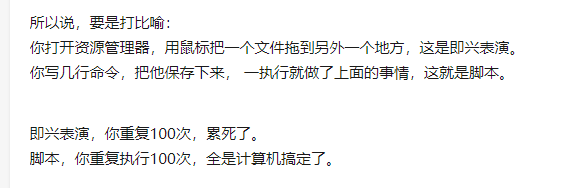
JAVASCRIPT和JAVA的关系：没啥关系

JS由三部分构成

1. language core:语言核心
2. Client-Side:浏览器控制和与user之间的交互
3. Server-side:与Web server之间的交互

Javascript的代码是一种script脚本

脚本是啥：



可以显式的explicitly或隐式的Implicitly嵌入Imbed在html文件中

explicit imbedding(明确嵌入)需要看场合使用，不总是好用

用在<head>里的情况：

1. 脚本影响了user action
2. 被要求这样了（function）

用在<body>里的情况：

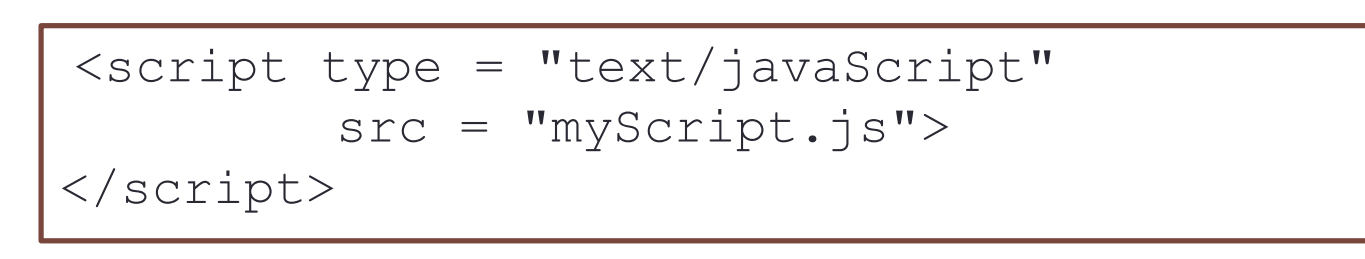
脚本只会被interept解释一次（interpreter找到的时候） 即使用一次

Implicit imbedding(隐式嵌入)在一个独立文件（.js）里

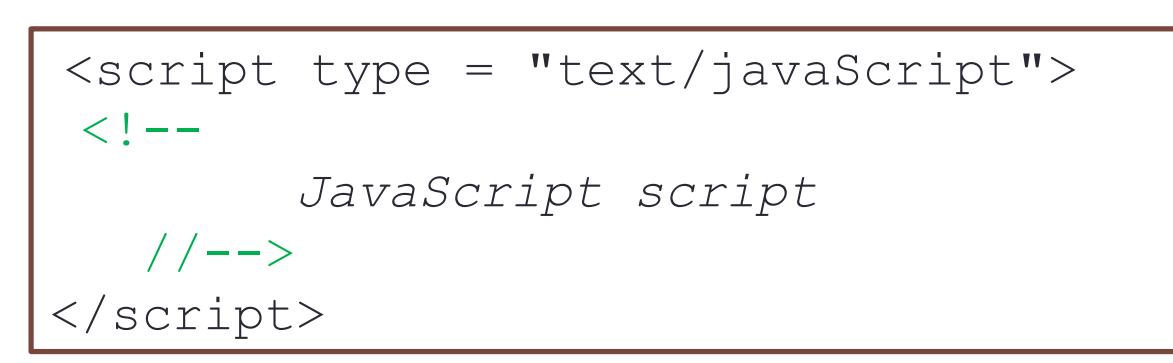
在browser中隐藏了脚本

当JS代码不止作用于一PAGE时使用

语法·：插入JS文件



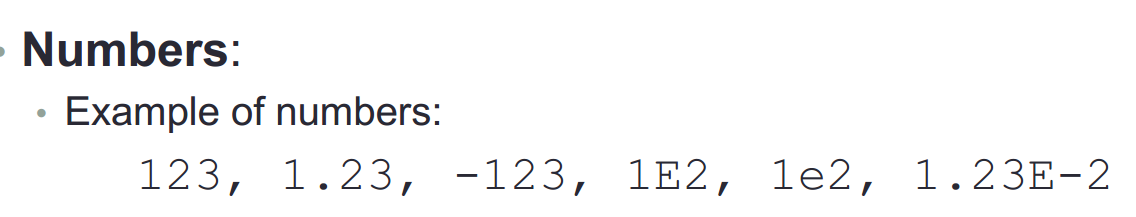
嵌入JS脚本



例子

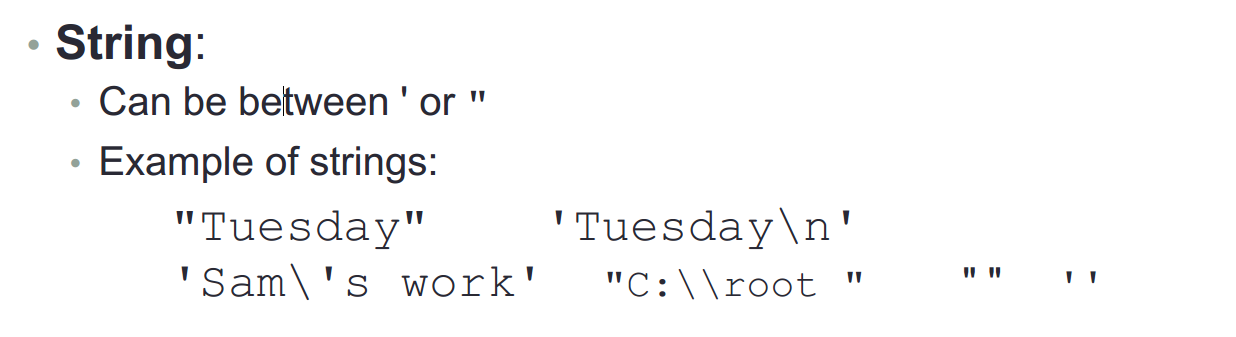


Primitive type数据表示手法



E是十的几次方1E2就是100

String可以写在单引号或双引号之间



Boolean



Null

在JS中，这也是一种数据类型，只有一个值null

Null不能作为任意function or variable的值

一个variable的值是null当他，没有被创建的时候，或者是没有正确指定值的时候

你可以去除一个variable的值，但又不删掉它通过给他赋null值

Undefined:

只有一个值undefined

当一个variable被declare创建又没赋值的时候3

怎么declare(创建)一个variable

Explicit declaration

Var num1，num2=10;

Implict declaration

Num3=10;

注意了这里num1的值是undefined

返回一个最大值

返回一个大于等于x的最小的整数

三角函数

Number Object

是Primitive type的 wrapper



各式各样的property



Number.MAX\_VALUE会得到JS里能得到的最大数，PI会给一个π

一个导致了overflow的数学运算会return一个NaN(不是一个数字Not a number)

