提前准备:

1. 今天的历届师兄师姐问过的经典的高频的问题清单都已经整理在小程序->首页->JSCORE->day04里

2. 今天的课程中所需的单词列表都已经整理在小程序->单词->JSCORE->day04里

3. 今天的重点视频、扩展高频鄙视题视频已经整理在小程序->在线->JSCORE->day04里

如果不知道问什么: 谁问的越多！谁培训费花的越值！

1. 笔记中或画图中那句话或者那个位置不明白，立刻问！

2. 案例中哪行代码或注释不明白，立刻问！

3. 课下自己做项目，功能不会做，立刻问！

4. 刷笔试题时遇到不会的笔试题，立刻问！——公众号: 前端大全 定期总结大厂笔试题

5. 看小程序问题清单时，遇到看不懂的问题，立刻问！

衡量是否学会的标准：

1. 可以用自己的话把注释的步骤写出来！才算会！（函数名或程序忘了都没关系！）

2. 可以自己画出内存变化过程图！才算会！（凡是只用眼睛看想得出揭露的！就都是错误的结论！）

不要去网上乱找案例看！只看我的视频！

正课:

1. 什么是面向对象

2. 封装

3. 继承

4. 多态

一. 什么是面向对象:

1. 什么是面向对象: 程序中先用对象保存现实中一个事物的属性和功能，然后再按需访问对象中保存的属性和功能

2. 为什么: 便于大量数据的维护和使用

3. 何时: 今后所有程序都使用面向对象思想开发出来的！

4. 如何: 面向对象三大特点: 封装，继承，多态

二. 封装:

1. 什么是: 其实就是创建一个对象，集中保存现实中一个事物的属性和功能

2. 为什么: 便于大量数据的维护和使用

3. 何时: 今后只要使用面向对象方式编程，都要先封装对象，再按需访问对象中的成员

4. 如何创建对象: 3种方式:

(1). 用{}直接量创建对象:

a. var 对象名={

属性名: 属性值,

... : ... ,

方法名: function(){

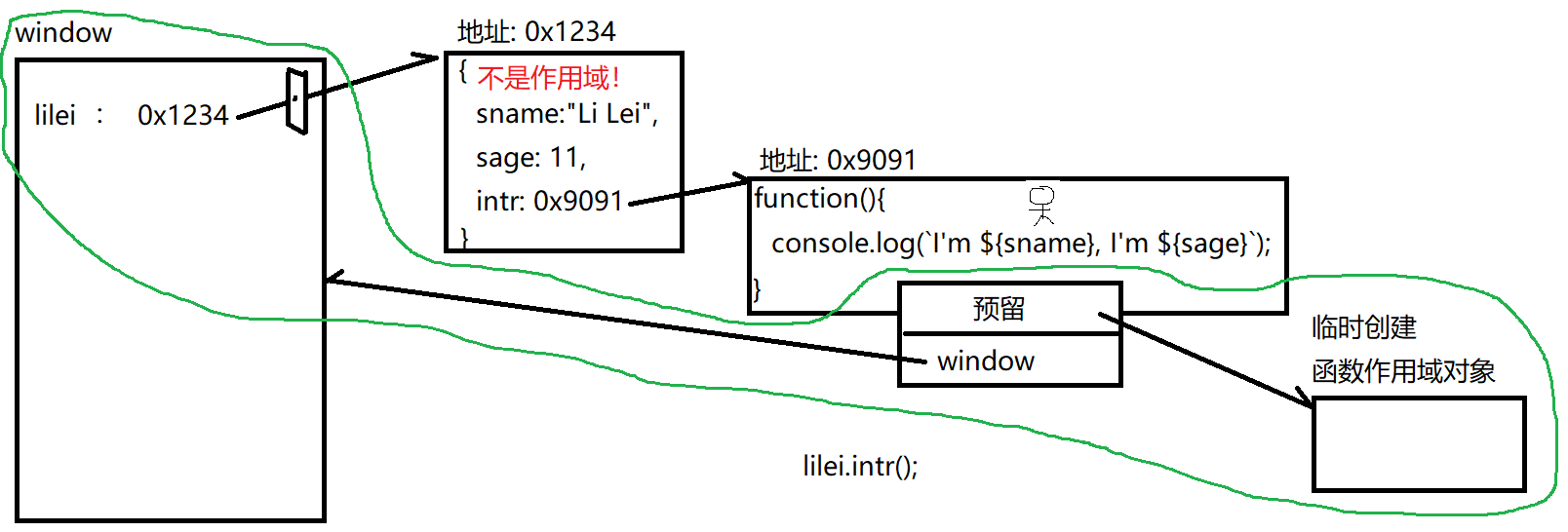
... this.属性名 ...

