[一.简介 2](#_Toc503714269)

[二.变量格式转换 2](#_Toc503714270)

[2.1 用户的输入 2](#_Toc503714271)

[2.2 字符串怎么变为数字 2](#_Toc503714272)

[三.js事件 2](#_Toc503714273)

[3.1 入口函数 3](#_Toc503714274)

[四.DOM 3](#_Toc503714275)

[4.1 DOM定义 3](#_Toc503714276)

[4.2节点 3](#_Toc503714277)

[4.3 访问节点 3](#_Toc503714278)

[4.4 封装自己的class类 3](#_Toc503714279)

[4.5 访问关系 5](#_Toc503714280)

[4.6 DOM节点操作 6](#_Toc503714281)

[4.7 设置节点属性 6](#_Toc503714282)

[五.数组 6](#_Toc503714283)

[5.1.追加数组 7](#_Toc503714284)

[5.2 删除数组 7](#_Toc503714285)

[5.3 concat，join，split数组 7](#_Toc503714286)

[六.定时器 8](#_Toc503714287)

[七.倒计时 8](#_Toc503714288)

[八.网址编码 8](#_Toc503714289)

[九.缓动动画（动画是慢慢停下来） 8](#_Toc503714290)

[十.offset家族 8](#_Toc503714291)

[十一.scroll 家族 9](#_Toc503714292)

[11.1 scrollTop scrollLeft 9](#_Toc503714293)

[十二.事件对象event 10](#_Toc503714294)

[十三.json 10](#_Toc503714295)

[十四.闭包 10](#_Toc503714296)

# 一.简介

js用来制作web页面交互效果，提升用户体验

web前端三层：

1）结构层 HTML 从语义的角度，描述页面结构

2）样式层 CSS 从审美的角度，美化页面

3）行为层 JavaScript 从交互的角度，提升用户体验

js是前台语言，而不是后台语言

js运行在客户的电脑里，而不是服务器上，所以我们称为“前台语言”，“后台语言”是运行在服务器上的，比如PHP、JSP、ASP等，这些语言都能对数据库进行操作。

-var g = function(){} 这种叫函数表达式

# 二.变量格式转换

## 2.1 用户的输入

var a = prompt(“请输入点什么东西”);

