原型和原型链

题目

知识点

解答

1如何准确判断一个变量是数组类型

答：

var arr=[1,2,3];

console.log(arr instanceof Array);//true 判断一个变量是否为数组类型

typeof arr // object typeof是无法判断是否是数组的

2写一个原型链继承的例子

写一个封装的DOM查询

答:

function Elem(id){  
 this.elem=document.getElementById(id)  
}  
Elem.prototype.html=function (val) {  
 var elem=this.elem  
 if(val){  
 elem.innerHTML=val;  
 return this;//链式操作  
 }else{  
 return elem.innerHTML  
 }  
}  
Elem.prototype.on=function (type,fn) {  
 var elem=this.elem  
 elem.addEventListener(type,fn)  
  
}  
var div=new Elem('div1');

3描述new一个对象的过程

答：

1创建一个新对象

2 this指向这个新对象

3执行代码，即对this赋值

4如果函数没有显示return,就默认返回this

4zetpo（或其它框架）源码中如何使用原型链

答

1阅读源码是高效提高技能的方式

2但不能“坦头苦钻”有技巧在其中

3 zepto设计和源码分析

**知识点**

构造函数

构造函数-扩展

原型规则和示例

原型链

instanceof

**构造函数**

function Foo(name,age){

this.name=name

this.age=age

this.calss=”class-1”

//return this//默认有这一行

}

var f=new Foo(‘eden zhang’,20);

//var f1=new Foo(‘list’,20)//创建多个对象

**构造函数-扩展**

var a={} 其实是var a=new Object()的语法糖

var a=[]其实是var a=new Array()的语法糖

function Foo(){}其实是var Foo=new Function()

使用instanceof判断一个函数是滞是一个变量的构造函数

instanceof Array

var arr=[1,2,3,4,5];

console.log(arr instanceof Array)

**原型规则和示例**

1. **原型规则**

原型规则是学习原型链的基础

1所的有引用类型（数组、对象、函数），都具有对象特性，即可自由扩展属性（除了“null”之外）

2所有引用类型（数组、对象、函数）都有一个\_\_proto\_\_（隐式原型）属性，属性值是一个普通的对象

3所有的函数，都有一个prototype（显式原型）属性，属性值也是一个普通的对象

4所有的引用类型（数组、对象、函数），\_\_proto\_\_属性值指向它的构造函数的”prototype”属性

例

var obj={}//var obj=new Object()的语法糖

obj.a=100

console.log(obj.\_\_proto\_\_===Object.prototype)

因为obj创建方式是new Object()

所以obj.\_\_proto\_\_(隐式原形)===（完全等于）Object.prototype

5当试图得到一个对象的某个属性时，如果这个对象本没这个属性,那么会去它的

\_\_proto\_\_(即它的构造函数的prototype)中寻找

例：

function Foo(name,age){

this.name=name

}

Foo.prototype.alertname=function(){

console.log(" 隐式原型"+this.name)

}

var f=new Foo('edenzhang')

f.printName=function(){

console.log(this.name);

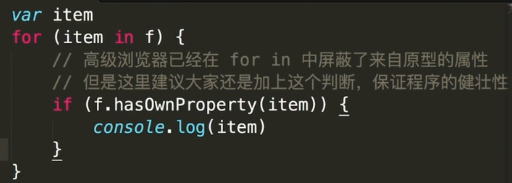
}

f.printName();

f.alertname();

以上中this永远指向对象本身

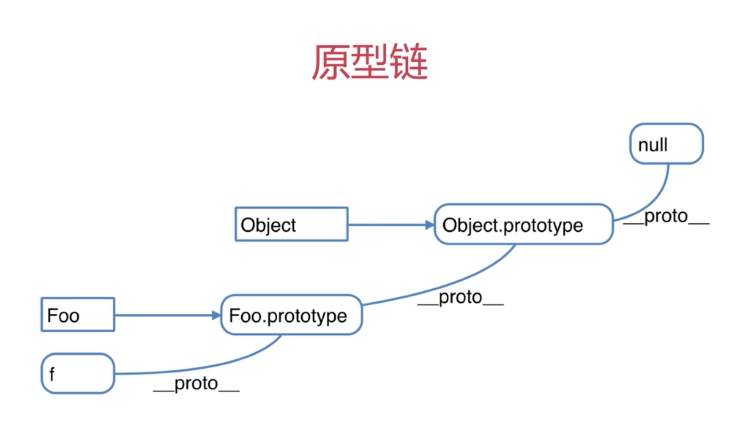
循环对象自身的属性



hasOwnProperty:判断一个对象是否有你给出名称的属性或对象

**原型链**

f.toString();//f.\_\_proto\_\_.proto\_\_中查找



Object.prototype的\_\_proto\_\_ 是null

instanceof:用于判断引用类型属于哪个构造函的方法

f instanceof Foo的判断逻辑是：

f的\_\_proto\_\_一层一层往上，能否对应到Foo.prototype

再试着判断f instanceof Object

解题

1 如何时准确判断一个变量是数组类型

答：

var arr=[1,2,3]

arr instanceof Array// true

typeof arr //object typeof 是无法判断是否是数组的

2 写一个原型链继承的例子

答：

3 描述new一个对象的过程

4 zepto(或其综框架)源码中如何命名用原型链