比如负数的平方根就是一个NaN

NaN和任何数字的关系都不是==，包括和他自己

+-X/规律

+：左右两边都是数字，加法，不然就是连接符

-x/：数字正常，如果有字符，尽量转换成数字

如果转换不了就会得到NaN

比大小，一个是数字，一个可以转化成数字，正常比大小

一个数字一个无法转化成数字，永远return false

两个都是String,作为String比大小

Number constructor构建一个数字

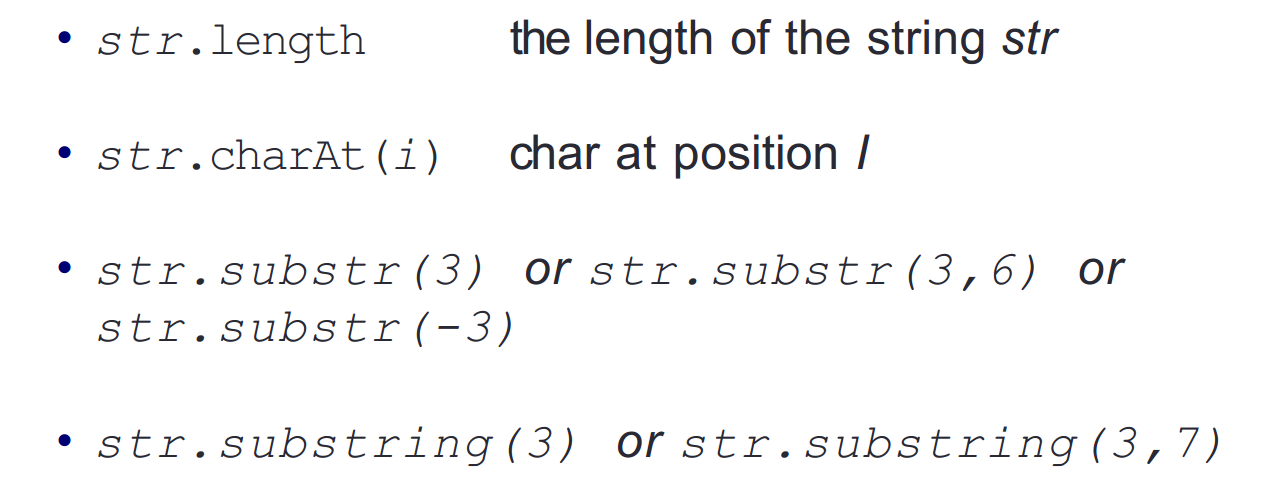
String constructor 构建一个String 

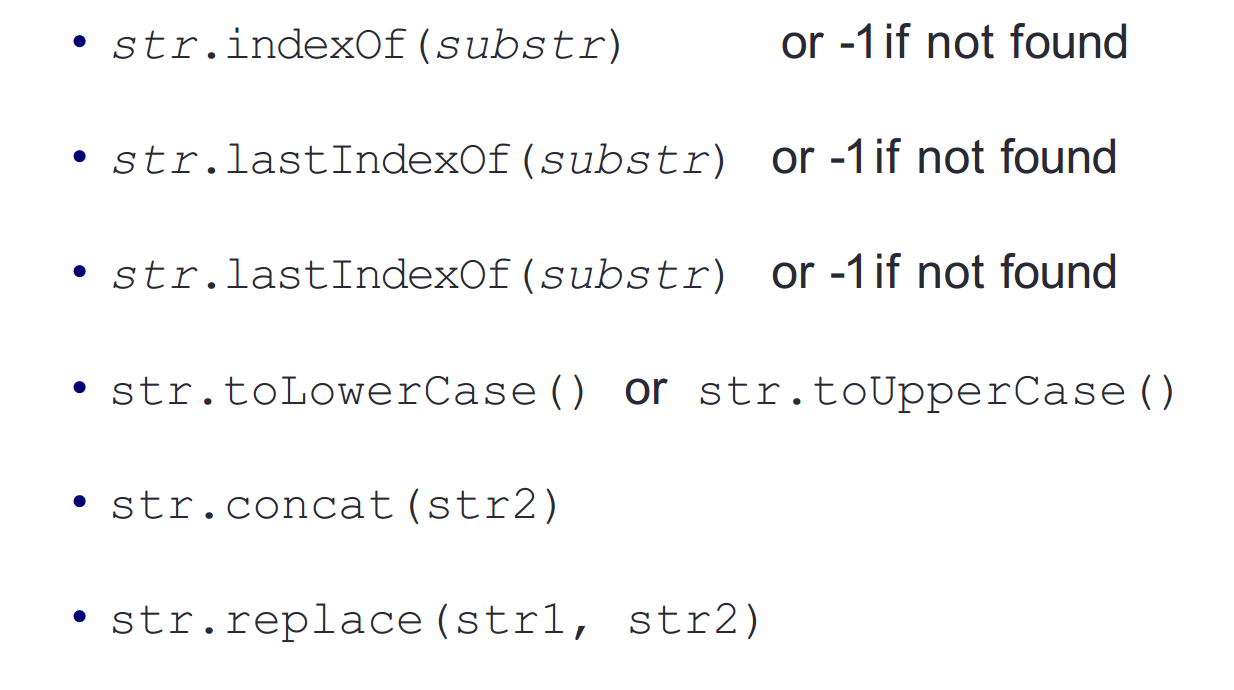
把一个数字转化成String

把String”6”转化成Number

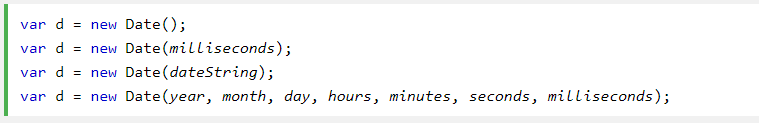
把String”6”转化成Number

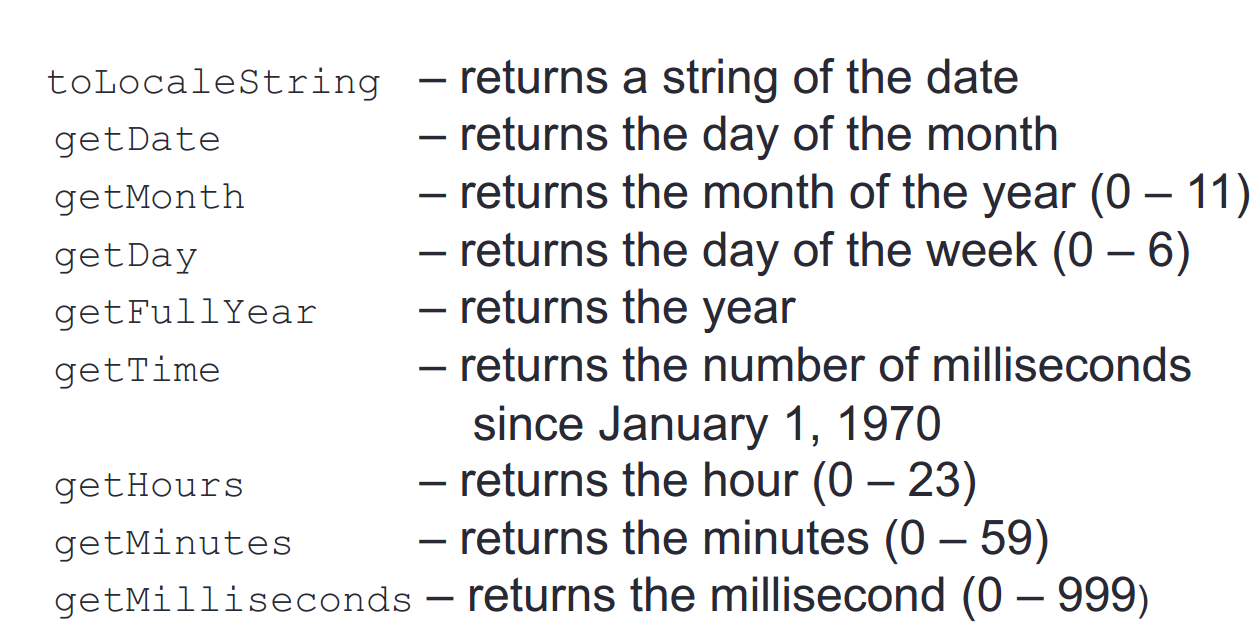
String 在JS中是object,有许多有用的method



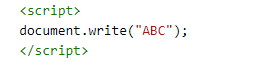


Date object，一种特殊的object





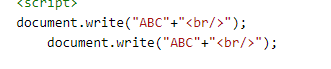
**document** object有一个method叫做write,会在浏览器中显示一些内容

System.OUT.PRINT差不多





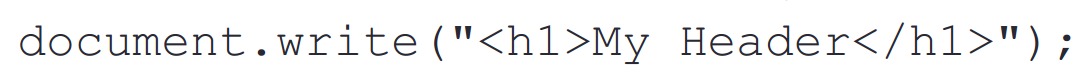
加上br就能空行



不加引号的话就是直接引用

Write method的小技巧

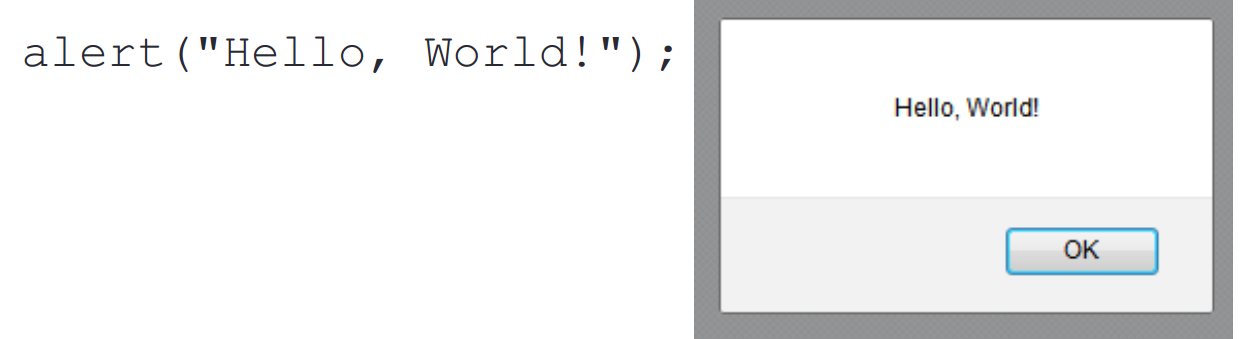
把browser当做一个console然后WRITE看自己写出来的东西输出是什么（其实就是system.out.println）



