**30、如何判断一个变量是否是数组**

（1）instanceof

var ary = [1,23,4];

console.log(ary instanceof Array)//true;

（2)原型链方法

var ary = [1,23,4];

console.log(ary.\_\_proto\_\_.constructor==Array);//true

console.log(ary.constructor==Array)//true 这两段代码是一样的

31、 你的职业规划是什么

**32、 document load与document ready的区别**

1.load是当页面所有资源全部加载完成后（包括DOM文档树，css文件，js文件，图片资源等），执行一个函数

问题：如果图片资源较多，加载时间较长，onload后等待执行的函数需要等待较长时间，所以一些效果可能受到影响

（所有的都加载完）

2.$(document).ready()是当DOM文档树加载完成后执行一个函数 （不包含图片，css等）所以会比load较快执行

在原生的jS中不包括ready()这个方法，只有load方法就是onload事件

（只加载树，没有CSS等）

**33、 实现一个字符串去重的方法**

function selectArr(arr){

var json={};

var arr2=[];

for(var i=0;i<arr.length;i++){

if (typeof(arr[i])=="number") {

if(json[arr[i]]==null){

json[arr[i]]=arr[i];

arr2.push(arr[i]);

}

}

}

return arr2

}

1.创建一个新的数组存放结果

2.创建一个空对象

3.for循环时，每次取出一个元素与对象进行对比，如果这个元素不重复，则把它存放到结果数组中，同时把这个元素的内容作为对象的一个属性，并赋值为1，存入到第2步建立的对象中。

1.先将原数组进行排序

2.检查原数组中的第i个元素 与 结果数组中的最后一个元素是否相同，因为已经排序，所以重复元素会在相邻位置

3.如果不相同，则将该元素存入结果数组中

**34、 编写方法实现Number类型数组大到小排序，不可以使用内置sort方法**

35、 排序算法有哪些，你最常用哪个，应用在哪里

/\* function sorting(s,l,r){

if (l < r) {

//Swap(s[l], s[(l + r) / 2]); //将中间的这个数和第一个数交换 参见注1

var i = l, j = r, x = s[l];

while (i < j){

while(i<j &&s[j]>=x){

j--;

} // 从右向左找第一个小于x的数

if(i<j){

s[i]=s[j];

i++

}

while(i<j &&s[i]<x){

i++;

} // 从左向右找第一个大于等于x的数

if(i<j){

s[j]=s[i];

j--;

}

}

s[i] = x;

sorting(s, l, i - 1); // 递归调用

sorting(s, i + 1, r);

}

}

arr=[3,4,1,7,6,2,0];

sorting(arr,0,arr.length-1);

alert(arr)\*/

36、 谈谈你对es6箭头函数的理解

37、 谈谈你对模块化的理解

38、 编写一个函数，把传入的“post-all-msg”格式的字符串转成“postAllMsg”格式驼峰表示法。

39、  **css如何使用em、rem单位来控制不同屏幕的字体比例尽量一致？**

Rem根据根目录的字体计算，还可以计算宽度等。Em根据父元素字体计算

一、 rem的特点：

1、rem的大小是根据html根目录下的字体大小进行计算的。

2、当我们改变根目录下的字体大小的时候，下面字体都改变。

3、rem不仅可以设置字体的大小，也可以设置元素宽、高等属性。

二、em的特点：

1、字体大小是根据父元素字体大小设置的。

40、  **如下接口，返回内容为html，请通过ajax从接口获取内容填充到div内，请求设为同步。接口url:http://test.cn/getHtml参数：name,age**

**41、 写一个数组去重的方法，使用hash表的方式**

**42、 如何监控文本框的回车事件并进行相应？请写出核心代码**

**43、 一个ul中有一千个li，每一个li都需要触发js操作。如何降低重复的事件绑定,从而降低dom操作的消耗性能？**

事件代理。

给ul绑定事件：

EventUtil.addHandler(list,”click”,function(event){

Event=EventUtil.getEvent(event);

Var targrt=EventUtil.getTargrt(event);

Switch(target.id){

Case “1”:breake;

Case “2”:breake;

}

}

**44、 怎样解决移动端点击穿透问题**

**45、 怎么解决移动端滑动穿透问题**

**46、 html网页自适应移动端设备需要添加什么标签？如何强制让文档与设备宽度保持1：1**

**47、 如何解决手机端web页面跳转白屏？**

**48、 只把图片的左上边框和右下边框设为圆角，css代码怎么写？**

#div1{

width:50px;

height:50px;

background-color:yellow;

border-top-left-radius:10px;

border-bottom-right-radius:10px;

}

**49、 ajax成功返回的readyState有几种状态？各自表示什么意思？在获取数据时候处于什么状态？**

Onreadystatechangge: readyState改变触发。（0，1，2，3，4）

Status：提示错误信息

**50、 ajax的success方法和complete方法有什么不同？**

|  |
| --- |
| $.ajax({     type: "post",     url: url,     dataType:'html',     success: function(data) { },     complete: function(XMLHttpRequest, textStatus) { },     error: function(){}  }); |

success ： 当请求成功时调用的函数。这个函数会得到一个参数：从服务器返回的数据。当请求成功时调用函数，即status==200。

complete ：当请求完成时调用的函数。这个函数会得到两个参数：XMLHttpRequest对象和一个描述请求成功的类型的字符串。当请求完成时调用函数，即status==404、403、302...。

所以，在写success或者complete的方法时，注意传入的参数，和使用传进来参数对象来解决我们的问题

**51、 自适应和响应式的区别**

自适应是随着浏览器的大小变化而变化，响应式是响应屏幕尺寸，需要通过CSS来实现

**52、 vue中子组件如何传值到父组件**

**53、 对vuex的理解**

**54、 对JavaScript面向对象的了解，是否有应用过？**

**55、 从后台收到两张图片，如何将其中的一张放到另一张的指定位置并合并成一张？**

**56、 编写一个方法，判断传进来的变量是否是整形**

parseInt(),Math.round();