prompt就是专门用来能够让用户输入的对话框

与alert不同，alert直接使用，不需要变量

prompt必须用一个变量，来接收用户输入的值

在prompt中不管输入什么，都是字符串

## 2.2 字符串怎么变为数字

parseInt（）；

1）不管里面写什么，都只将最开头的数字部分由字符串转化为数字，后面的自动消失（自动净化功能）

2）自动带有取整功能，会自动把小数去掉，不会四舍五入

parseFloat（）是小数

# 三.js事件

事件源.事件 = function（）{事件处理函数}

## 3.1 入口函数

window.onload = function(){

内部放js

}

这个函数的意思就是，当我们页面加载完毕之后，才去执行函数体里面的js部分

# 四.DOM

js由三部分组成

1）ECMAscript（核心）欧洲计算机制造商协会，描述了JS的语法和基本对象

2）文档对象模型（DOM），处理网页内容的方法和接口

3）浏览器对象模型（BOM），与浏览器交互的方法和接口

## 4.1 DOM定义

DOM为文档提供了结构化表示，并定义了如何通过脚本来访问文档结构。目的其实就是为了能让js操作html元素而制定的一个规范。

DOM树：将网页划分了结构，帮我们访问每一个结构。

## 4.2节点

由结构图可以看到，整个文档就是一个文档节点。

每一个HTML标签都是一个元素节点。

标签中的文字则是文字节点。一切都是节点。

## 4.3 访问节点

getELementById（）id访问节点

getELementsByTagName() 标签访问节点

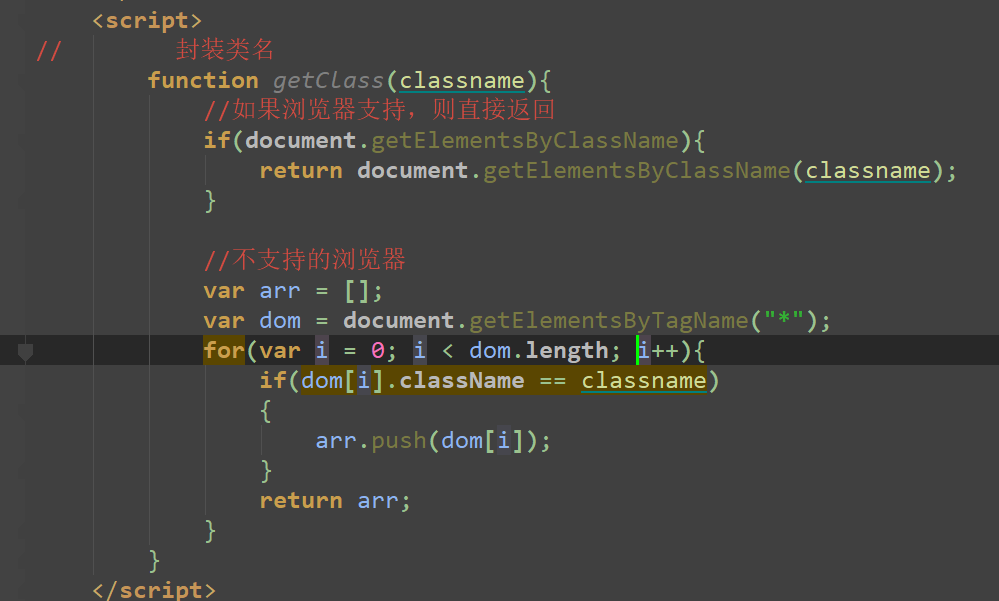
getELementsByClassName() 类名

主流浏览器支持，IE6,7,8,不认识

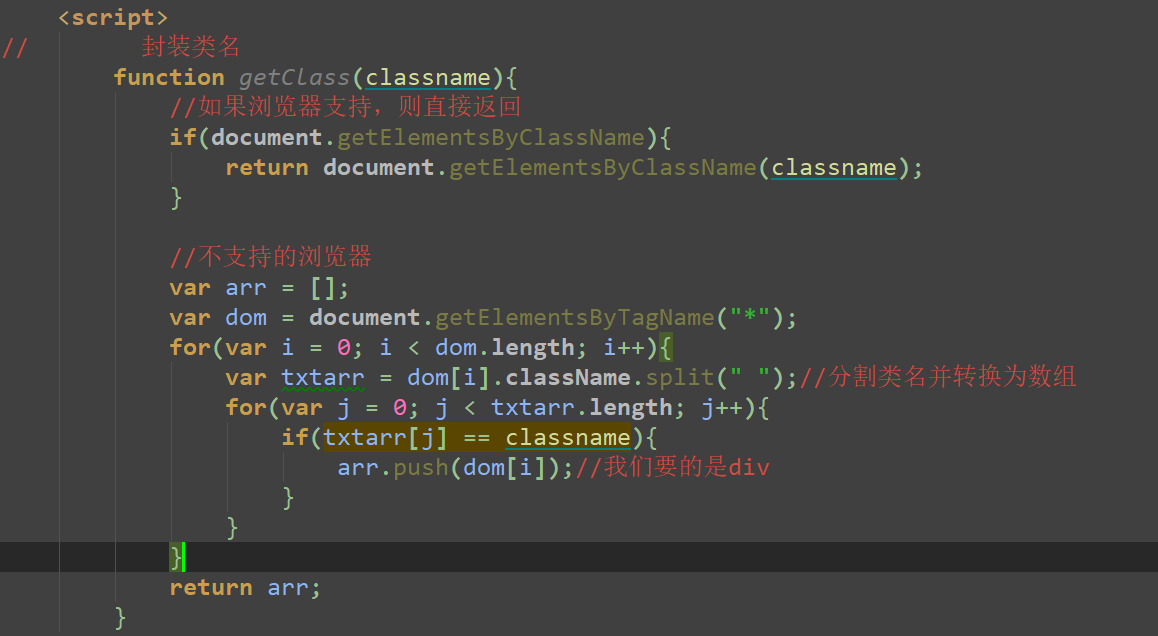
## 4.4 封装自己的class类

原理：

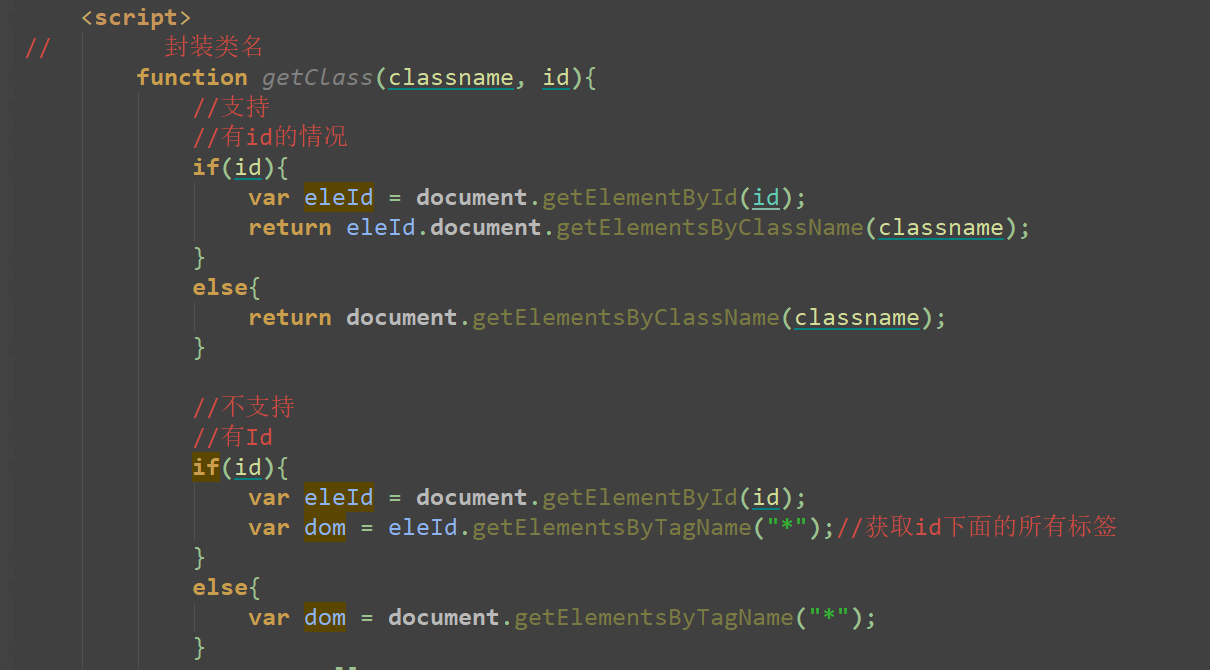
我们要取出所有的盒子，利用遍历的方法，通过每一个盒子的className来判断。如果相等就留下。

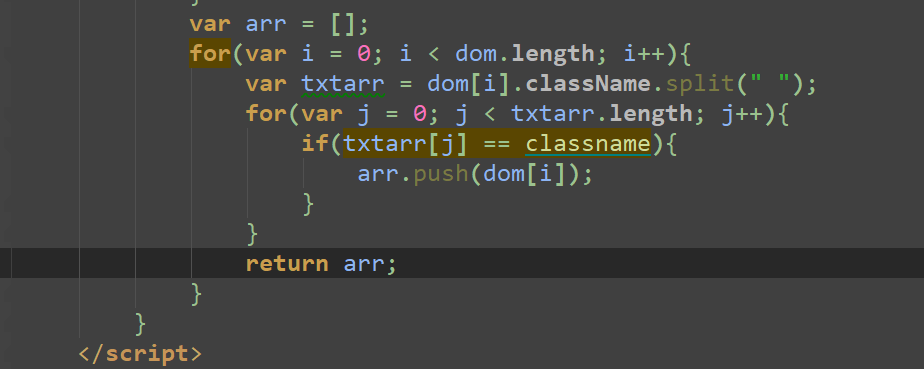


以上只适用于class名只有一个：getClass(“demo”)



