**JS 高级程序设计&ES6**

## 一. 引言html中的js

* + js外部引用一般放在body后面，防止js加载时间过长，页面加载速度慢。
  + script中：
    - defer属性：外部js立即下载，但是延迟执行，响应domcontentload事件；
    - async属性：适合外部js文件，立即下载，但是不保证外部js的执行顺序；

二. 外部js文件优势

* + 可维护性
  + 可缓存

## 二. 第3章 基础知识

### 严格模式：

* 1. “use strict” 严格模式声明，可在html头部或者函数内部；

### es6 简介：

* 1. babel转码器将es6转为es5代码；

### 变量：

* 1. var定义的变量为定义该该变量作用域内的**局部变量，省略**var可以定义全局变量；
  2. let声明变量，代码块内有效；可在for循环内使用，减少全局变量；（for循环的循环变量属于父作用域，循环体属于子作用域）；let声明的变量必须在声明后才能使用（**变量不提升**）；子作用域内的let定义变量和外部变量同名时，变量与子作用域绑定，需let声明后才能使用（**暂时性死区**）；相同作用域内，不能在函数内部重新声明参数（**no重复声明**）；
  3. 函数声明会提升到全局作用域或者函数作用域的头部（**函数声明提升**）；
  4. es6允许块级作用域内声明函数，会被提升到块级作用域头部；
  5. **do表达式：**块级作用域没有返回值，采用do｛｝可以返回内部最后执行的表达式的值；
  6. **const命令：**声明一个只读常量，即声明时必须初始化，性质和let相似；**本质**：不是变量的值不能改动，而是变量指向的那个内存地址不得改动；因此可以改变变量的数据结构，即添加属性；
  7. **小结：6中声明变量的方法：var function let const import class；**

### 数据类型

1. **undefined**：对未声明和未初始化的变量执行typeof均返回undefined；派生自null；
2. **null：空对象指针，**typeof（null）返回object；null＝＝undefined //true;

**Boolean:** true &false，小写，true不一定等于1，false不一定等于0；boolean()转型函数；

|  |  |  |
| --- | --- | --- |
| **数据类型** | **True** | **False** |
| **Boolean**  **Number**  **String**  **Undefined**  **Object** | **True**  **非0数值**  **非空字符串**  **n/a**  **任何对象** | **False**  **0和NAN**  **空字符串**  **undefined**  **null** |

**c. Number：**不要测试某个特定的浮点数值，存在误差；

* + 1. isFinity()判断是否为无穷数；
    2. NAN：用于表示一个本来要返回数值的操作数未返回数值的情况，防止抛出错误，如 1/0；任何涉及nan的操作都返回nan；nan与任何数值不等，包括nan本身；isNaN()将参数转为数值，不能转为数值的数据，返回true;

**d. 数值转换**

* + 1. **number(x):** 忽略前导的零，x=undefined//NaN, **x=null//0, x=””//0**,x=非小数点，数字，进制符号等，返回NaN;
    2. **parseInt(x,basicNum)**：参数＋基数，基数（进制）规定参数的转换格式，忽略前导的零，如果第一个字符不是数字字符或者负号，返回**NaN（number（“”）＝0**）;只识别数字字符，不能转换为浮点型；
    3. **parseFloat()，**可识别浮点数值个是，但是十六进制字符串返回0；且直识别第一个小数点，第二个小数点后的数值直接别忽略；

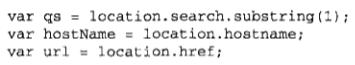
**string类型**

* + 1. **x.toString(basicNum):**数值，布尔值、对象和字符串值有该方法，null和undefined没有该方法；
    2. **String()：**可以转换任何类型的数值为字符串；

### 语句：

1. **for－in语句：**可以枚举对象属性，当遇到undefine和null时，循环体停止；
2. **label语句：代码中添加 标签；**
3. **with语句：**将代码作用域设置到一个特定对象，（**严格模式下禁止使用with**）
4. **没有重载**：函数名属于后定义的函数；

### 变量的解构赋值：

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

## 三．第四章：变量、作用域和内存问题

### 变量

1. 基本类型：number、string、boolean、undefined，null，固定大小空间，保存在栈内存中；
2. 引用类型：对象，堆内存中，变量实为指向对象的指针；
3. 检测引用类型是什么类型对象：instanceof，x instanceof array／object…

