学习方法

1. 建立逻辑思维
   1. 多分析
   2. 理解逻辑算法
   3. 多练、多敲、多读
2. 编程习惯
3. 大胆动手
4. 保持好奇好
5. 乐观面对错误

就业能力

1. 分析能力
2. 独立思考能力
3. 程序阅读能力
4. 程序排错能力
5. 独立编写程序能力
6. 团队协作能力
7. 学习能力
8. 善于总结的能力

专业技能

闭包和继承

1. 什么是闭包
   1. 概念：闭包是指在JavaScript中，内部函数总是可以访问其所在的外部函数中声明的参数和变量，即使在其外部函数被返回（寿命终结）了之后。
2. 闭包的特点
   1. 使用闭包有一个优点，也是它的缺点：就是可以把局部变量驻留在内存中，可以避免使用全局变量（全局变量污染导致应用程序不可预测性，每个模块都可调用必将引来灾难，所以推荐使用私有，封装的局部变量）。
3. 闭包的原理
   1. 闭包就是定义在函数中的函数，是函数内外部连接的桥梁闭包的意义是：当前作用域总是能够访问外部作用域中的变量；函数是唯一拥有自身作用域的结构，所以闭包的创建依赖于函数
4. 闭包的应用场景
   1. 保护函数内的变量安全。
   2. 在内存中维持一个变量。
   3. 通过闭包返回局部变量  
      function fn (){  
       var num = 3;  
       return function(){  
       return num;  
       };  
      }  
      alert(fn()());
   4. 使用全局变量进行累加和
   5. 使用局部变量进行累加和
   6. 循环里的匿名函数的取值问题

第二节：应用

1. 掌握闭包的特点和原理
2. 掌握闭包的应用场景

第三节

1. 构造函数继承
2. call/apply继承
   1. function  Person(name,age,love){    
              this.name=name;    
              this.age=age;    
              this.love=love;    
              this.say=function say(){    
                  alert("姓名："+name);    
              }    
          }    
        
          //call方式  (冒充继承)  
          function student(name,age){    
              Person.call(this,name,age);    
          }    
        
          //apply方式    
          function teacher(name,love){    
              Person.apply(this,[name,love]);    
              //Person.apply(this,arguments); //跟上句一样的效果，arguments    
          }
   2. //call与aplly的异同：    
          //1,第一个参数this都一样,指当前对象    
          //2,第二个参数不一样：call的是一个个的参数列表；apply的是一个数组（arguments也可以）
3. prototype的概念
   1. javascript中的每个对象都有prototype属性，Javascript中对象的prototype属性的解释是：返回对象类型原型的引用。

\_ \_ proto\_ \_

1. 原型链继承
2. 混合继承

第四节：应用

1. 掌握继承的原理

第五节：综合应用

1. 分析微信飞机大战游戏的继承关系