可以使用于多个类名: getClass(“demo one”)





适用于带有id的：getClass(classname, id),且支持多个类名。id放在后面是为了避免没有id而发生错误

## 4.5 访问关系

各个节点的相互关系





不带Element的 IE678认识，带的其他浏览器认识

为了兼容，可以合写，用||符号：

var nextSib =名字.nextElementSibling || 名字.nextSibling;

必须先写正常浏览器 后写IE678

子节点很少用，因为空格也算第一个孩子

## 4.6 DOM节点操作

新建节点 插入节点，删除节点 克隆节点等

1）创建节点

var jiedian = document.createElement(“li”);

上面的意思就是生成一个新的li标签

2）插入节点

1.appendChild(想要添加的标签) 添加子节点，放进盒子最后面

2.insertBefore(插入的节点，参照节点) 添加子节点

写满两个参数。如果第二个参数为null，默认吧新生成的放到最后面

3）删除节点

removeChild()

4）克隆节点

cloneNode()括号里面可以跟参数，如果是true则为深层复制，除了复制本节点，还复制子节点；如果为false 浅层复制，只复制本节点

## 4.7 设置节点属性

1）getAttribute(“属性”) 获取属性

通过这个方法可以得到某些元素的某些属性

2）setAttrbute(“属性”, “值”)设置节点属性

3）removeAttrbute(“属性”)删除节点属性

# 五.数组

数组常用方法：

push、pop

unshift 、shift

concat、join、split

## 5.1.追加数组

1）push（）

可以向数组末尾添加一个或多个元素，并返回新的长度

var arr = [1,3,5] arr.push(7) 结果变成[1,3,5,7]

console.log(arr.push(7));这样返回的是数组的长度

2）unshift（）

可以向数组开头添加一个或多个元素，并返回新的长度

var arr = [1,3,5] arr.unshift(0) 结果变成[0,1,3,5]

## 5.2 删除数组

1）pop()

移除最后一个元素，返回值是返回最后一个值

var arr = [1,3,5] arr.pop() 结果[1,3]

console.log(arr.pop());返回的是最后一个元素5

2）shift（）

用于吧数组第一个元素从其中删除，并返回第一个元素的值

var arr = [1,3,5] arr.shift() 结果[3,5]

## 5.3 concat，join，split数组

1）concat（）

用于连接多个数组，他不会改变现有数组，仅仅返回被链接数组的一个副本

var arr1 = [1,3,5] arr1.concat(arr2) 结果[1,3,5“a”,“b“,”c“]

var arr2 = [“a”,“b”,“c”]

2）join

作用是将数组各个元素通过指定的分隔符进行连接成为一个字符串

语法：arrayObject.join（separator）//数组名.join（符号）；

数组转换为字符串

var arr = [“aa”, “bb”, “cc”] console.log(arr.join(“-”)) 结果aa-bb-cc

数组本身不会改变。如果省略separator参数，则使用逗号做为分隔符

3）split（）

将字符串转换成数组

语法：arrayObject.split（separator，howmany）

如果省略separator参数，则使用逗号做为分隔符

howmany可选。该参数可指定返回的数组的最大长度

separator的符号要跟字符串中的符号保持一致，不然不知道分割点

# 六.定时器

我们可以设定时间，让某个动作不断的去执行。这个我们在js中用定时器来表示

setInterval（“执行的函数”，间隔时间）；

# 七.倒计时

倒计时=将来时间-现在时间

如果 date括号里写日期，就是定义想要的时间，时间之间用斜线隔开，时间和时分秒中间用空格隔开

# 八.网址编码

我们知道一个网址，自己的网址，不同页面也有自己的id网址，我们经常会做一些，吧网址送入到后台，但是后台再处理不认识比如换行、特殊符号等的。所以我们要事先编码，然后再传到后台。