### 执行环境及作用域：

1. 可向上搜索作用域链，查询变量和函数名；
2. 不能向下搜索作用域链，进入另一个执行环境**；**

### 延长作用域链

1. try-catch语句的catch块
2. with语句中的变量

### js中没有块级作用域，只有全局和函数作用域，所以在逻辑语句如if for中定义的变量是全局变量；

### 垃圾收集机制；

1. 标记清除：进入环境／离开环境
2. 引用计数：初始化为1，赋值给其他变量＋1，其他变量值发生改变－1，当为0时，收回占用空间。（循环引用不能使用该方法，或者手动断开，均设置为null）

### 内存管理

1. 原因：系统分配给浏览器的内存较少，防止加载js导致系统崩溃；
2. 优化思路：为执行中的代码只保存必要的数据，当数据不再有，设为空（解除引用），适用于全局变量和全局对象属性**；**

## 四．第五章：引用类型

### object类型：new object／对象字面量

### array类型：

1. new array(x),x可以为值或者长度，array（“a”，“b”）／array（10）；
2. array.length属性可读可写，为赋值的值为undefined
3. **数组检测**:
   * 1. value instanceof array，当一个框架向另一个框架传递参数时，数组的的原生创建的数组具有不同的构造函数；isarray解决此问题；
     2. Array.isArray(value)
4. **转换方法**：toString, toLocalString, valueOf;
5. **栈方法**：push()后增, pop()后溢;
6. **队列方法**：push()后增, shift()前溢
7. **重排序：**reverse(), sort(compare) function compare(x,y){return x-y};
8. **操作方法：**
9. **concat()**连接两个数组或者参数，返回组合后的数组；
10. **slice()**子数组,创建新的数组，一个参数表示起始位置，2个参数表示起始和结尾，返回**子数组**；
11. **splice(x,y,z)**,x表示起始位置，y表示删除元素的长度，z表示新增的元素，可实现删除，替换，插入，返回被删除的元素；
12. **位置方法：indexof(), lastindexof();**
13. **迭代方法：**
14. **every(function())**，每一项元素执行function都返回true，则every返回true；
15. **some(function())**，存在一项执行function返回true，则some返回true；
16. **foreach(function())**，每一项执行function，无返回值；
17. **map(function())**，每一项执行function，返回执行结果组成的数组；
18. **filter(function())**，每一项执行function，返回结果为true的数组组成的新数组；
19. **缩小方法：**reduce（function（pre，cur，index，array）｛｝）正向遍历数pre为function上一次返回的数值；lastreduce反向遍历数组；

### Date类型

1. **创建方法：new Date(Time);**
2. **日期格式化方法：**
3. **toDateString()，**
4. **toTimeString()，**
5. **toLocaleDateString()，**
6. **toLocaleTimeString()，**
7. **toUTCString()，**
8. **toLocaleString()，**
   1. **日期／时间组件方法（详情查手册）**

### RegExp类型

### Function类型：

声明语法定义和表达式定义；function name（）；name＝function（）；

1. **函数名为指向函数的指针；**
2. **没有重载，**后者会覆盖前者；
3. **作为值的函数：**函数名本身就是变量，所以函数也可以作值使用；
4. **函数内部属性：**
5. **arguments**（该对象由callee属性指向拥有argument的函数，可解耦递归）
6. **this，**引用的是函数据以执行的环境对象**；**
7. **caller，**保存调用当前函数的函数引用,无法赋值；（argument.caller）
8. **函数的属性和方法：**
9. **length：**函数接受参数的个数
10. **prototype：**保存引用类型的实例方法；
11. **bind()**，绑定传入的参数给bind函数；
12. **call（）、apply（），**作用相同，参数不同，实质：在当前上下文对象调用调用别人的方法，就像调用自己的一样；eg：function a(x,y){

return x+y;} function b(x,y){a.call(this,x,y)} a(1,2)该方法是在a方法中调用b方法。

### 基本包装类型

1. number类型
   1. 创建：new number（x）
   2. 数值格式化字符串的方法
      1. num.toFixed(x)，返回数字x位有效位的字符串，x位有效位数；
      2. num.toExponential(x)，返回数字指数形式的字符串，x位有效位数；
      3. num.toPrecision(x),根据num返回合适的数字表示方法，x同上。
2. Boolean类型
   1. 创建：var a＝new Boolean（true）／b＝false，最好使用后者，因为a转化为基本类型的值为1，boolean（obj）＝1；
3. String类型
4. 创建：new string（“hello world“）；
5. 字符方法：