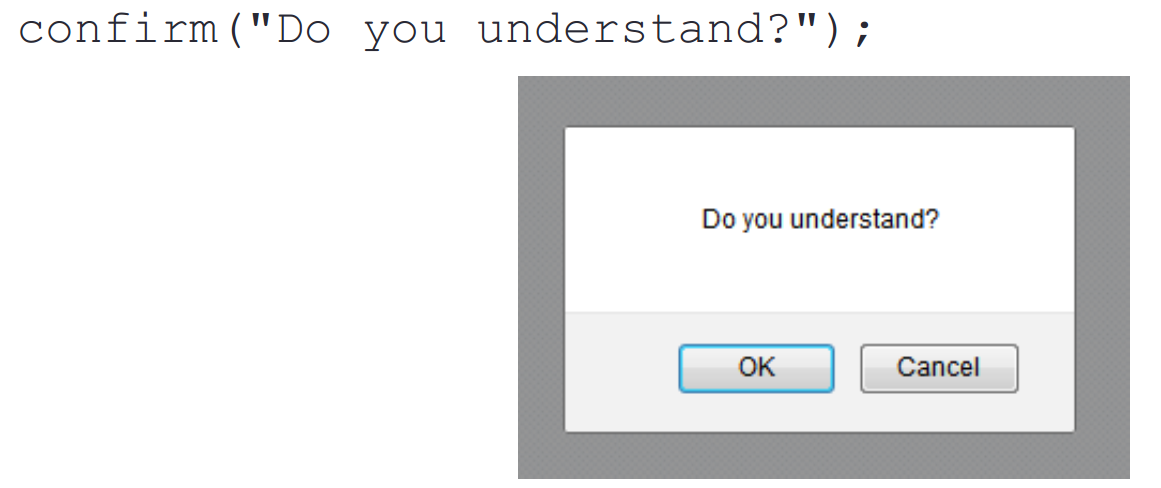
注意要加双引号

INPUT/INTERACTING WITH USER，和用户的交互

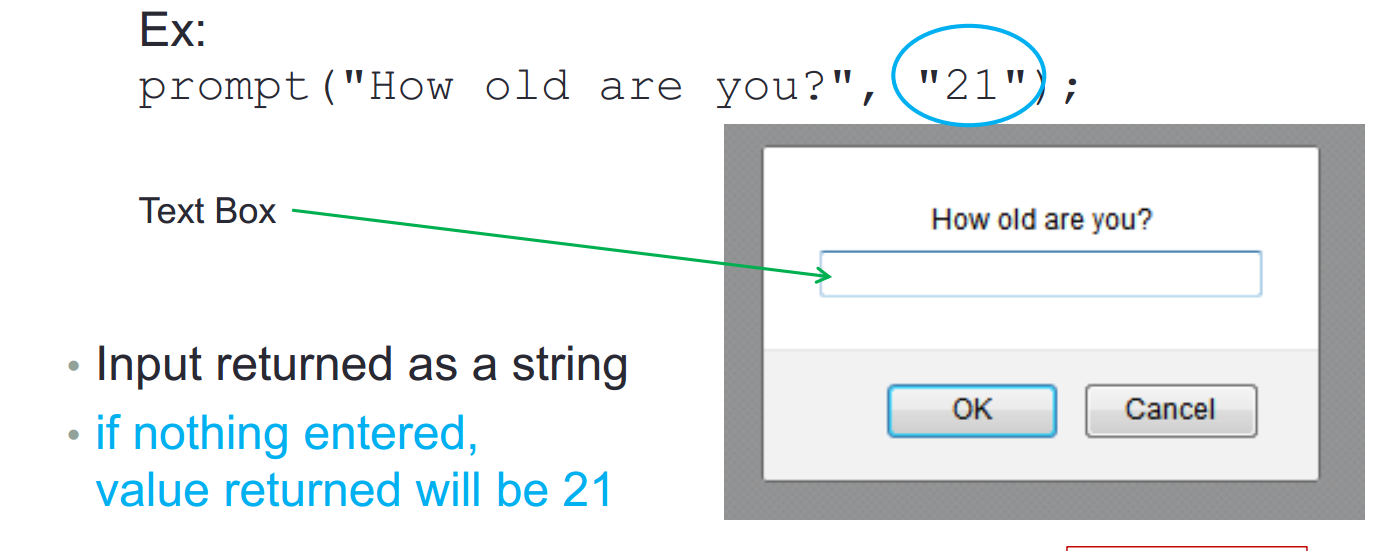
1. alert，跳出一个对话框



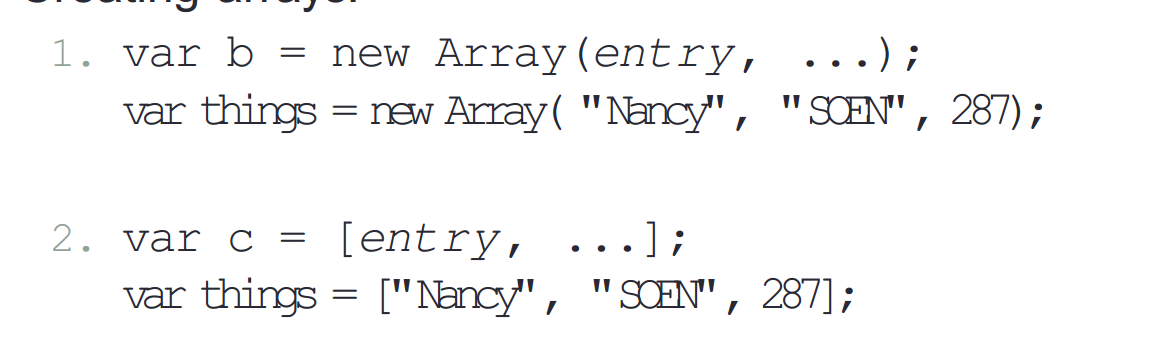
1. Confirm这次有选择



1. prompt输入框

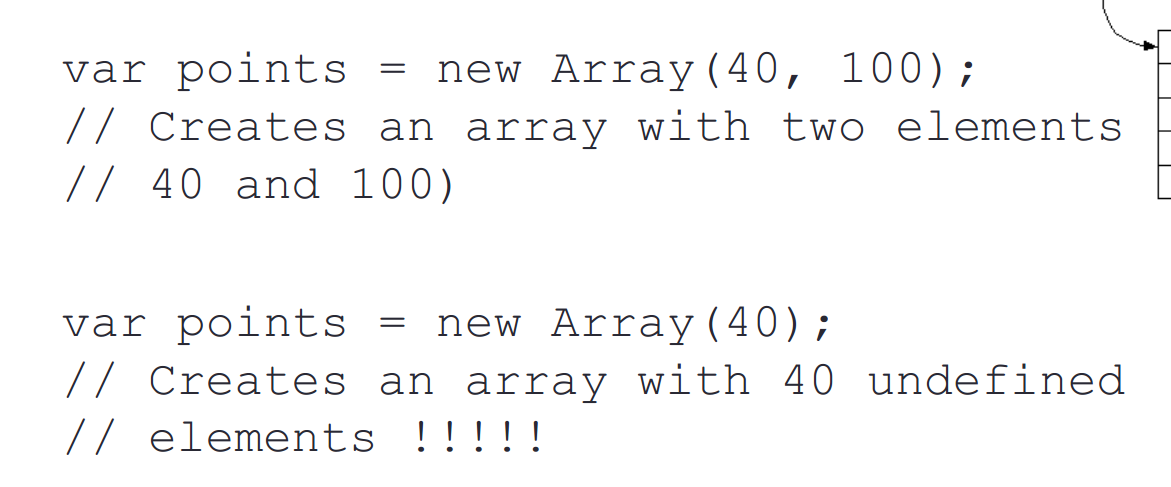


怎么创建一个array

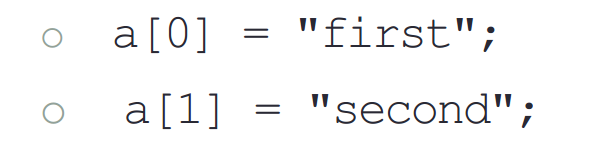


有这两种表达形式

最好用下面这种，因为上面这种有BUG



创造了40个element的array

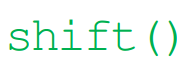
给array中的因素赋值

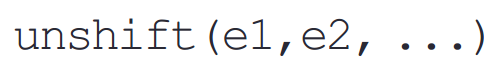
把array中的值赋给别人（注意，没值的element会附一个undefined的值）

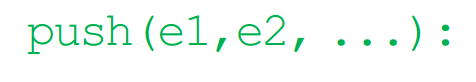
如果array的长度只有6，而我要给一个a[7]赋值，事实上不会出现BUG，而会把ARRAY的长度拓展到8

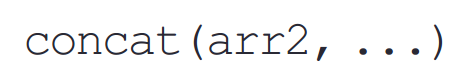
Array中特有的method

取出最后一个element,并将之return

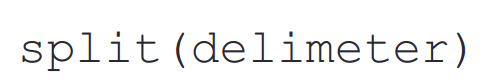
取出第一个element，将之return

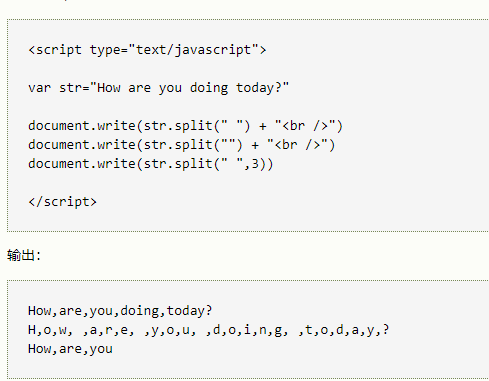
在开头加入e1,e2,e3，然后return新的length

在末尾加入e1,e2,e3，然后return新的length

return一个新的array通过把几个array串起来

：把整个array顺序颠倒

：把一个String分成一串String，分别存入array中



Splice(index从哪开始,howmany有几个被删除,item1,item2)



Howmany0的话就只加不删

Control statement，判断

1.primitive type

如果括号内是String 是true,除非这个String是“0”或empty

如果是Number，是true，除非这个number是0

2.relationship



以上6个符号会强制转换：

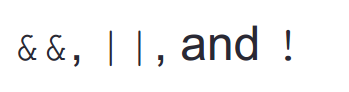
一个是String一个是Number,尽量转成Number

一个是boolean一个不是,boolean转成01

===和！==

不会强制转换，String不会变成Number ，应该尽量使用===

3.compound

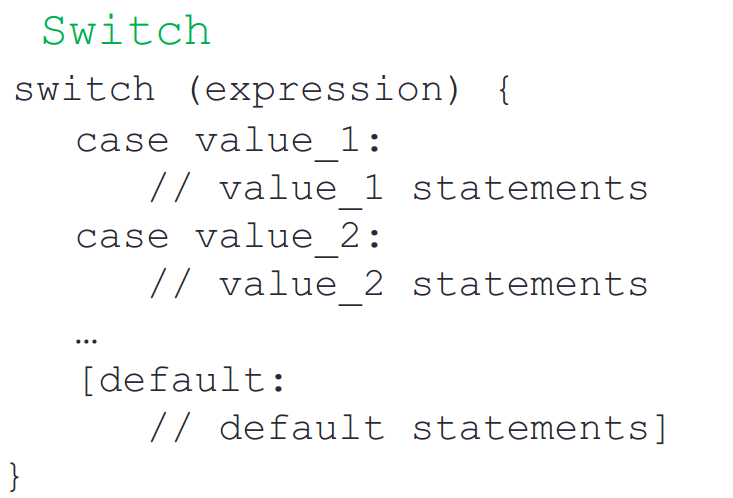


所有的boolean值都可以加上toString()这样就可以Print出”true” or”false”