}

}

b. 问题: 对象自己的函数，想访问对象自己的属性，竟然报错！xxx is not defined

因为: 所有不加.的变量，默认只能在函数作用域和全局作用域window中查找变量。无权进入对象中查找属性。

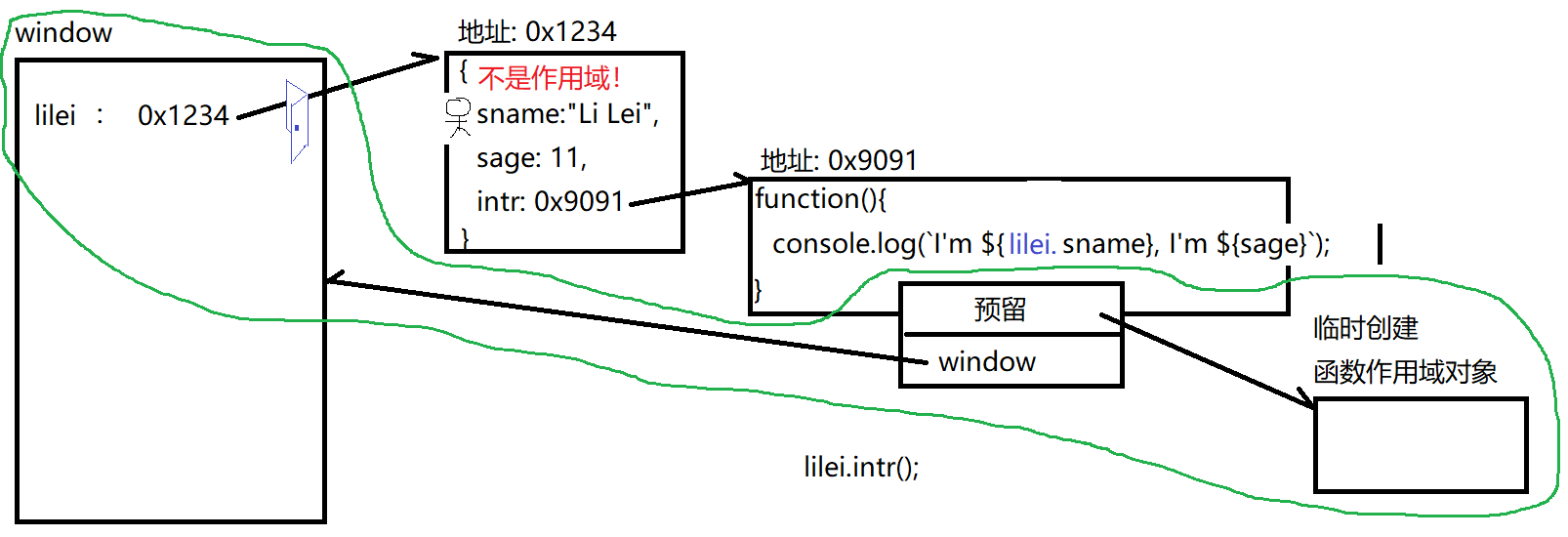


c. 解决:

1). 不好的解决: 在对象方法内写死"对象名.属性名"，来指引js引擎先找到对象，再进入对象中查找属性

问题: 紧耦合。万一对象名修改了，就要被迫跟着修改方法中的写死的对象名。一旦忘记修改方法中写死的对象名，依然会出错！

希望: 松耦合。即使外部对象名修改了，方法内也不需要做任何修改，依然可以访问对象中的属性！



2). 好的解决: 用this关键词代替写死的对象名: "this.属性名"

