[1.基础知识 2](#_Toc3682)

[1.1 typeof类型 2](#_Toc4332)

[2. 原型和原型链 2](#_Toc2604)

[2.1 原型链规则 2](#_Toc15748)

[2.1原型链 5](#_Toc8478)

[2.3 instanceof 5](#_Toc11418)

[2.4面试问题解答 5](#_Toc6585)

[2.4.1 如何准确判断一个变量是数组类型 5](#_Toc19542)

[2.4.2 写一个原型链继承的例子 6](#_Toc3203)

[2.4.3 描述new一个对象的过程 6](#_Toc12828)

[2.4.4 zepto(其他框架)如何使用原型链 7](#_Toc1124)

[3. 作用域和闭包 7](#_Toc9216)

[3.1执行上下文 7](#_Toc8897)

[3.2 this含义 7](#_Toc27375)

[3.3 作用域 8](#_Toc28037)

[3.4 闭包（函数套函数） 9](#_Toc6926)

[3.5相关面试问题解答 9](#_Toc24198)

[3.5.1变量提升的概念 10](#_Toc11289)

[3.5.2 说明this几种不同的应用场景 10](#_Toc18139)

[3.5.3 创建10个<a>标签，点击的时候弹出对应的序号 10](#_Toc18900)

[3.5.4 如何理解作用域 10](#_Toc28194)

[3.5.5 实际开发中闭包的应用 11](#_Toc17596)

[4.异步和单线程 11](#_Toc21709)

[4.1 什么是异步 11](#_Toc3494)

[4.2 异步单线程 11](#_Toc31209)

[4.3 问题 12](#_Toc15083)

[4.3.1. 同步和异步的区别是什么？举例子 12](#_Toc18422)

[4.3.2. 一个关于setTimeout的笔试题 12](#_Toc19532)

[4.3.3. 前端使用异步的场景有哪些 12](#_Toc10712)

[5. 日期和Math 13](#_Toc19132)

[5.1 知识点 13](#_Toc4065)

[5.2问题 14](#_Toc2992)

[5.2.1 获取2-17-05-12格式的日期 14](#_Toc11900)

[5.2.2 获取随机数，要求是长度一直是字符串格式 14](#_Toc23455)

[5.2.3 写一个能遍历对象和数组的通用forEach 函数 14](#_Toc19217)

[6 数组API 15](#_Toc12762)

[7 JS-Web-API（w3c标准） 15](#_Toc20752)

[8 DOM（document object modle） 16](#_Toc24558)

[8.1 DOM的本质 16](#_Toc30789)

[8.2 DOM的节点操作 16](#_Toc286)

[8.3 DOM的结构操作 16](#_Toc20198)

[8.4题目 17](#_Toc21189)

[8.4.1 DOM是哪种基本的数据结构 17](#_Toc19706)

[8.4.2 DOM操作常用的API 17](#_Toc9364)

[8.4.3 DOM的attr 和 property有和区别 17](#_Toc1032)

[9 事件 17](#_Toc18451)

[9.1 通用事件的绑定 17](#_Toc21808)

[9.2 事件冒泡 18](#_Toc24777)

[9.3 代理 18](#_Toc10526)

[题目 18](#_Toc17137)

[编写一个通用的事件监听函数 18](#_Toc16497)

[描述事件冒泡的流程 18](#_Toc15540)

[对于一个无限下载加载图片的页面，如何给每个图片绑定事件,（代理）设置在div上面，用target拿到点击的那一个的标签。 19](#_Toc10386)

[10 Ajax 19](#_Toc17971)

[10.1 知识点 19](#_Toc16)

[10.1.1 XMLHttpRequest 19](#_Toc14060)

[10.1.2 跨域 20](#_Toc20211)

[10.3 问题 21](#_Toc21309)

[10.3.1 手动编写一个Ajax，不依赖第三方库 21](#_Toc6831)

[10.3.2 21](#_Toc23320)

[11 存储 21](#_Toc23191)

# 1.基础知识

## 1.1 typeof类型

Number、boolean、string、undefined、function、symbol

1.2

# 原型和原型链

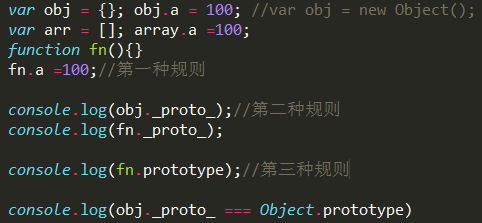
## 2.1 原型链规则

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：

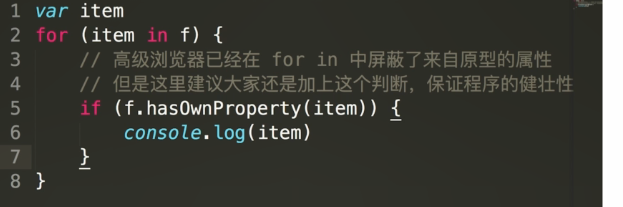


2.1.4 对象的隐式原型（\_ptoto\_）指向构造函数的 显示原型(prototype)