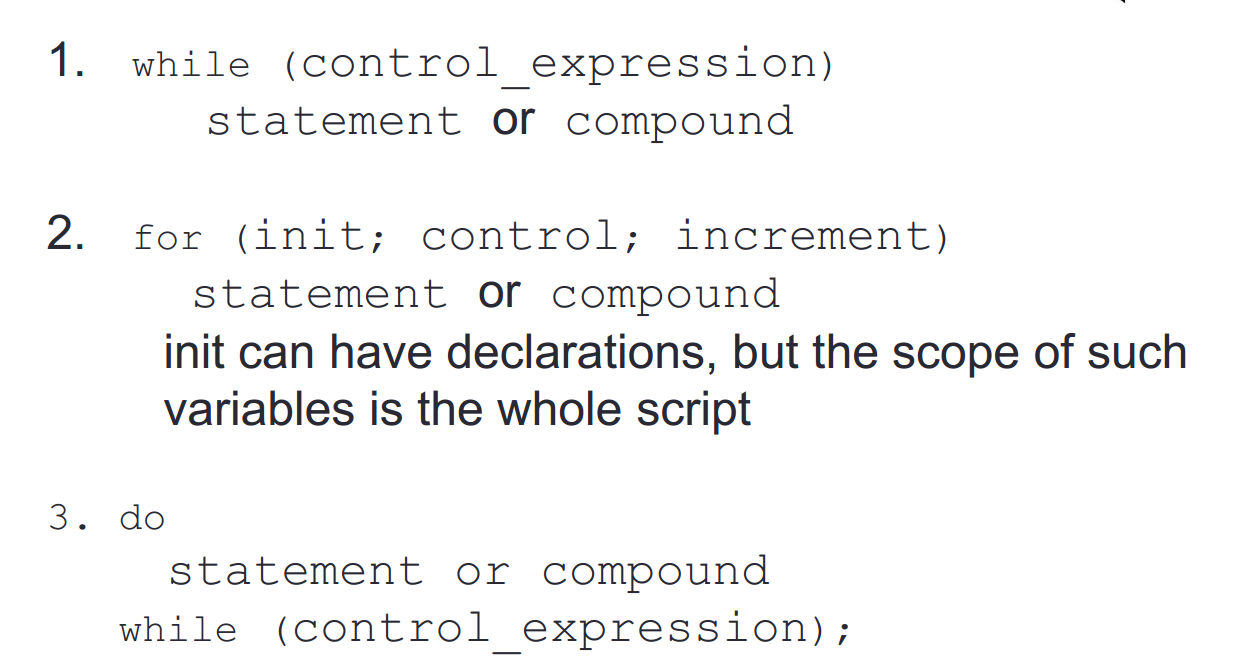
Java 中if else



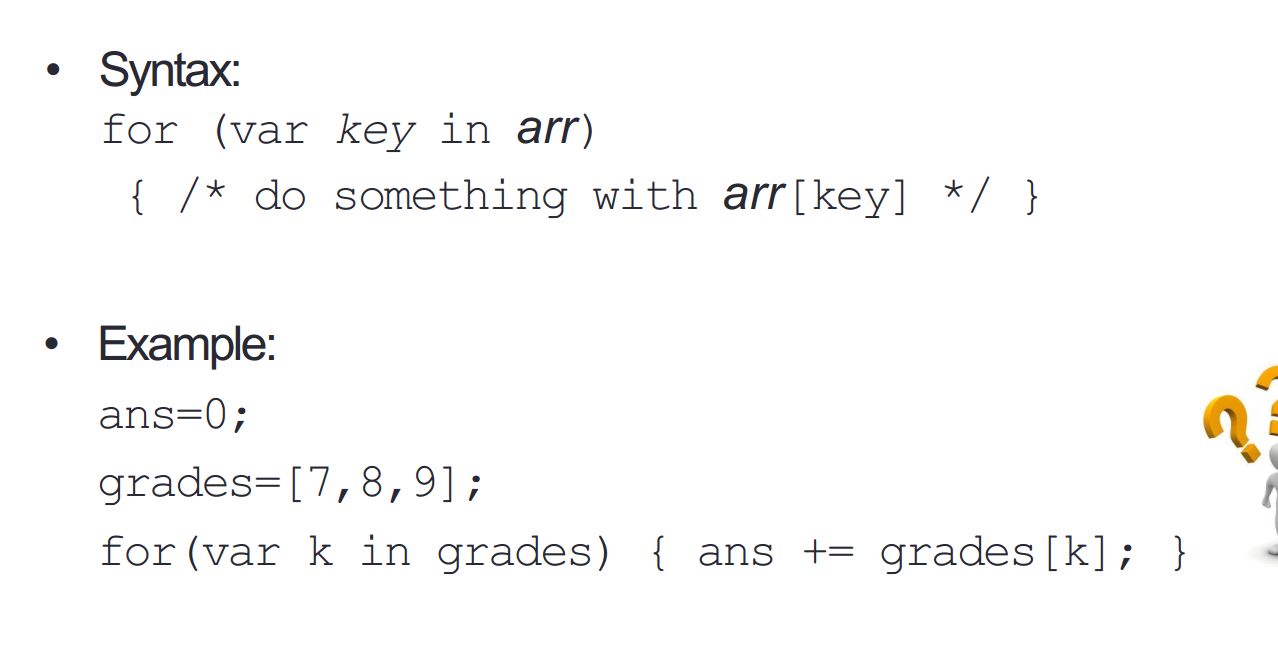
Switch



Expression可以是Number,String,Boolean

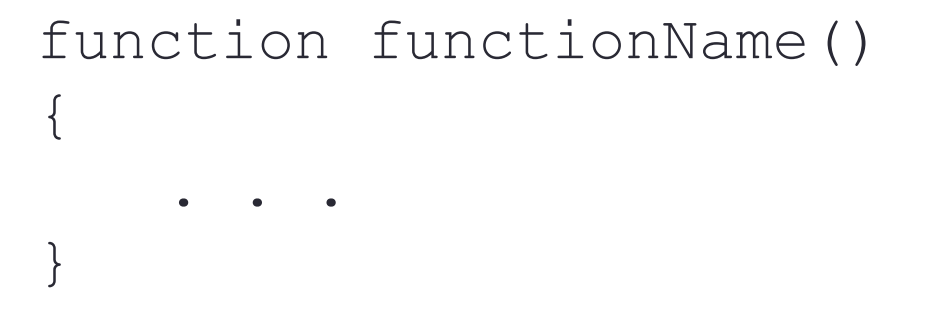


Foreach loop



Function

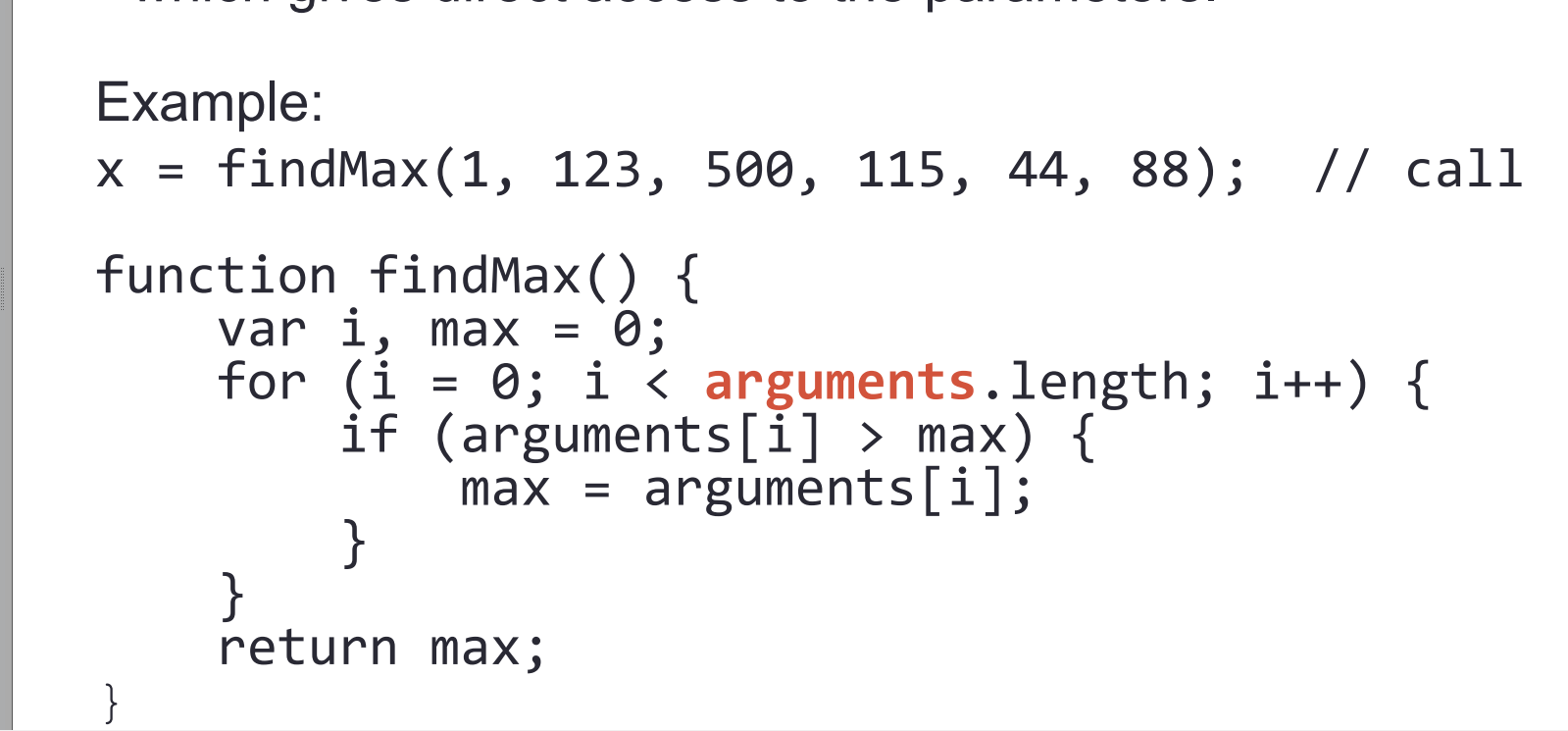
基本格式



Function的参数

1. 对参数不会限制形式（int double等）
2. Function可以有也可以没有参数
3. 不对参数数量有限制（没输入的就按undfined）
4. 值传递而不是参数传递（改变parameter就会改变实际值）

Js function内置了一个叫做arguments的object,这个Object包含了一个array叫做arguments，由所有参数组成



Function最好直接写在<head>里，

Function必须写在使用function的前面

函数内：

全局变量不加var， 有一个global scope （所有人都能使用）

最好别这么使用

函数内变量加var 有一个local scope （只有函数内能使用）

这样其他地方出现相同的Name的变量也不会影响

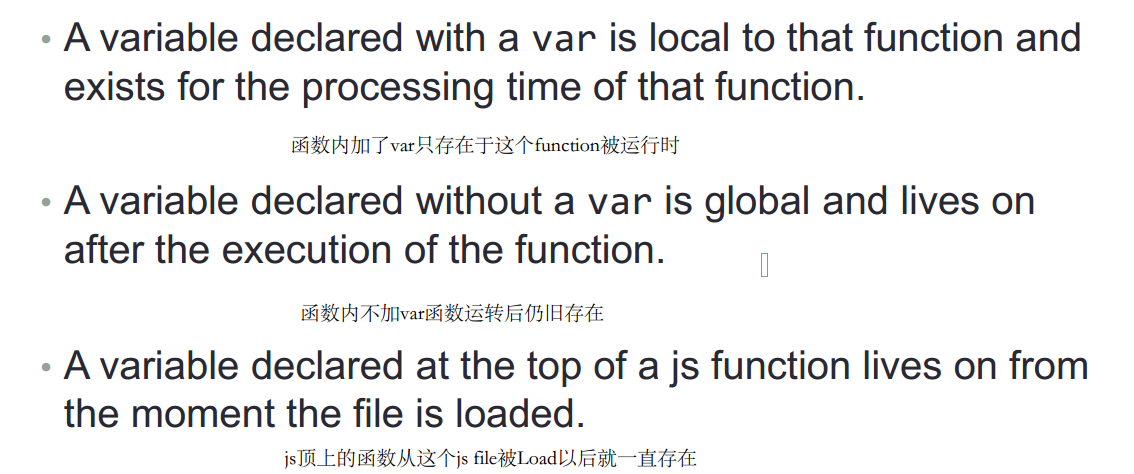
函数外：加不加都是global scope，区别在于加了var的无法删除，不加的可以删除