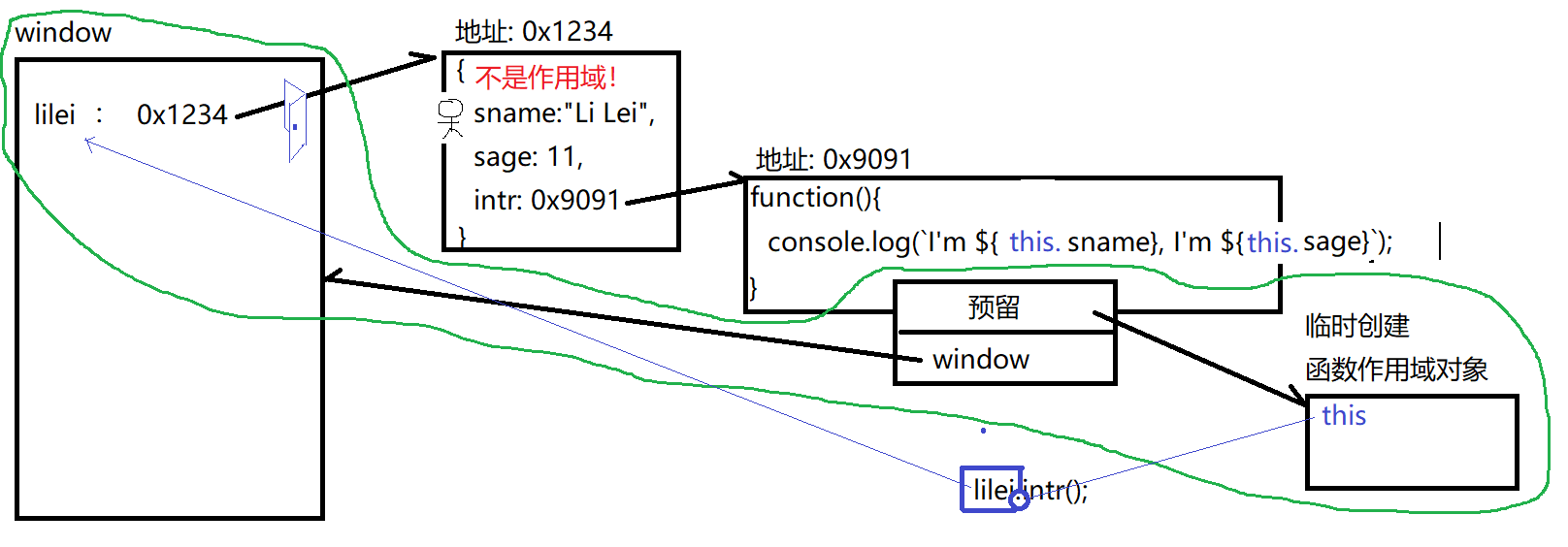
i. 什么是this:

每个函数内自带的，

可以自动获得正在调用当前函数的.前的对象

的关键词

ii. 何时: 只要对象自己的方法内，想使用对象自己的属性，必须加"this.属性名"



|  |
| --- |
| 方法 vs 函数  相同: 都是function！  不同: 独立于任何对象之外，单独存在的function，称为函数  保存在对象内的function，才称为方法！ |

d. 示例: 定义对象保存了李磊的属性和功能

|  |
| --- |
| //想创建一个对象lilei，保存李磊的两个属性: 学生姓名、学生年龄和一个李磊会做的事儿:自我介绍  var lilei={    sname:"Li Lei",    sage: 11,    intr:function(){      console.log(`I'm ${this.sname}, I'm ${this.sage}`)    }  }  //想获取李磊的年龄  console.log(lilei.sage);  //想请李磊做个自我介绍  lilei.intr();  //过了一年，李磊长了一岁  lilei.sage++;  //再获取李磊的年龄  console.log(lilei.sage);  //再请李磊做个自我介绍  lilei.intr(); |

(2). 用new Object()创建:

a. 2步:

1). 想用new Object()创建一个空对象

var lilei=new Object(); //{}其实就是new Object的简写！

2). 为空对象强行添加新属性

lilei.sname="Li Lei";

lilei.sage=11

lilei.intr=function(){

... this.sname ...

}

b. 揭示了: js中一切对象低层都是关联数组！

1). 和关联数组一样，对象内部保存的也是"名值对儿"的集合

2). 都能用下标方式访问对象中的成员属性，且都能用.简写下标访问:

lilei["sage"] 可简写为 lilei.sage

lilei.sage 会被自动翻译为 lilei["sage"]

强调: 如果要访问的属性名不是固定的，而是来自于其他变量或需要动态拼接，则~~不能用.~~，只能用[]，还不能加""

for(var key in lilei){

//key变量接住的值每一轮都在变化: sname sage intr className

console.log(`${key} : ${lilei[key]}`)

