[1.基础知识 1](#_Toc18945)

[1.1 typeof类型 1](#_Toc19883)

[2. 原型和原型链 2](#_Toc19960)

[2.1 原型链规则 2](#_Toc10597)

[2.1原型链 4](#_Toc18325)

[2.3 instanceof 4](#_Toc19277)

[2.4面试问题解答 4](#_Toc5693)

[2.4.1 如何准确判断一个变量是数组类型 4](#_Toc8274)

[2.4.2 写一个原型链继承的例子 5](#_Toc14808)

[2.4.3 描述new一个对象的过程 5](#_Toc7334)

[2.4.4 zepto(其他框架)如何使用原型链 6](#_Toc8721)

[3. 作用域和闭包 6](#_Toc5247)

[3.1执行上下文 6](#_Toc20628)

[3.2 this含义 6](#_Toc20453)

[3.3 作用域 7](#_Toc11824)

[3.4 闭包（函数套函数） 8](#_Toc22986)

[3.5相关面试问题解答 8](#_Toc12800)

[3.5.1变量提升的概念 9](#_Toc19708)

[3.5.2 说明this几种不同的应用场景 9](#_Toc13081)

[3.5.3 创建10个<a>标签，点击的时候弹出对应的序号 9](#_Toc13761)

[3.5.4 如何理解作用域 9](#_Toc16448)

[3.5.5 实际开发中闭包的应用 10](#_Toc5389)

[4.异步和单线程 10](#_Toc30391)

[4.1 什么是异步 10](#_Toc15031)

[4.2 异步单线程 11](#_Toc22572)

[4.3 问题 11](#_Toc5663)

[4.3.1. 同步和异步的区别是什么？举例子 11](#_Toc24897)

[4.3.2. 一个关于setTimeout的笔试题 11](#_Toc22808)

[4.3.3. 前端使用异步的场景有哪些 11](#_Toc27986)

# 1.基础知识

## 1.1 typeof类型

Number、boolean、string、undefined、function、symbol

1.2

# 原型和原型链

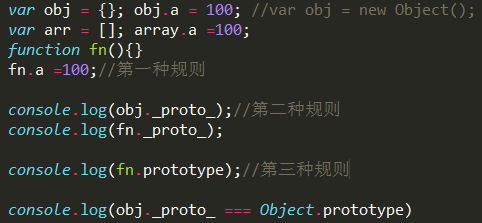
## 2.1 原型链规则

2.1.1 所有引用类型（数组、对象、函数）都具有 对象特性即可拓展属性（null除外）

2.1.2 所有引用类型（数组、对象）都具有一个 \_ptoto\_属性，属性值是一个普通对象

2.1.3 所有函数都具有一个显示原型（ptototype） 属性，属性值是一个普通对象

例：

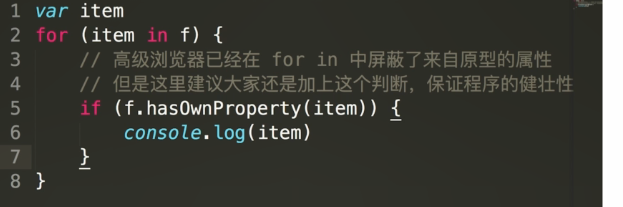


2.1.4 对象的隐式原型（\_ptoto\_）指向构造函数的 显示原型(prototype)

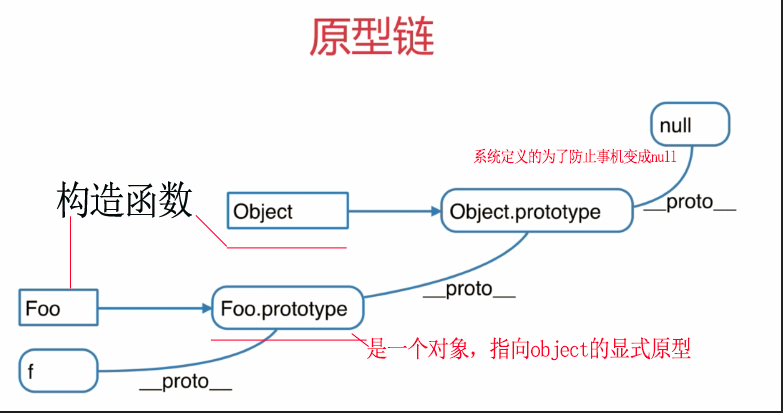
2.1.5 当试图得到一个引用类型的某个属性时，但 这个变量的本身没有这个属性，那么就会去它的 \_proto\_（即构造函数的prototype）里去寻找。



2.1.6 循环对象自身的属性



## 2.1原型链



## 2.3 instanceof

(用 instanceof 判断一个函数是否是一个变 量的构造函数)

例：f instanceof Foo(f 的 \_ptoto\_一层层往上找，能否 对应Foo的prototype)

## 2.4面试问题解答

### 2.4.1 如何准确判断一个变量是数组类型

用instanceof Array 来判断.