1）encodeURIComponent()函数可把字符串作为URI组件进行编码

2）decodeURIComponent()函数可把字符串作为URI组件进行解码

# 九.缓动动画（动画是慢慢停下来）

初始值为零 var leader = 0 目标位置var target

公式：

leader = leader + (target - leader) / 10（随意一个数，一般10就可以） //算的是步长

元素.style.left = leader + "px";

盒子要设置定位，初始位置left：0；

# 十.offset家族

js中有一套方便获取元素尺寸的方法就是offset家族

1）offsetWidth,offsetHeight

得到对象的宽度和高度（自己的，与他人无关）

offsetWidth = width + padding +border;

.style.width只能得到行内的数值

2）offsetLeft,offsetTop

offsetLeft返回距离上级盒子（带有定位）左边的位置，如果父级都没有定位则以body为准

这里的父级指的是所有上一级，不仅仅指父亲

注：1）offsetTop和style.top区别

最大区别在于offsetTop可以返回没有定位盒子的距离上侧的位置，style.top不可以

2）offsetTop返回的是数字，而style.top返回的是字符串，除了数字还有单位：px

3）offsetTop只读style.top可读写

4）如果没有给HTML元素指定过样式，则style.top返回的是空字符串

offsetLeft从父亲的padding开始算，父亲的border不算

3）offsetParent

返回该对象的父级（带有定位）

1.如果当前元素的父级元素没有进行css定位（position为relative或absolute），offsetParent为body

2.如果当前元素 的父级元素中有定位（position为relative或absolute），offsetParent取最近的那个父级元素。

# 十一.scroll 家族

## 11.1 scrollTop scrollLeft

scrollTop被卷去的头部。他就是当你滑动滚轮浏览网页时网页隐藏在屏幕上方的距离

我们学过一个页面滚动效果：

window.onscroll = function(){页面滚动语句}

谷歌：document.body.scrollTop

火狐和其他：document.documentElement. scrollTop

ie9+和最新浏览器

window.pageXoffset； pageYoffset(scrollTop)

兼容性写法：

var scrollTop = window.pageYoffset || document.documentElement.scrollTop || document.body.scrollTop || 0

# 十二.事件对象event

我们学过一些事件：onclick，onmouseover。。。

btn.onclick = function(event){语句}

event是事件的意思，最常在这样函数中写

event就是事件的对象（比如上面的event指的是onclick的对象）

在触发DOM上的某个事件时，会产生一个事件对象event，这个对象中包含着所有与事件有关的信息。所有浏览器都支持event对象，但支持的方式不同。比如鼠标操作时候，会添加鼠标位置的相关信息到事件对象中

常见写法：var ev = event || window.event;

# 十三.json

json是一种轻量级的数据交换格式，我们称之为Javascript对象表示法。使用json进行数据传输的优势之一。json实际上就是Javascript。json很像我们学过的样式条；

