Javascript整型数据定义：

可以是正整数、负整数和0

719 //十进制

071 //八进制的71（以0开头代表8进制）

0x987 //十六进制的987（以0x开头代表16进制）

\r代表回车符

Js中的string操作

Substr(start, [length] )

Substring( from, to)

Js调试，使用window.onerror,必须放在页面中所有js代码的最前面

<script>

Function myerror(\_message,\_url,\_line){

Alert(“错误信息:”+\_message

+”\n错误的URL：”+\_url

+”\n错误的行数：”+\_line

);

Return true;

}

Window.onerror = myerror;

指导在页面html加载结束后再调用这个js代码

Window.onload = function()

{ func1()

Func2()

…

}

Func1，func2的位置不作要求（可以在onload之前也可以在onload之后）

函数(特殊的对象类型)

1.命名方式一：function a(){…}

命名方式二：var a = function(){…}

(注意：此种方式中a代表的是这个函数，而不是函数的返回值)

2.参数:js中所有函数的参数都是可选的，如果调用时函数参数表为空，也是合法的但是返回值不一定合法。

3.若调用函数时输入的参数个数大于规定的参数个数，函数仍可以运行，忽略规定参数个数之后的所有参数

数据类型：

6个基本数据类型，一个对象类型

对象类型：name:value pairs（passing by reference）

基本数据类型：boolean

(passing Undefined（缺少值）:此处应该有一个值，但是还没有定义

by Null（没有对象）:一个值被定义了，定义为“空值”

value) Number：64位浮点

String

Symbol

===:严格相等（需要比较类型） “4”===4 false

==：值想等（不比较类型）”4”==4 true

对于布尔类型，

Boolean(false || null || undefined || “” || 0 || NaN) 返回值都是false

If(…||…||… … …||…)从左到右判断T/F，只要碰到第一个值为true的值，则停止判断，执行if之后的语句

Var x = “”||”hello”;

X = “hello” (默认选择第一个布尔值为true的变量的值)

Js中的构造函数:

Function Circle(radius) {

This.radius = radius; js中的构造函数没有返回值，否则返回一个空对象

}

Var myCircle = new Circle(10);

Console.log(myCircle);

输出为：Circle {radius:10}

某个类构造方法之外的方法

Circle.prototype.getArea =

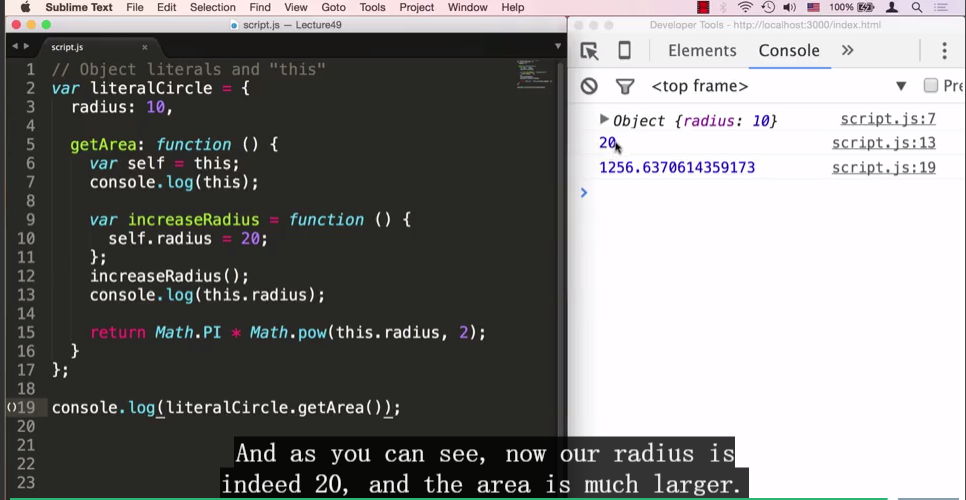
Function () {

Return Math.PI \* Math.pow(this.radius, 2);

};

Var Circle = new Circle(10);

Console.log(circle.getArea());



在函数increaseRadius中，其中的this指代的是window,而不是literalCircle

只有在increaseRadius方法之外先把this赋值给一个新的变量，之后其中才能获取到literalCircle的this值

Array

1.js中的array没有类型限制

var array = new Array();

var names = […, …, …, …];