Variable test

县引用的js会被后引用的Js覆盖

没加var之前，用过一次就会被覆盖，

加varr以后，每次function使用被重置，当然括号外的还是变了



JS的一些点：用的不是JAVA一样的class-based inheritance，而是prototype inheritance

不支持polymorphism

JS的object实际上是一系列属性的集合

和JAVA一样，Object是通过reference进入的

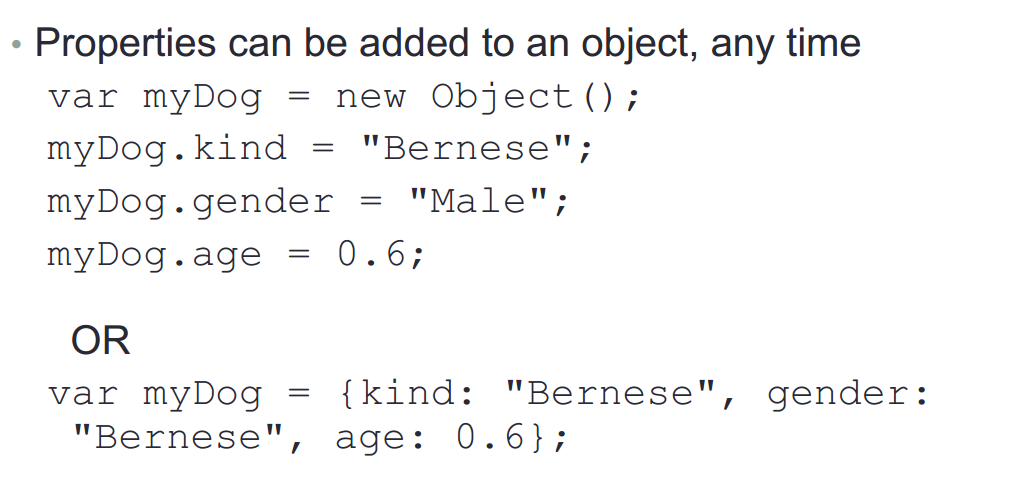
Object可以通过new创建

最基础的blank object

Blank object是没有任何属性的property

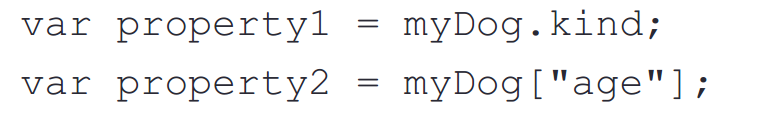
与JAVA不同，property的数量是可以改变的

Property不是variable，只是variable的name



Object可以是nested的，也就是说可以在Object的属性里面再加一个object，通过new

使用object属性的两种方式（.和array中某一属性）

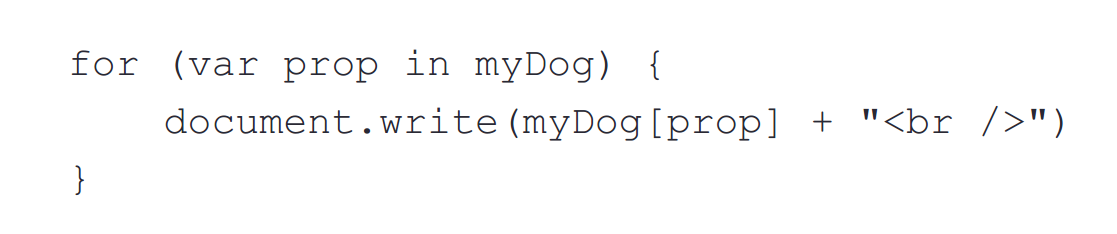


删除某一属性

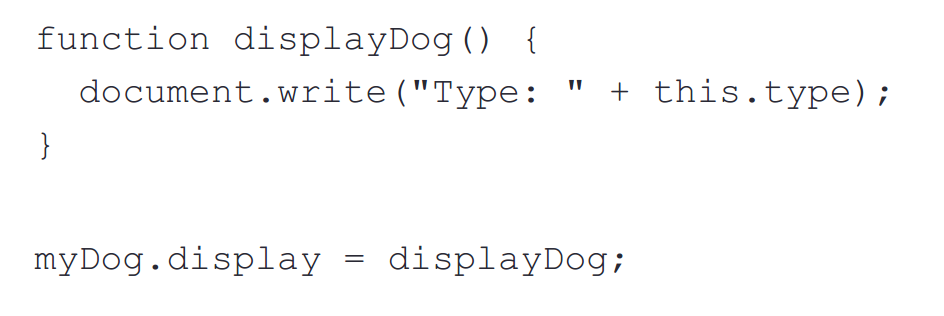


怎么把一个object所有属性打出来

Prop只是随便编的一个Name



Function也可以作为object的属性



注意这个this.type，this指的是这个Object

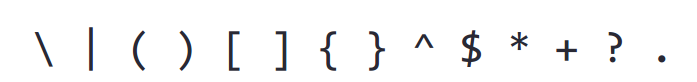
Constructor，用来初始化一个object，实际上是创造了许多属性property