~~//console.log(`${key} : ${lilei["key"]}`) //变量不能加""~~

~~//console.log(`${key} : ${lilei.key}`) //.key会被自动翻译为["key"]~~

}

3). 关联数组可通过强行赋值方式随时添加新元素，对象也可通过强行赋值方式随时添加新属性！而不会报错！

4). 访问关联数组中一个不存在的位置，不会报错，而是返回undefined。

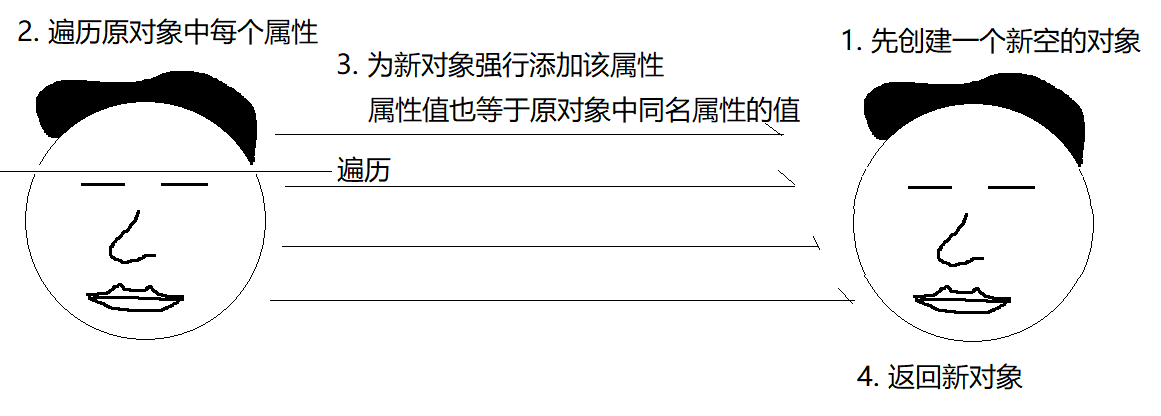
同样，访问对象中一个不存在的属性，也不会报错！也返回undefined

5). 都能用for in遍历每个成员！

c. 示例: 比较关联数组和对象

|  |
| --- |
| var lilei={    sname:"Li Lei",    sage: 11  }  console.log(lilei);  //想获取李磊的年龄  console.log(lilei["sage"]);  //尝试为李磊强行添加一个新属性  lilei.className="初一2班";  //试图访问李磊的爱好  console.log(lilei.fav);//undefined  //遍历lilei中每个成员  for(var key in lilei){    //每一轮都在变化: sname  sage   intr    className    console.log(`${key} : ${lilei[key]}`)  //console.log(`${key} : ${lilei["key"]}`) //变量不能加""  //console.log(`${key} : ${lilei.key}`)  //                             .key会被自动翻译为["key"]  }  var ym=[];  ym["math"]=89;  ym.chs=69;  ym["eng"]=91;  console.log(ym);  //试图访问杨幂的className  console.log(ym["className"])//undefined  //强行为ym添加一个新属性，爱好:跑步  ym.fav="跑步";  //遍历ym中每个成员  for(var key in ym){    //in 依次取出对象中每个属性名,将属性名保存到in前的变量中    //变量key每次循环会依次得到: math  chs  eng   fav    //又因为对象或关联数组中的属性名，有一个技术学名叫"键"    //属性值，简称为值    //所以有人也称对象和关联数组为"键值对儿"的集合    //而键英文就是key    //所以key可以用来表示属性名    console.log(`${key} : ${ym[key]}`)  } |

d. 示例: 克隆一个对象



|  |
| --- |
| var lilei={    sname:"Li Lei",    sage:11  }  //定义一个函数，接受一个旧对象，克隆出一个一模一样的新对象  function clone(oldObj){    //1. 先创建一个新的空对象，准备接收旧对象中的属性和属性值    var newObj={};//=new Object()    //2. 遍历旧对象中每个属性和属性值    for(var key in oldObj){      //3. 每遍历旧对象中一个属性，就为新对象强行赋值添加一个同名的新属性，属性值也等于旧对象中当前同名属性的值      newObj[key]=oldObj[key];    }    //4. 克隆完毕，返回新对象    return newObj;  }  var lilei2=clone(lilei);  console.log(lilei);  console.log(lilei2);  console.log(lilei==lilei2);  //lilei2挣了100块钱  lilei2.money=100;  console.log(lilei.money);  console.log(lilei2.money); |

5. 如何访问对象中的成员: 通过.操作符访问

(1). 访问保存在一个对象中的属性值: 对象名.属性名

(2). 调用保存在一个对象中的方法: 对象名.方法名()

作业:

1. (必须)看小程序视频学习对象相关高频笔试题：

小程序->在线->JSCORE->day04:

高频笔试题 深克隆

高频笔试题 去掉数组中重复的元素

高频笔试题 查询字符串search转对象

高频笔试题 统计字符串中每种字符出现的次数

2. (必须)看小程序视频学习继承相关高频笔试题：

小程序->在线->JSCORE->day04: 判断一个对象是不是数组类型，共有几种方式

3. (必须)看小程序视频学习包装类型:

小程序->在线->JSCORE->day04: 高频笔试题 包装类型 string

4. (扩展)看小程序视频学习面向对象相关高频笔试题:

小程序->在线->JSCORE->day04: 所有以"高频笔试题" 开头的小视频