1. 除了以下四种原始数据类型，Javascript中的一切都是对象

原始类型：String;Number;Boolean;Undefined;

获取变量的类型：

typeof(“变量名”) 可以得到变量的类型；

“变量名” Instanceof Object 某个变量是否是某对象的实例

1. var num = 100;

var obj = new Number(100);

全局变量：window

Window.parseInt(“34”,10);10进制

全局名可以不写，parseInt(“23dfs4b”,10);23

NaN:这不是一个数字

动态类型的编程语言：类型可以随便改

1. 变量的判等问题；null和Undefined

var x = 42;

var y = ‘42’;

x == y True 弱类型的判断；只判断值，不判断类型

x===y False 判断值和类型必须都相等

var year; undefined

year === undefined true

var obj = null; 不引用任何对象（有值）

typeof obj ‘object’

obj == null true

obj === null true

var myvar;

var myvar2=null;

myvar == myvar2 true

myvar === myvar2 false

1. 函数

定义的方式

1. 命名函数(全局,在函数定义的前后都能调用)

function add(x,y){

return x+y;

}

1. 匿名函数(适合定义并立即使用，只能在定义的后面调用)

var square = function(x,y){

return x+y;

}

1. 立即执行的匿名函数（定义并立即调用）

console.info(

function(x,y){

return x+y;

}(100,200)

)

1. 函数的返回值问题

return 后面不能跟回车；

因为return后面如果没有检测到代码的话编译器会自动加上分号

1. 函数里面还可以再定义函数，函数嵌套

变量的访问方式 ：内部函数可以访问到外部函数的变量；

1. 返回值为函数的函数（实际上返回的是函数对象）

调用函数后再调用一次

1. 参数的问题

传多或者传少参数都可以执行函数

function sayHello(name,message){

//判别是不是undefined,是就赋值

if(typeof message === ‘undefined’){

message = ‘您好’;

}

}

arguments:参数对象，可以用此属性判断调用函数的时候传递的参数个数（实参）

函数名.length:表示形参的个数（形参）

1. 事件监听

var button = document.getElementById(“btnClick”);

var result = document.getElementById(“result”);

button.addEventListener(‘click’,function(){

clickCount++;

result.setAttribute(‘value’,clickCount+”:Hello world”);

})

1. 变量作用域

**1.函数内没有作用域块这个概念，只要是函数内部定义的变量，该变量的作用域都是该函数；无论里面覆盖多少层ifelse**

**2.变量提升功能：假如定义并复制的变量在后面，则编译器自动会变成定义在前，赋值在后**

1. Javascript对象 || &&
2. 先创建一个空对象，然后再给空对象的相关属性赋值；
3. 利用匿名对象的方式创建对象；

对象的属性可以是任何对象；

Exampleobj.notExist || ‘默认值’:如果Exampleobj没有notExist属性的话就设置为’默认值’

exampleObj.sayHello && exampleObj.sayHello(‘jxl’):如果前面那句为false就不会执行后面这句

1. 利用工厂函数创建对象；

在工厂函数里面创建一个对象，再给该对象的相关属性赋值，再返回创建的对象；

1. 利用构造函数创建对象；

在构造函数里面利用this关键字给构造函数的本对象赋值；

**注意：利用此方式创建对象的时候，如果不加new关键字的话则该对象的属性变成了全局属性**

**对象名.constructor 获取该对象的构造函数；**

**构造函数加不加new关键字很重要，会直接影响该对象的上下文属性；表示window或者本身自己；**

**arguments.callee:该对象**

**this instanceof arguments.callee:判断利用该构造函数时创建对象时是否有new关键字**

**创建对象的时候先判断有没有加new关键字，没有加上的话我们就自己加；if()return**