Patter matching

Normal character普通的character就显示自己

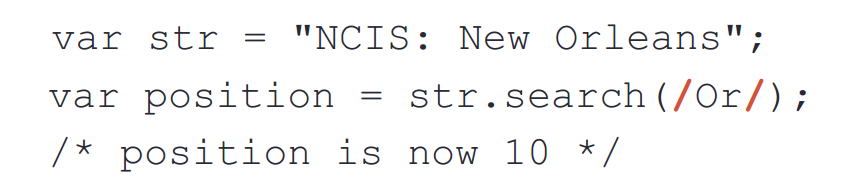
matchcharacter特殊的character 不显示自己



在\以后的matchcharacter会变成Normal character

.(句号)可以匹配任意character除了换行符

Search(具体要search的内容)会return所在位置（从0开始），没search到就return-1



Search的内容夹在//里面

Character class

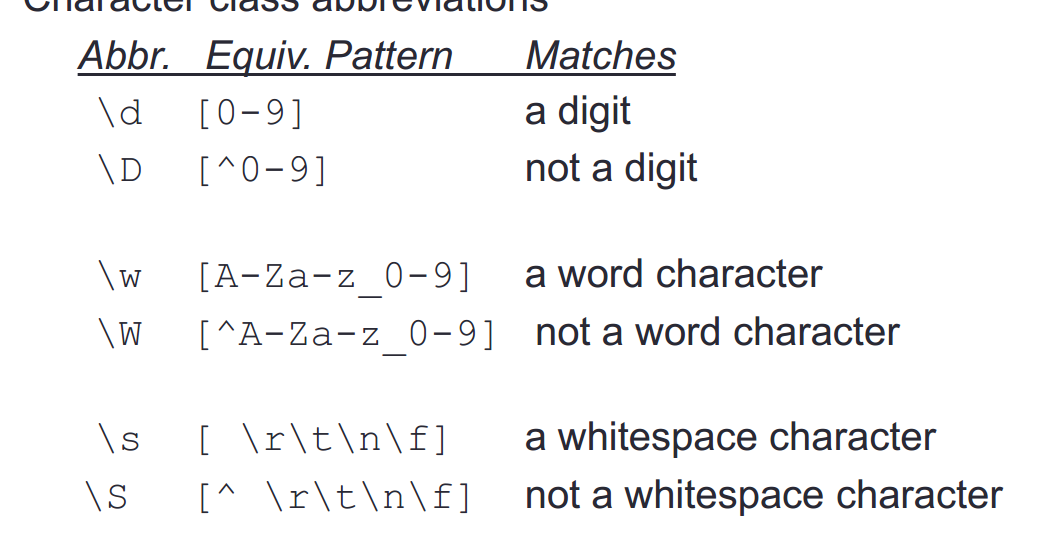
把一串character放在[]中，

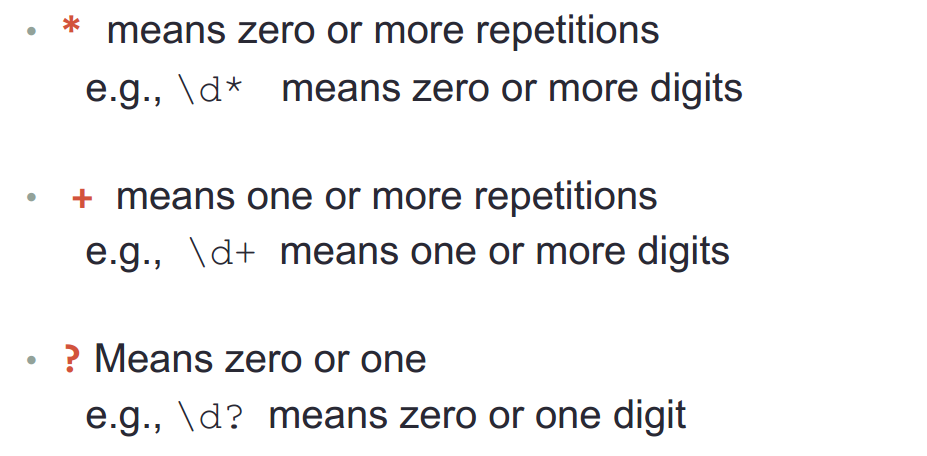
a,b,c中的一个

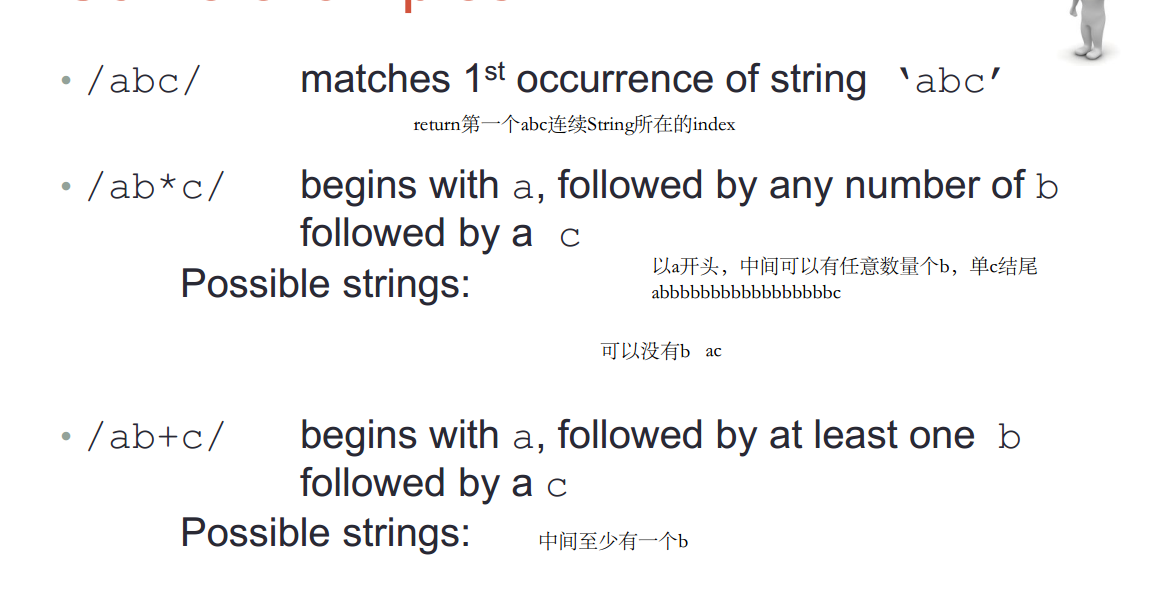
a到z中的一个

不在0-9中的一个

简化写法







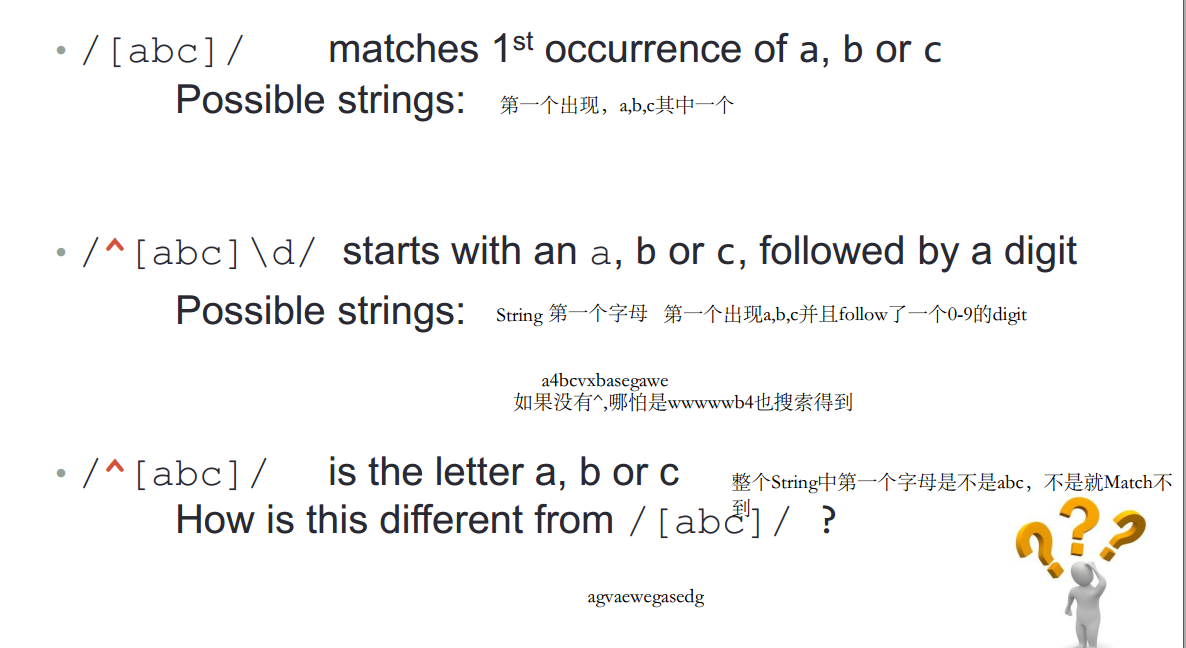


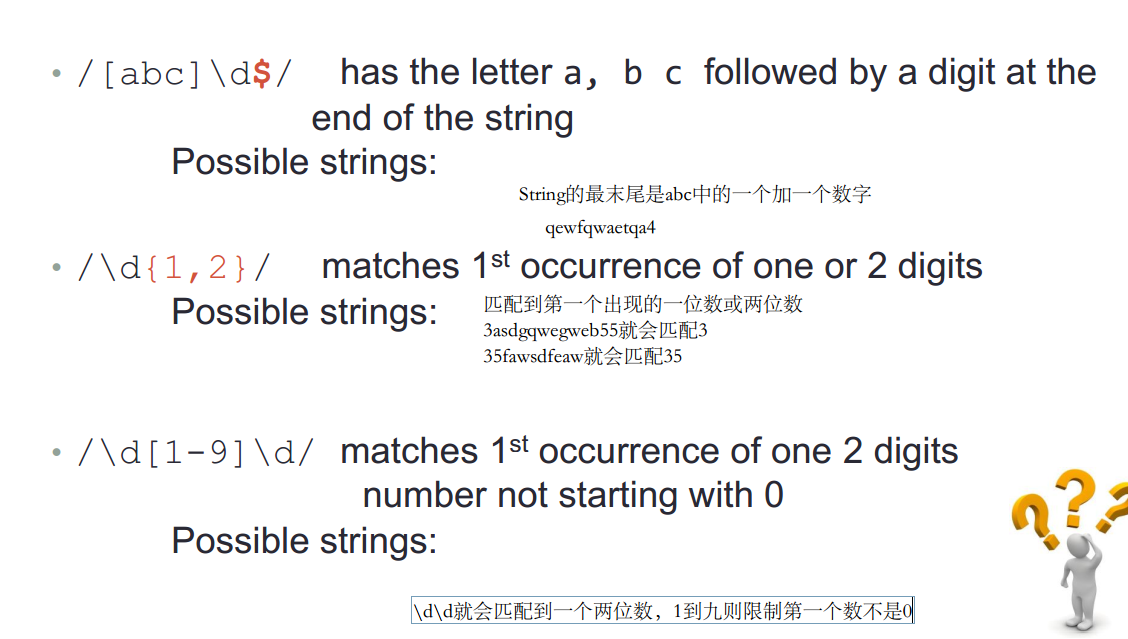
(XX)\*这个XX可以0-无限次重复

（）XX）+至少一次重复

(Xx){一个数字,}至少有几个XX

？只有0or1个





^代表从开头开始算

\d代表着连数字可重复叠加

\d{5,6,7}可连数字个数

\d【1-7】所连数字，

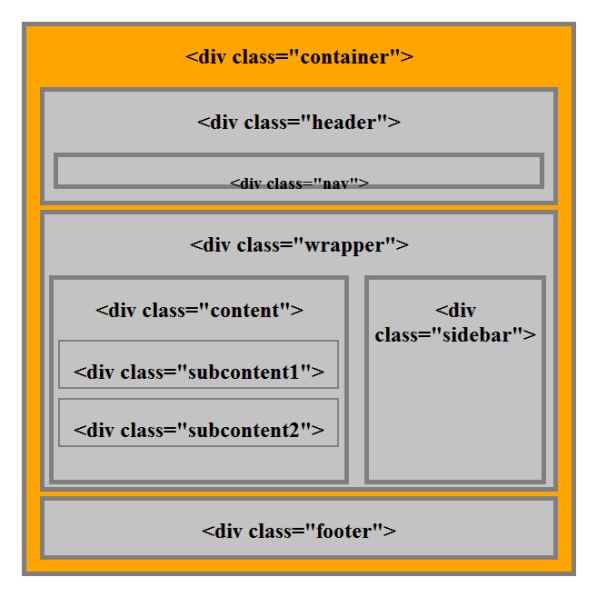
$以什么结尾

Semantic element

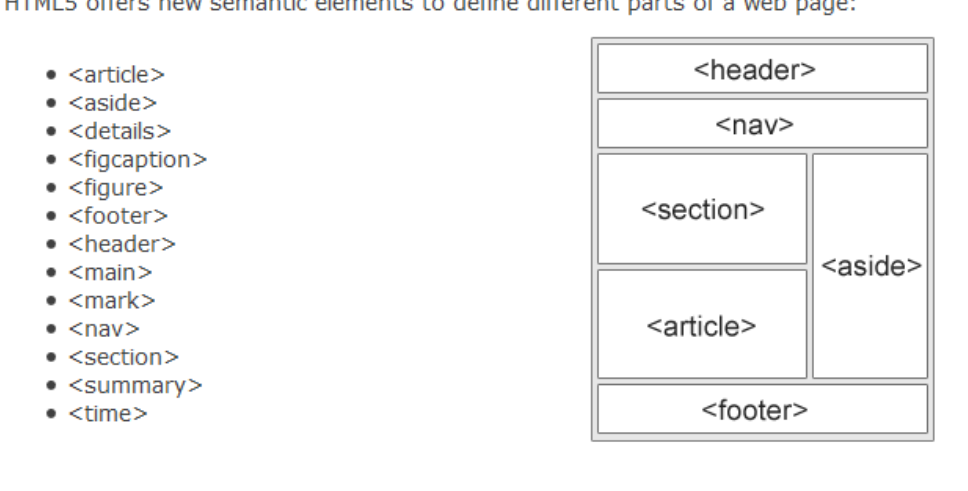
是一个element-把meaning描述给浏览器和用户

在传统的HTML中，需要把class分类





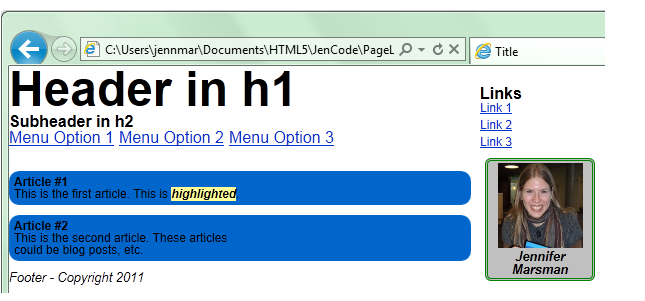
通过semantic element



自带了一个模板

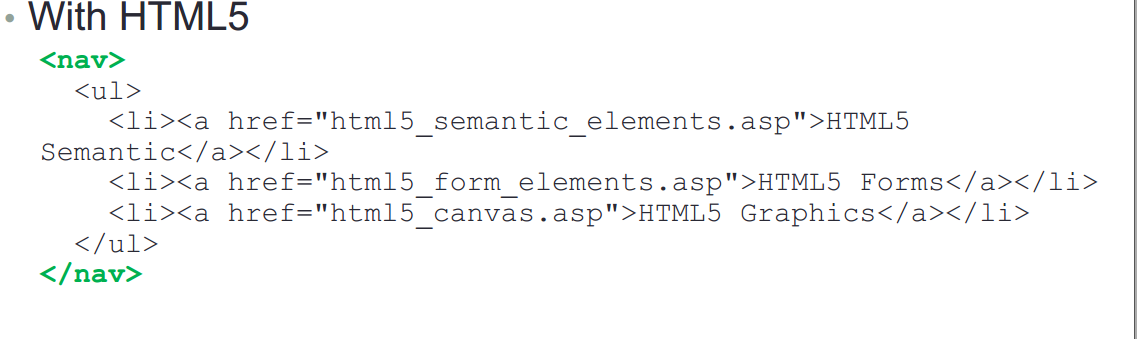


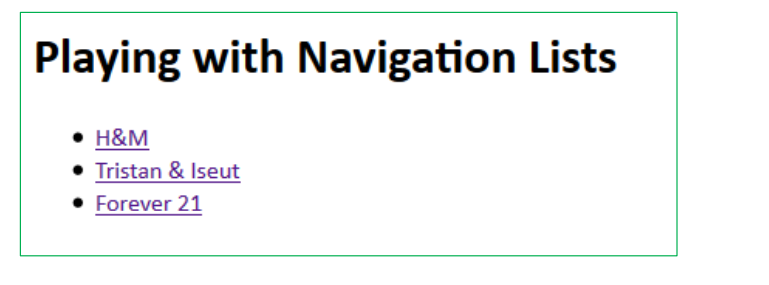
作用在于，清楚地描绘了内容，更好的格式





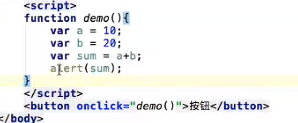
Navigation list





作用将有导航性质的链接放在一起

在<script>外部激活function()



