复习：

1. 值类型
2. 特殊类型
3. 引用类型
   1. 原生对象
   2. 宿主对象 BOM DOM
   3. 自定义对象

原生对象——17个

Object Function

Array Date Math RegExp

Number String Boolean

Error SyntaxError TypeError ReferenceError URIError RangeError EvalError

Global

练习：

创建一个ATM界面需要的函数：withdraw( money )，负责从当前账户余额中减去待取的金额

1.错误/异常处理机制

错误处理机制：专用于处理项目中“非功能性需求(安全性需求)”，把这种与主业务需求并非密切相关的需求从核心代码中剥离出来。

**错误/异常：**程序中出现的“非正常的运行状态”，一般情况下，解释器/用户可以抛出一个Error对象，以通知当前函数的调用者：此处发生了一个异常情形，需要用户加以处理。

**错误处理：**若用户不处理程序中抛出的Error对象，则程序将终止执行；若用户“试着...捕捉...”错误的发生，并加以处理的话，程序仍然可以绕过错误继续执行下去。

**语句1;**

**try{**

**语句2;**

**语句3;**

**}catch(err){**

**语句4; //错误处理语句**

**}**

**语句5;**

若上述代码没有错误抛出： 语句1=>语句2=>语句3=>语句5

若语句2错误抛出： 语句1=>语句2=>语句4=>语句5

若语句3错误抛出： 语句1=>语句2=>语句3=>语句4=>语句5

try...catch结构中，还可以在尾部添加一个finally子句，**不论错误抛出与否(甚至执行了return)，都要执行finally操作**——表达的含义：执行try块不论正常与否，都要进行的扫尾/清理操作。

try{

语句1

语句2

return;

语句3

}catch(er){

语句4

}finally{

语句5

}

语句6

2.JavaScript-OOP

OOP OOT OOA

Object-Oriented Programming/Programmer

Object-Oriented Testing/Tester

Object-Oriented Architect

面向对象：Java C# AS4 C++ PHP JavaScript

面向过程：C C++ PHP JavaScript

面向切面：基于面向对象 而又高于面向对象——发现并抽取面向对象中切面对象

|  |
| --- |
| 面试题：  面向过程和面向对象的异同？  面向过程：把复杂业务划分为若干个“Procedure、Function”。  function giveOrder(order, money){  }  function findShop(){  }  function buy(){  return bread;  }  function getBack(){  }  function eat(){  }  面向过程的思维方式：彻底的理解现实场景，理清其中的逻辑关系和运行顺序，划分为若干个小的处理单元——落实为function |
| Teacher{  money;  giveOrder()  eat();  }  Student{  speed;  findShop()  buy()  getBack();  }  LadyBread{  bread[]  makeBread()  sellBread()  }  面向对象的思维方式：首先观察现实场景，发现其中有哪些角色和对象，赋予这些对象以属性和行为，让他们彼此发消息，从而构建整个大的应用场景。 |

对象(Object)：现实应用/场景中的某个事物在程序中的体现。对象是无特定顺序的属性的集合。

面向对象的程序需要具备的三个/四个基本特征：

1. 封装(capsulation)：把零散的多个变量组成一个整体
2. 继承(inheritance)：子对象自动获得父对象的所有特征
3. 多态(polymorphism)：一个方法根据参数的不同可以运行出不同的结果
4. 聚集(aggregation)：多个对象可以聚合为一个更大的对象

JavaScript具备上述四种能力。

Object = Property\* + Method\*

属性和方法统一称为对象的特性(attribute)或成员(member)

3.创建对象的四种方式

(1)通过new关键字调用Object()构造方法

(2)对象直接量

(3)?

(4)?