### **2.4.2 写一个原型链继承的例子**

function Arez(name){

this.name = name;

}

Arez.prototype.showName = function(){

alert('showName')

}

var Bay = new Arez('wang');

Bay.showName();

### **2.4.3 描述new一个对象的过程**

function Foo (name,age){

this.name = name;

this.age = age;

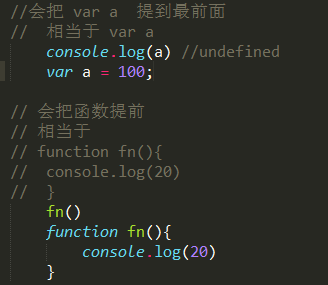
}

var Rec= new Foo (‘cheng’,12);

### **2.4.4 zepto(其他框架)如何使用原型链**

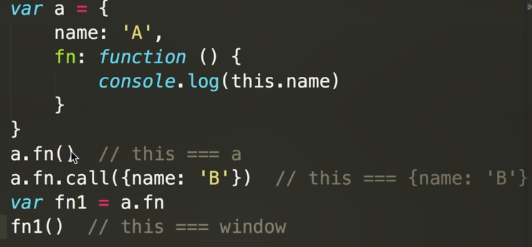
# 作用域和闭包

## 3.1执行上下文



## 3.2 this含义

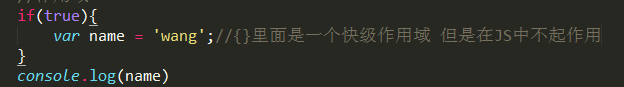
this要在执行时才能确认值，定义时无法确认





## 3.3 作用域

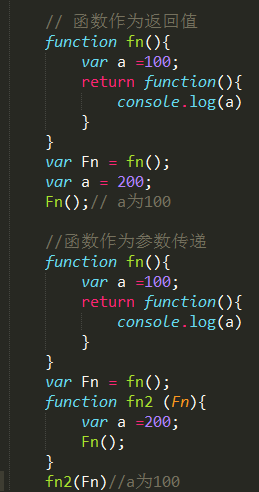
js无块级作用域



## 3.4 闭包（函数套函数）

闭包的使用场景

1. 函数作为返回值
2. 函数作为参数传递



3.5相关面试问题解答

### 3.5.1变量提升的概念

把var a提前、把函数提前

### **3.5.2 说明this几种不同的应用场景**

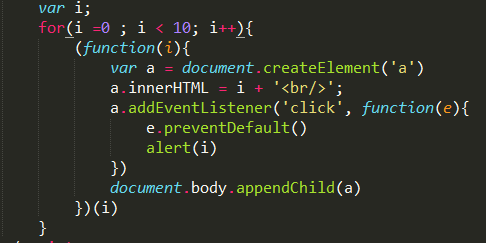
作为构造函数执行

作为对象属性执行

作为普通函数执行

Call apply bind

### **3.5.3 创建10个<a>标签，点击的时候弹出对应的序号**



### 3.5.4 如何理解作用域

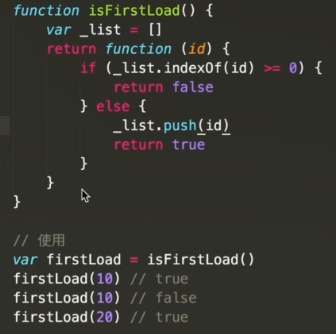
自由变量，自由变量的查找（没有定义会去父级作用域去 寻找定义的参数）

作用域链

闭包的两个场景

### 3.5.5 实际开发中闭包的应用

实际运用在封装变量，收敛权限



# 4.异步和单线程

## 4.1 什么是异步

## 4.2 异步单线程

## 4.3 问题

### 4.3.1. 同步和异步的区别是什么？举例子

### **4.3.2. 一个关于setTimeout的笔试题**

### **4.3.3. 前端使用异步的场景有哪些**

定时任务：setTimeout 、setInverval

网络请求：ajax请求，动态<img>加载

事件绑定