俩个值是否严格相等
- `Object.assign()`;
- 增强的对象写法
- 属性名表达式

函数的扩张

- 函数参数的默认值
- rest参数
- 扩张运算符
- 箭头函数
- 箭头函数注意:1:this指向 不可以当做构造函数 不能使用arguments对象

Set和Map数据结构

- ES6提供了新的数据结构Set。它类似于数组，但是成员的值都是唯一的，没有重复的值。
- Size属性

操作方法

`add(val);delete(val);has(val);clear()`

遍历方法

`keys();values();entries();forEach();`

Map数据结构

- ES6提供了Map数据结构。它类似于对象，也是键值对的集合，但是“键”的范围不限于字符串，各种类型的值（包括对象）都可以当作键。也就是说，Object结构提供了“字符串—值”的对应，Map结构提供了“值—值”的对应，是一种更完善的Hash结构实现。
 - `Var map=new Map([[‘name’,’张三’],[‘author’,’hwb’]]);`
 - 实例的属性和方法
 - size属性
 - 操作方法
 - `set(key,value);get(key);has(key);delete(key);clear();`
- 遍历方法
- `keys();values();entries();forEach();`

异步的解决方案Promise对象

- Promise的构造函数接收一个参数，是函数，并且传入两个参数：resolve, reject，分别表示异步操作执行成功后的回调函数和异步操作执行失败后的回调函数。resolve是将Promise的状态置为fulfilled，reject是将Promise的状态置为rejected。
- 原型上的then方法

Class的基本语法

- ES6提供了更接近传统语言的写法，引入Class类的概念作为对象的模板通过class关键字，可以定义类

- ```
class Add{
 constructor(x,y){
 this.x=x;
 this.y=y;
 }
 equal (){
 return this.x+this.y;
 }
}
```

注意:定义类的方法时不需要加function关键字 也不需要逗号分割



# Module

- Module是解决模块化的问题
- 模块的导出export
- 模块的导入import
- 1.js `var name = 'jw'; var age = 23; export {name, age}`
- 2.js `import {name, age} from './1.js'` 也可以 `import * as pro from './1.js'`