DOM：一种用来改变html element的标准

DOM的Method是对html element的action

DOM的property是html element的value——你可以设置或改变

在JS里

HTML里的element元素被认作是object, element变量attribute被认为是property

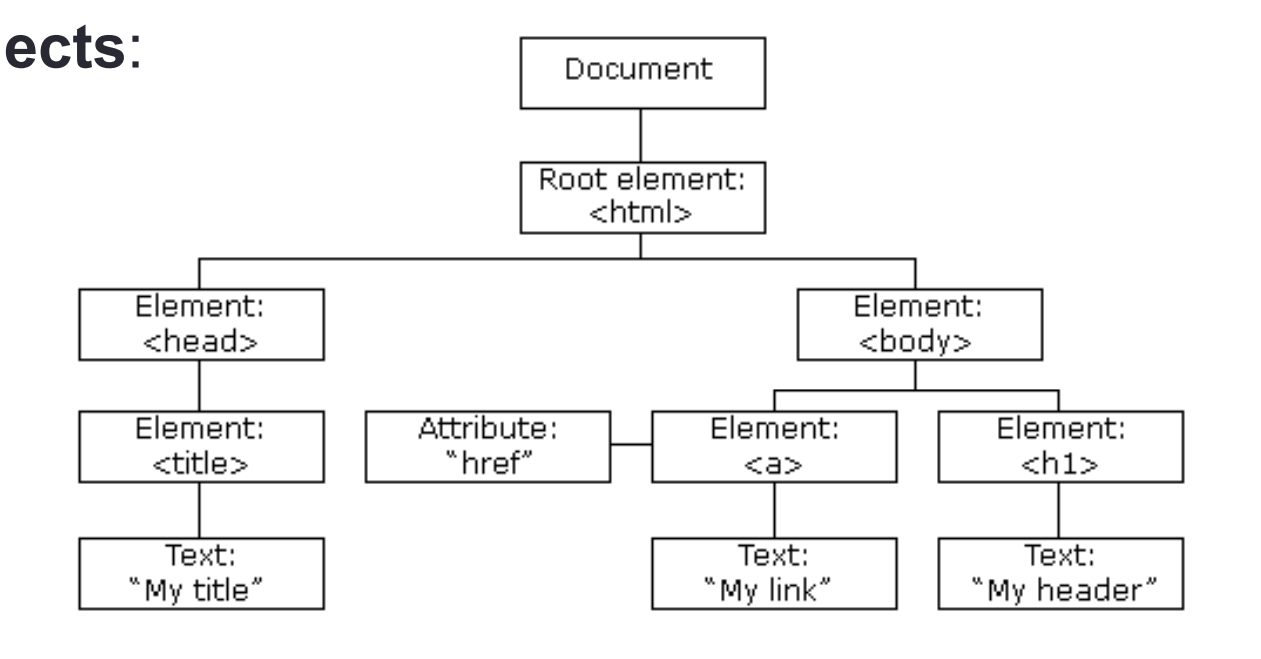


蓝色的是object，绿色的是property，红色的是value

Input元素，各种属性property，name,这些属性的值是……

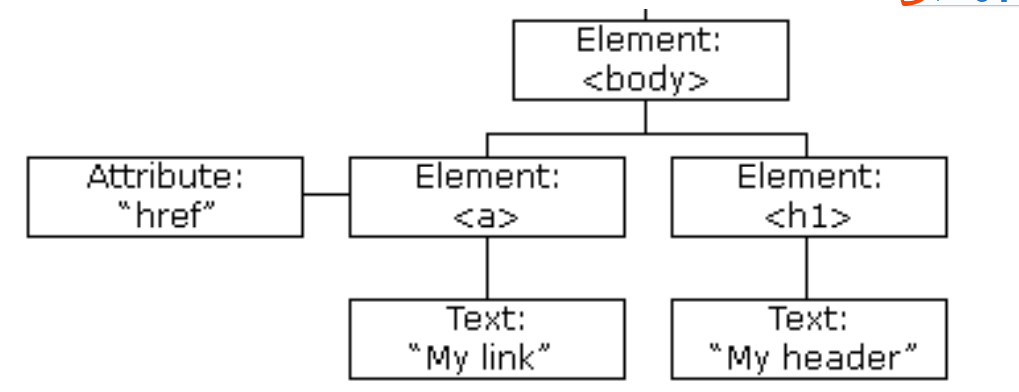
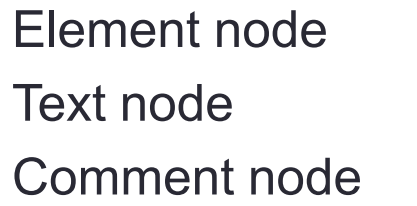
当一个网页被读取完毕的时候，浏览器会为这页创造一个DOM（document object model）

这个HTML DOM model是作为由object构成的树图被创建的

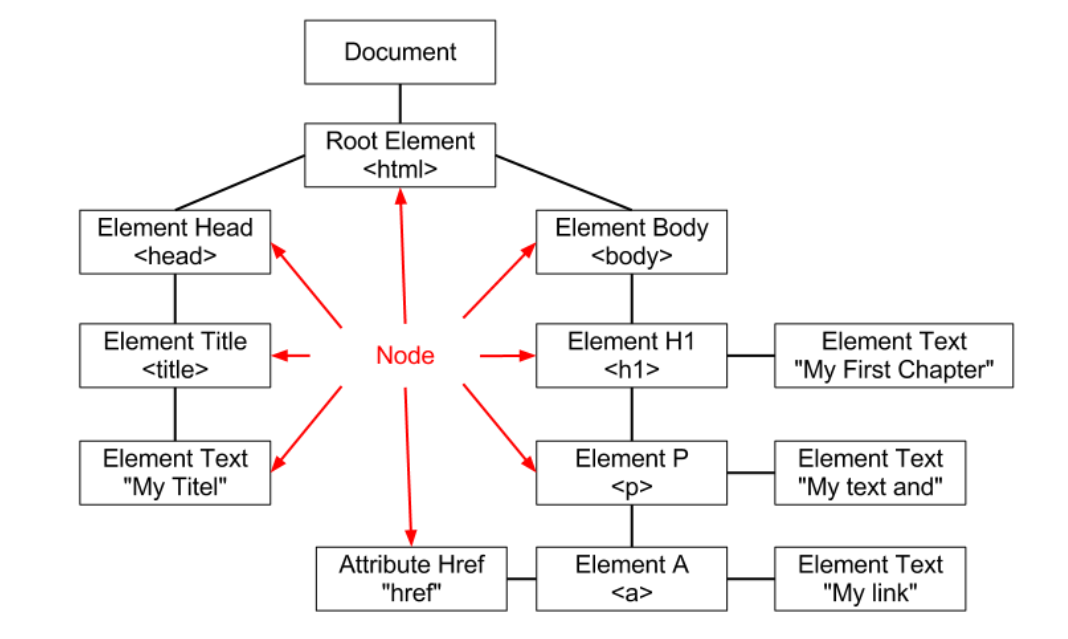


Element object:就是html element

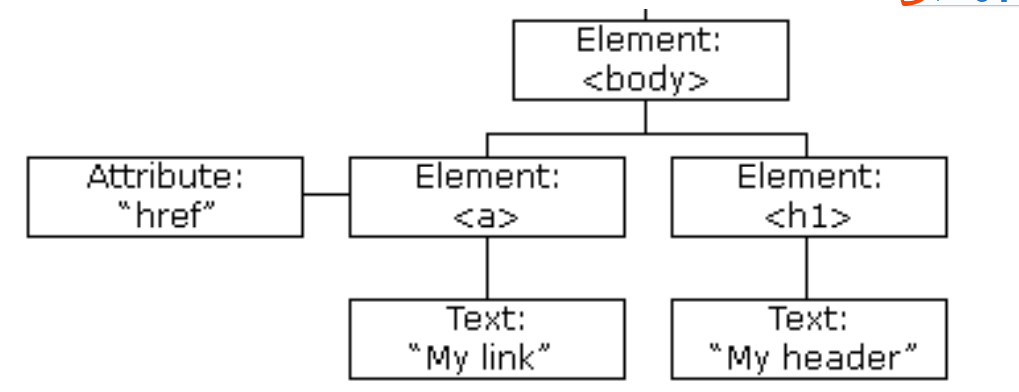
每个element object都有child nodes子节点：



Element也可以具有attribute: attribute node



对于

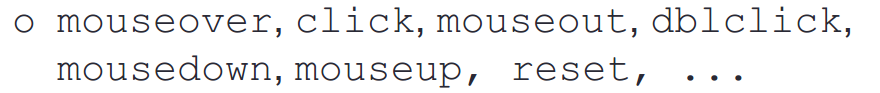


那么我们怎么告诉系统input 两个text中的内容并进行所需操作呢？

对于body element



对于button element



对于text box和password element



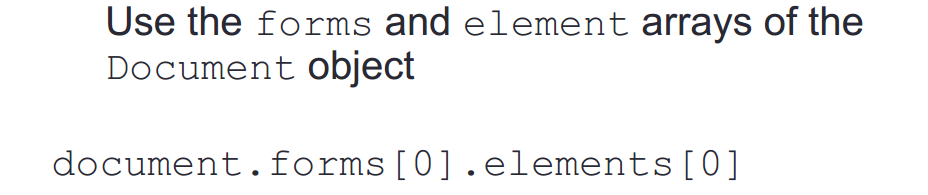
Event 是一个提示，当一些特殊情况发生的时候，包括浏览器或者操作者的action