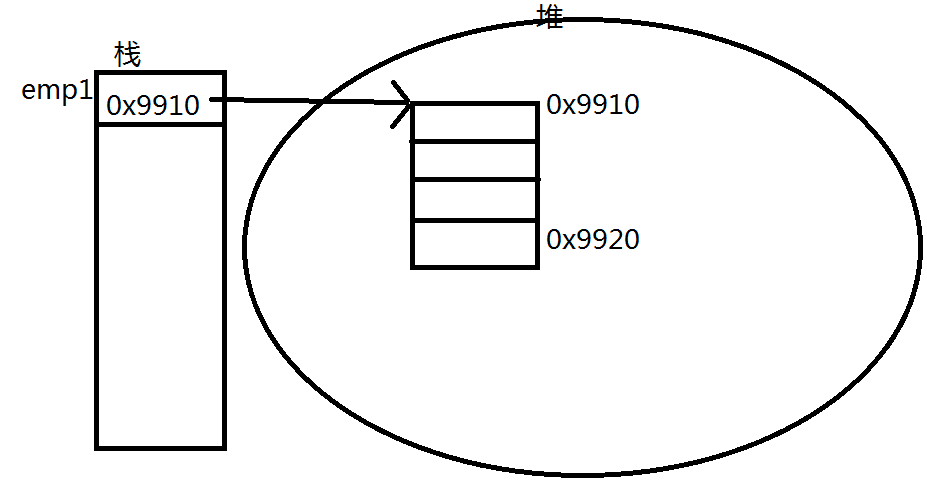
方式一：

如何创建新对象：

**var emp1 = new Object();**

对象的引用 向堆申请新空间 构造方法格式化内存

在栈中分配4/8个字节的引用必需空间；new在堆中申请分配足够的对象空间，对象中的成员全部保存在堆中。



var emp2 = null;

emp2.name = “Tom”; //TypeError：没有空间可用于保存name

console.log( emp2.name ); //TypeError：

如何为对象分配新的属性：

**emp1.ename = ‘Tom’;**

**emp1[‘ename’] = ‘Tommy’;** //注意[]中必需一个string表达式

如何为对象分配新的方法：

**emp1.work = function(){ //赋值为一个匿名函数**

**console.log(this.ename+’在工作’);**

**}**

**emp1.getSalary = gs; //赋值为一个命名函数**

**function gs(){**

**console.log(this.ename+’在领工资’);**

**}**

**emp1.work();**

**emp1.getSalary();**

练习：

创建画图板应用中需要的“矩形对象”，有宽、高、背景颜色三个属性，还有两个方法，getSize()、getPerimeter()

4.创建自定义对象的方式二

**对象直接量(object literal )创建一个新的对象：**

var emp1 = null;

var emp2 = { }; //在堆中创建新的对象，作用于new Object();

emp2.ename = ‘Mary’;

emp2.salary = 3500;

emp2.work = function(){

}

var emp3 = {

属性1: 值1,

‘属性2’: 值2,

方法名1:function(){ },

‘方法名2’: function(){ }

};

emp3.属性1 = 新值;

练习：

在线书城中需要2个Book对象，分别有书名、单价、出版日期、是否特价、购买数量等属性，还有价格小计、获取该书至今为止的出版天数 这两个方法+。

面试题：如何判定一个对象中是否具有某个属性

(1) ‘属性名’ in 对象

(2) 对象.hasOwnProperty(‘属性名’)

(3) 对象.属性 === undefined

(4) if(对象.属性){ ... }

5.JSON

JavaScript Object Notation，js对象表示法，原本是js中创建对象的“对象直接量语法”；现在已经广泛的被所有语言所采用，用于描述和存储数据——JSON是一种轻量级的数据描述和交换方案，很多场合下已经替代了XML，尤其是AJAX应用。

JSON语法：

数据有两种形式：

JSON对象： { }

JSON数组： [ ]

对象中可以存储多个“名/值对”，名和值间用冒号分隔；不同的名值对间用逗号分隔，**名需要使用双引号引起来，值中若是字符串也需要使用双引号引起来**。

JSON字符串和JavaScript对象间的转换：

var data = ‘{“x”:50, “y”:30}’; //JSON格式的字符串

console.log( data.x ); //undefined

把JSON格式的字符串解析为JavaScript对象：

(1) var obj = eval( ‘(’+data+’)’ ); obj.x; //50

(2) var obj = JSON.parse(data); obj.x; //50

把JavaScript对象转换为JSON字符串：

var emp = {ename:’tom’, salary:3500};

var jsonString = JSON.stringify( emp );

console.log(jsonString);

练习：

对象相关的练习：

1.创建一个表示计算器的对象，其中有两个属性可以存储两个数字变量，使用方法分别为这两个属性赋值。再创建加减乘除四个方法，分别返回这两个操作数的对应数学运算的结果。

2."面向对象的分析与设计"

(1)网上商城应用中，销售的主要商品是书籍，用户注册并登录后，可以查看某本书籍的详细信息，并对书籍进行评论

(2)根据上述需求描述，创建所需的对象及属性，模拟出该场景

JSON相关练习：

网上书城“书籍列表”页面中，接收到一段文本字符串，其中描述的是一个JSON形式的书籍数组，包含三本书籍，每本书包含书名、价格、出版日期、作者列表这样几个属性，试把这个字符串解析为JSON数组，并输出每个书籍对象的属性值