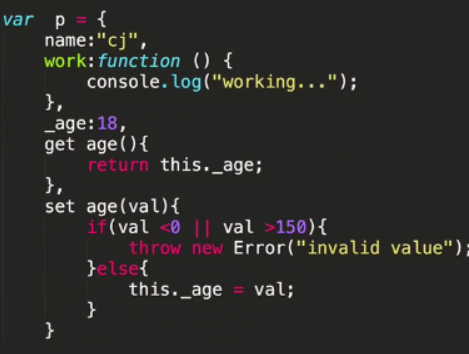
var myjson = {k:v, k:v, k:v …}

json一般就是被当作一个配置单用

var json = { key:value, key:value, key:value, …}

# 十四.全局预处理

创建对象的属性可以这么写：

-

-处理冲突

-处理函数声明有冲突，会覆盖

-处理变量声明有冲突，会忽略

# 十五.作用域

-大括号中为块作用域（js中无块作用域）

-动态作用域

-词法作用域（也称静态作用域活闭包）

# 十六.闭包

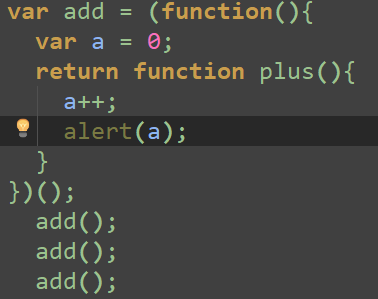
-减少全局变量

-减少传递给函数的参数数量

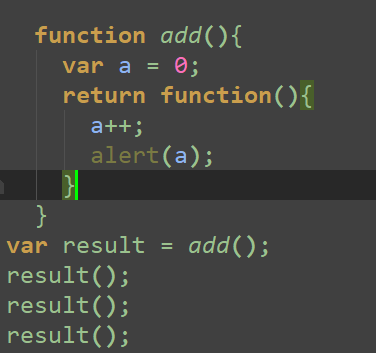
-封装

两种写法

-1

-

-2



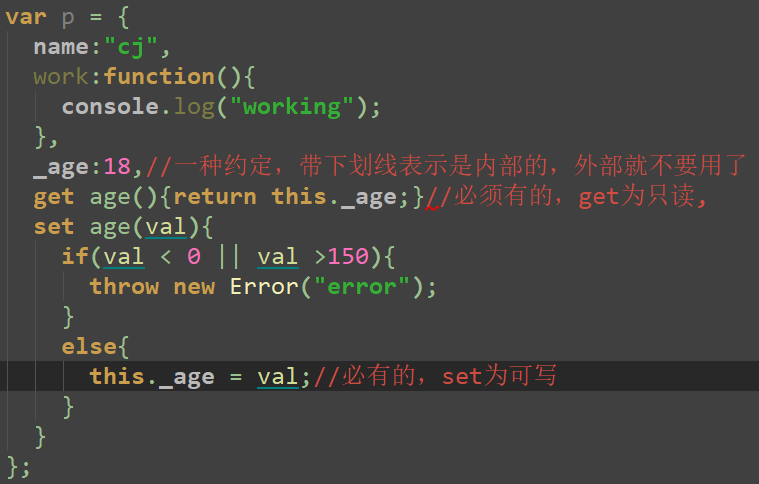
# 十七.对象

分三类：js内置的（如Number），宿主环境（如window），自己创建的

-构造器：.constructor

## 17.1 对象的创建

-对象字面量形式：属性，方法，get、set型属性



-object形式

+Object.defineProperty

+Object.defineProperties

## 17.2 对象的基本操作

-成员的遍历

-检查对象是否有某个特性

-成员的删除（有些属性是不能删除的）：delete sth

## 17.3 成员特性

-得到对象的属性特性描述：Object.getPropertyDescriptor

-enumerable

-writable

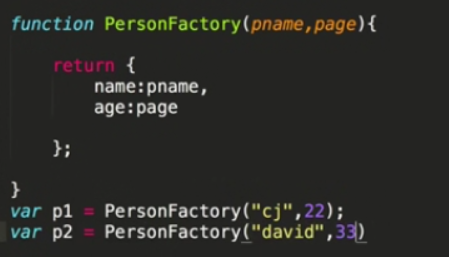
-configurable

## 17.4 类型检查

-typeof

-instanceof

## 17.5 对象工厂



如果是重复相同的模式，可以用函数，如上

缺点是无法创建共有的东西，这样都是单独的。

## 17.6 构造器函数

当吧函数当做类来用，则名字首字母大写

## 17.7 prototype

-只有函数有prototype