Event handler是一种隐式implictly进行的script脚本， 作为对于event出现时的回应

把EVENT和Event handler相连的过程叫做registration注册/挂号

怎么进入element

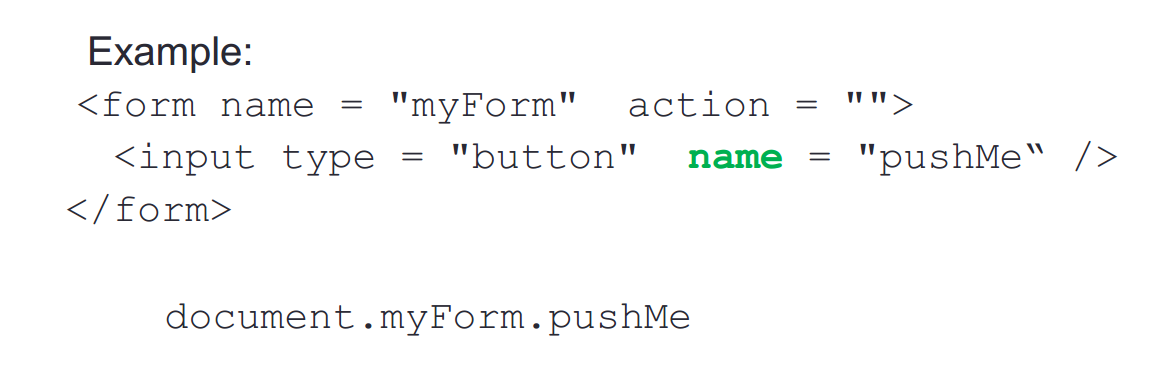
1. DOM address



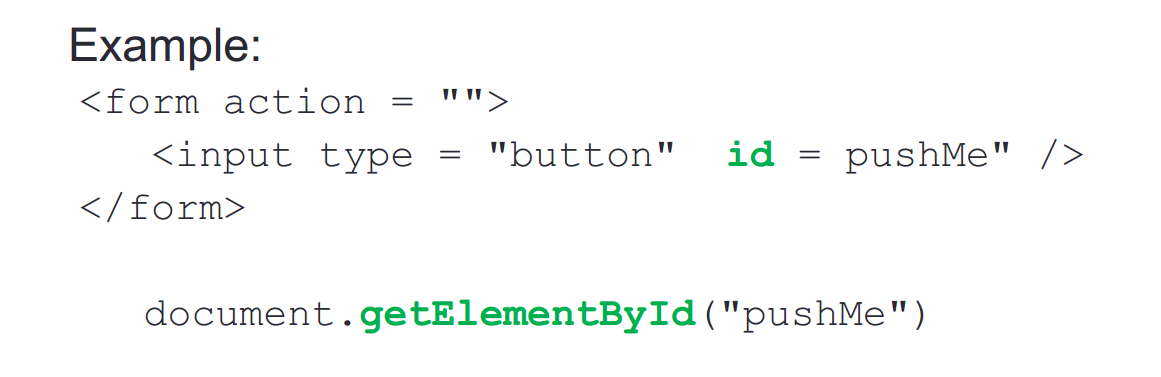
导致的问题是document被改变了

1. Element name

需要element 和所有祖类ancestor（除了body）都有Name这个attribute

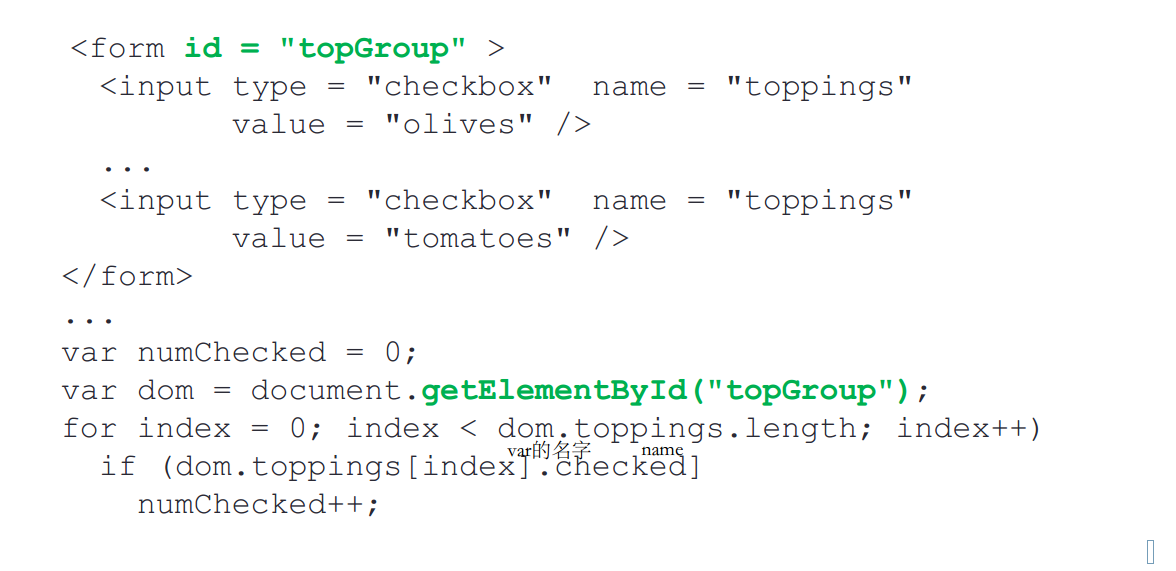


1. getElementById method

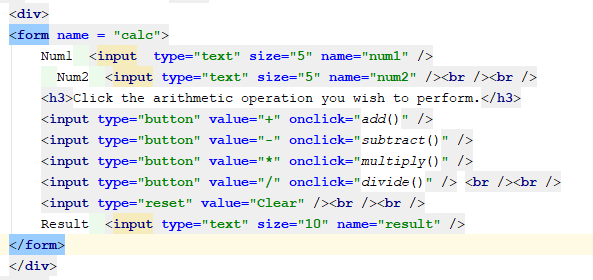


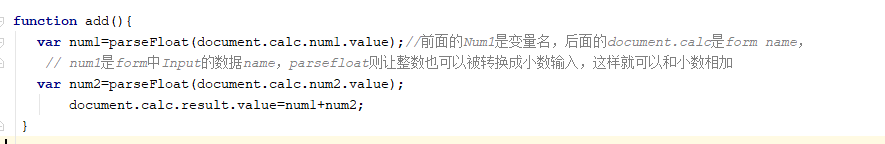
对于checkbox和radio来说

他们是有一个隐式的array的

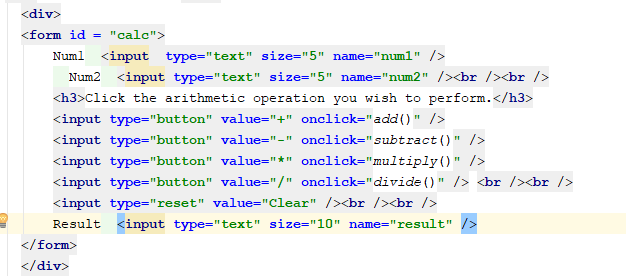


一个name的实例

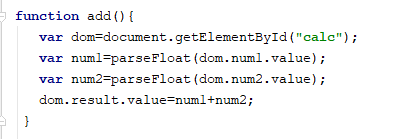




Id的实例



其实也就是个form加了id



Dynamic html

到现在为止我们的Html文档是静态的

一旦在浏览器上显示，他就不会再变，直到操作者进行了一些动作，比如鼠标移到按钮上

一个dynamic html，在dynamic html 浏览器展示时，他的tag，内容等等，是可以随时改变的

Positioning elements

CSS-P 1997年释放，

任何元素的位置position元素都有三种style属性

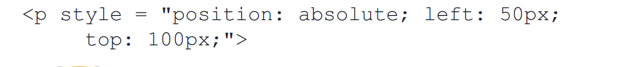
三种可能的值

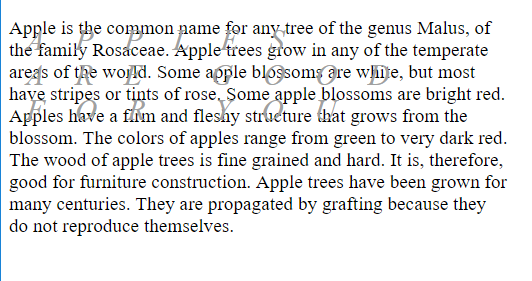
Absolute positioning

1.当我们需要把一个element放在一个特定的地方

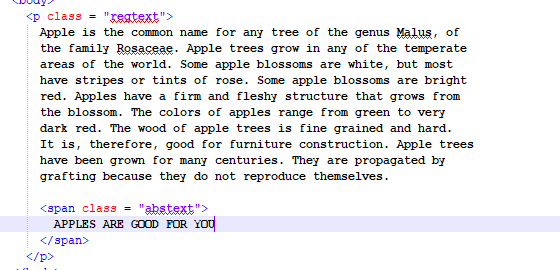
2.与其他元素的位置或序列无关

比如：用来在段落中添加文本，像水印一样



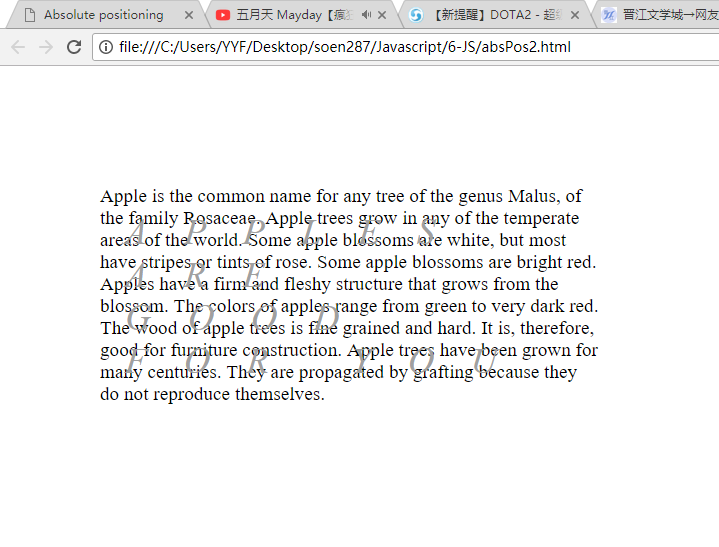


如果absolute element 被包括在另一个element里面，那么他的top和left属性是相对于包括他的那个element的



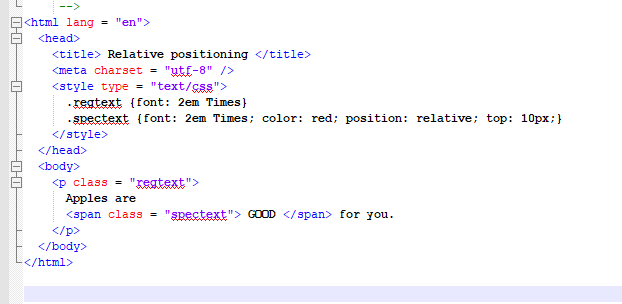
Span是在p里的

这个是相对于包括他的文本的边框的



Relative positioning

设置top和left属性，他会出现在指定的地方，





指定的top是10px，所以会往下10px

如果没有top 和left属性，那么element就会被放置在没position属性时候所放置的地方（默认位置）

但在将来可以移动

Static positioning

Static positioning是position这个属性没指定时的默认值