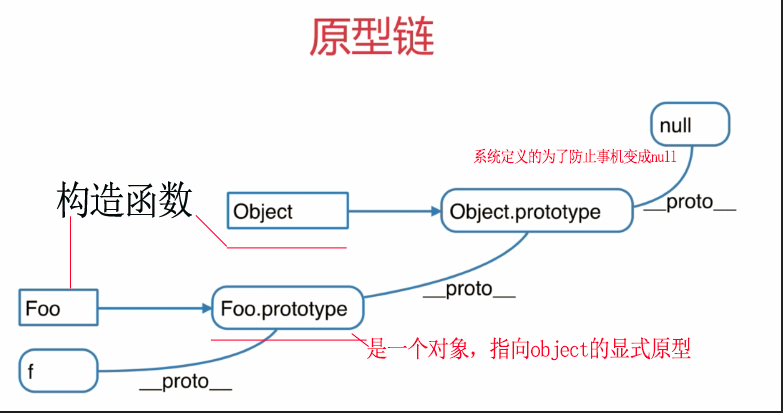
2.1.5 当试图得到一个引用类型的某个属性时，但 这个变量的本身没有这个属性，那么就会去它的 \_proto\_（即构造函数的prototype）里去寻找。



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;

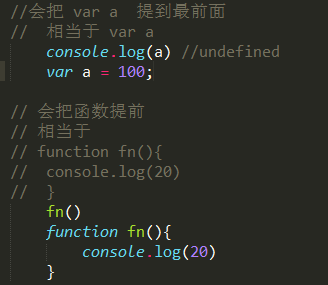
}

var Rec= new Foo (‘cheng’,12);

### **2.4.4 zepto(其他框架)如何使用原型链**

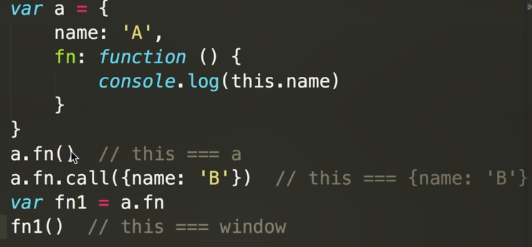
# 作用域和闭包

## 3.1执行上下文



## 3.2 this含义

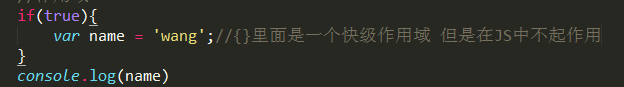
this要在执行时才能确认值，定义时无法确认





## 3.3 作用域

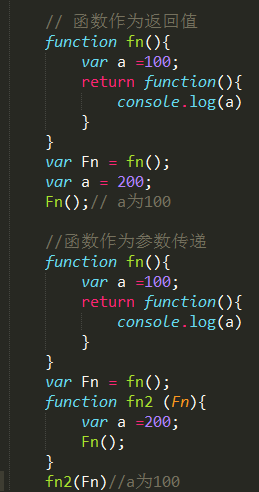
js无块级作用域



## 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**

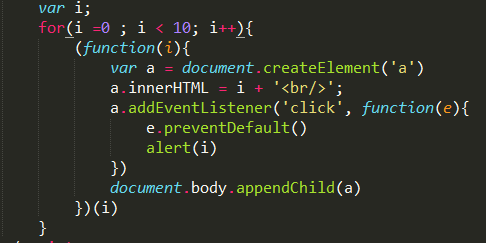
作为构造函数执行

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

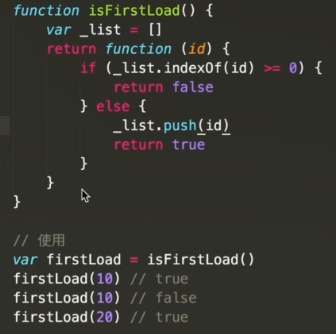
自由变量，自由变量的查找（没有定义会去父级作用域去 寻找定义的参数）

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

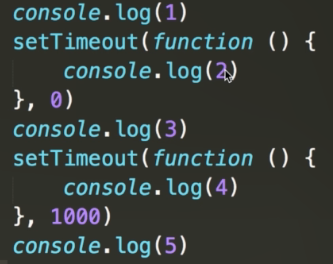
js是一种单线程语言，一次只能做一件事，函数一个一个执行，不能一起执行，不能让他阻塞所以用异步