可以是各种数据类型（对象+基本）+函数

2.array.length = array的长度

3.中间可以间隔不定义 如 array[0],[1],[2]都已定义，之后来个array[100]定义，也是合法的，未定义的部分看作undefined

4.遍历

①for( var I = 0; I < array.length; i++)

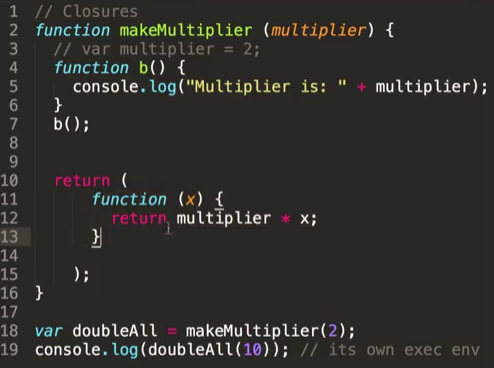
②for( var prop in array/Object)

For array: console.log ( array[prop]);prop看作下标

For object: console.log( Object[prop]);

Array是特殊的object

Closure



doubleAll代表的是返回的function(x)

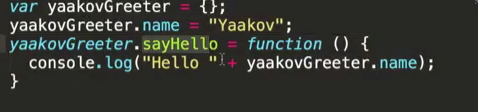
为什么这个function可以知道原先的multiplier？

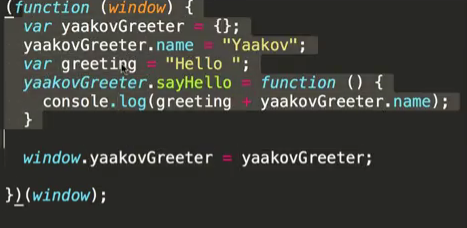
Js中内函数的作用域会保存其外部函数的变量

Js中的导入

在html中导入js文件，js文件中的全局变量导入到global namespace中，若几个js文件中的全局变量名重复，则以最后一个同名变量名的值为准

(若想避免这种情况，可以给每个js文件中的变量添加一个持有对象)





(IIFE Immediately Invoked Function Expressions)

文档

Document.getElementById(“…”).value获取对应id的值

Document.getElementById(“…”).innerHTML=… 向对应id处插入html文本

Document.querySelector(“#title”) 以css选择器的样式选择文档内容，可选择id，class，标签等多种

事件响应：

Document.addEventListener(“DOMContentLoaded”, function(event){

…

}

这段js代码表明js将在网页加载完成（在外部链接和图片加载完成之前）执行

GET

POST

JS中的response状态码

HTTP/1.1 200 OK : HTTP-version Response-status-code English-explanation

200 OK : 获取对方要求，并且拿到了对方所要求的数据

404 Not Found: 服务器找不到对方所要求的数据

403 Forbidden: 未认证客户尝试连接安全连接

500 Internal Server Error: 服务器发生未知错误

Ajax

Asynchronous(异步的) Javascript And XML

Ajax是异步执行的，即sendGetRequest的方法在执行中也会执行同级的其它代码，不能保证在ajax代码之后的代码就是在其后执行的，此时要把所有需要保证序列行的代码都加入到ajax中

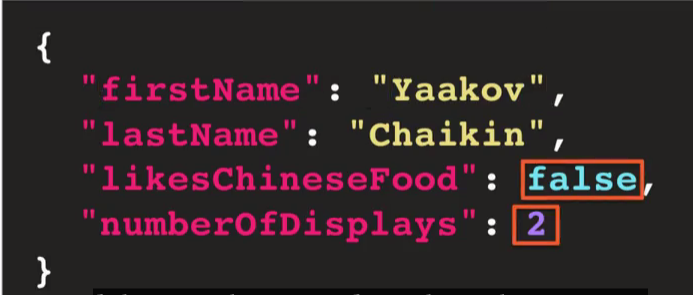
JSON

JavaScript Object Notation

轻量级数据交换格式

JSON的语法与js中的对象类型很类似，但是它实际上还是一个字符串

语法：属性名和字符串的值必须用双引号引起来



Converts from json string to object

Var obj = JSON.parse(jsonString);

Converts from object to json string

Var str = JSON.stringify(obj);