

## ES6

### let/const

let  
const

### String

查找方法

String.prototype.includes  
String.prototype.startsWith  
String.prototype.endsWith

utf-16

两个码元

String.prototype.codePointAt  
String.fromCodePoint  
String.prototype.normalize

原有的正则表达式处理串元, 所以新增了正则  
处理方法, 看下面正则部分

模板字符串

多行字符串  
单双引号不必转义  
字符串格式化, 提供变量替换的能力  
触发执行函数  
自定义标签函数

### RegExp

两个码元字符串处理

u标志(模式) —— 直接操作代码点

备注

不支持的情况下报错, 用之前注意浏览器支持

RegExp.prototype.source

RegExp.prototype.flags

### Function

默认参数

不能再剩余参数中使用具名参数

剩余参数

不能用于 setter 中  
增强构造函数

扩展运算符(...)

函数名(name)

new.target

解决函数调用不确定性 —— Person.call(Person)

箭头函数

没有下列属性

super  
new.target  
arguments

不允许重复的具名函数

es5 this 由函数在哪里调用决定, 并不在函数定  
义决定

this

箭头函数的 this 有包含他的函数的定义时的  
this 决定

尾调用

返回的时候只能一个纯的函数调用, 不能有其他  
的操作 —— 目前只有 safari 支持

### 对象

字符串语法扩展

{ name, age }  
{ 0 }  
{ name: 'a' }

静态方法

Object.is —— Object.is(NaN, NaN) true  
Object.is(+0, -0)  
Object.assign —— 用main

重复属性名检查

自有属性的枚举顺序

数字按照升序  
字符串/符号按照添加顺序

原型

Object.setPrototypeOf  
Object.getPrototypeOf  
super —— 对象里只能适用于缩写方法  
函数中私有化了\_\_proto\_\_属性

### 数组

### 解构

对象解构  
数组解构  
参数解构

### Symbol

### Set/Map

### 迭代器和生成器

Symbol.iterator

迭代器

内置迭代器

entries —— Map  
Set  
Array

values —— Map  
Set  
Array

扩展运算符 —— 用在任意迭代器上

是否迭代判定

生成器

iterator.throw()  
iterator.next(data)  
return —— 让生成器更早的完成, 结果作为 done 时候的返  
回值

运行器

异步任务运行器  
Promise 运行器  
组合生成器

### 类

### Promise

异步解决方案

pending  
fulfilled  
rejected

生命周期

Promise.prototype.then  
Promise.prototype.catch  
Promise.resolve  
Promise.reject  
Promise.all  
Promise.race  
Promise和generator

### 代理和反射接口

### 模块

### Number

IEEE754

Number.isInteger  
安全整数范围 —— 2\*\*53-2\*\*53  
Number.MAX\_SAFE\_INTEGER  
Number.MIN\_SAFE\_INTEGER  
Math新增高精度方法