## 4.3 问题

### 4.3.1. 同步和异步的区别是什么？举例子

阻塞代码（alert是同步，setTimeout是异步）

### **4.3.2. 一个关于setTimeout的笔试题**



会打印出来1 3 5 2 4

### **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

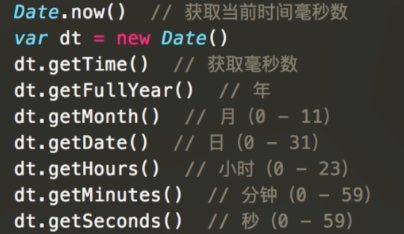
事件绑定

特点：等待需要异步（js是一个单线程语言 只能同时做 一件事情）

# 5. 日期和Math

## 5.1 知识点

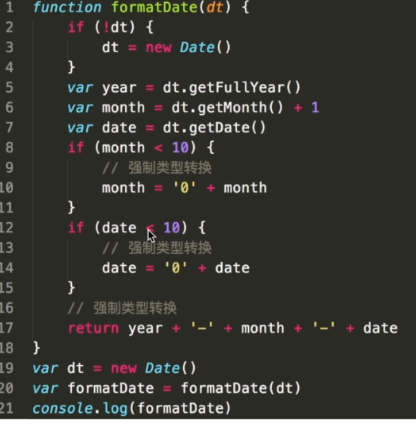
日期



Math(获取随机数Math.random(), random是清除缓存用)

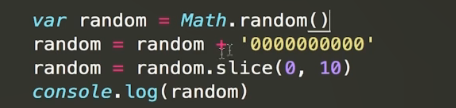
## 5.2问题

### 5.2.1 获取2-17-05-12格式的日期



### 5.2.2 获取随机数，要求是长度一直是字符串格式

先加10个0 是为了截取不报错



### 5.2.3 写一个能遍历对象和数组的通用forEach 函数

# 6 数组API

forEach 遍历所有元素

every 判断所有元素是否都符合条件

some 判断是否有一个元素符合条件

sort 排序(a-b 从小到大排序)

map 对元素重新组装，生成新数组

filter 过滤符合条件的数组

# 7 JS-Web-API（w3c标准）

Dom操作

Bom操作

事件绑定

Ajax请求

存储

# 8 DOM（document object modle）

## 8.1 DOM的本质

## 8.2 DOM的节点操作

## 8.3 DOM的结构操作

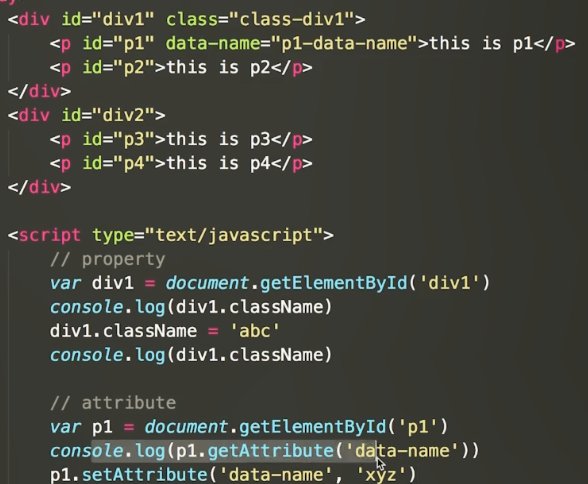
新增节点、获取父元素、获取子元素、删除节点

## 8.4题目

### 8.4.1 DOM是哪种基本的数据结构

### 8.4.2 DOM操作常用的API

### 8.4.3 DOM的attr 和 property有和区别



# 9 事件

## 9.1 通用事件的绑定

## 9.2 事件冒泡

## 9.3 代理

## 题目

### 编写一个通用的事件监听函数



### 描述事件冒泡的流程

点击a1的点击事情会再去父元素上面找，冒泡到最外层。

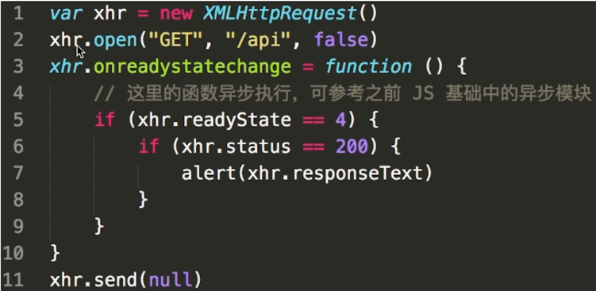
### 对于一个无限下载加载图片的页面，如何给每个图片绑定事件,（代理）设置在div上面，用target拿到点击的那一个的标签。



# 10 Ajax

# 10.1 知识点

# 10.1.1 XMLHttpRequest





# 10.1.2 跨域

浏览器有同源策略，不允许Ajax访问其他域接口。（协议、域名、端口有一个不一致就会跨域。Img link script 这3个标签允许跨域）

Jsonp(请求一个api是一个代码片段,在JS写一个callback函数把api返回的传过来)



服务器端：http header



# 11 存储

Cookie(缺点):

是一个字符串，存储比较麻烦，需要封装才能存储， documnet.cookie = ..

每次请求都会带AJAX请求中

储存量太小了只有4KB。

LocationStorage和sessionStorage

最大存储5M

API简单易用(setItem、getItem)

区别：sessionStorage浏览器关了就会清理

坑：ios safari的隐藏模式下 LocationStorage的getItem会报错（建议用try-catch包起来）

# 12 开发工具