Number()用于任何类型，parseInt()用于字符串类型

**57、 对Nodejs的理解**

**58、 instanceof和typeof的区别**

instanceof判断一个**对象**是否是另一个**对象**的实例，而数字1是基本数据类型，不是对象。

typeof（）返回六个值：null undefined number object function boolean

**59、 原型链constructor指向的确定。**

function SuperType(argument) {

this.property=true;

}

SuperType.prototype.getSuperValue = function() {

return this.property

};

function SubType(argument) {

this.subproperty=false;

}

SubType.prototype=new SuperType();

SuperType.prototype.getSubValue = function() {

return this.subproperty

};

var instance=new SubType();

instance.constructor = SubType.prototype.constructor;

而SubType.prototype= new SuperType()

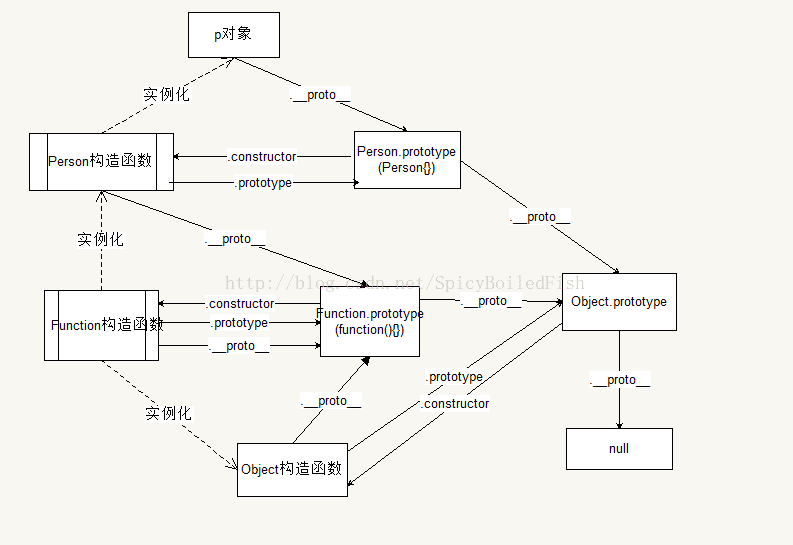
因此SubType.prototype.constructor= SuperType.prototype.constructor;

最终即SuperType构造函数

# 60、详解prototype与\_\_proto\_\_区别

Prototype:函数才具有的属性。指向一个对象。hasOwnProperty()判断是否为自有属性。

\_\_proto\_\_:隐式原型，对象才有。一个对象的隐式原型指向构造该对象的构造函数的原型



图中有几个难点：

1.Function构造函数（可以看作一个对象）可以用Function.\_\_proto\_\_来访问Function.prototype. 这是因为Function构造函数的构造函数是他本身，作为实例化对象的角色来访问，可行。**（重要：function的特殊性。Person是function构造出来的实例）**

2.任何函数都是函数，他都继承Function的所有属性和方法，而Function是内置的构造函数，；也是对象，都是继承Object的所有属性和方法。

hasOwnProperty()判断是否为自有属性。

SuperType.ptototype.isProtoTypeOf(instance):判断原型关系。

Object.getPrototypeOf()方法读取[[Prototype]]属性的值

**61、 定时器参数为0，并不立即执行**

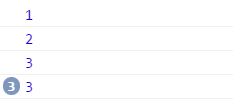
var fuc = [1,2,3];

for(var i in fuc){  
 setTimeout(function(){console.log(fuc[i])},0);

console.log(fuc[i]);

}

打印结果：



这里涉及到javascript单线程执行的问题：javascript在浏览器中是单线程执行的，必须在完成当前任务后才执行队列中的下一个任务。

另外，对于javascript还维护着一个setTimeout队列，未执行的setTimeout任务就按出现的顺序放到setTimeout队列，等待普通的任务队列中的任务执行完才开始按顺序执行积累在setTimeout中的任务。

所以在这个问题里，会先打印1 2 3，而将setTimeout任务放到setTimeout任务队列，等循环中的打印任务执行完了，才开始执行setTimeout队列中的函数，所以在最后会接着打印3次3。

<body>

<p>

<input type="text" id="input" value=""/>

<span id="preview"></span>

</p>

</body>

<script type="text/javascript">

(function(){

function $(id){

return document.getElementById(id);

}

$('input').onkeypress = function(){

$('preview').innerHTML = this.value;

}

})();

</script>

</html>

这个keypress函数原意是监听到用户输入字符串就将其完整的显示出来，但是奇怪的是最后一个字符串总是没能显示出来:

C:\Users\dw\AppData\Local\Temp\1522759085(1).png

但是只要改下onkeypress函数就好：

$('input').onkeypress = function(){

setTimeout(function(){$('preview').innerHTML = $('input').value;},0);

}

解释：keydown，keypress，keyup的顺序执行事件。当keypress按下后，要执行完对应的事件，才更新DOM的状态。因此，value值是取不到最后一个的。（单线程）

get('makeinput').onmousedown=function(){

var input = document.createElement('input');

input.setAttribute('type', 'text');

input.setAttribute('value', 'test1');

get('inpwrapper').appendChild(input);

input.focus();

}

get('makeinput2').onmousedown = function(){

var input = document.createElement('input');

input.setAttribute('type', 'text');

input.setAttribute('value', 'test1');

get('inpwrapper2').appendChild(input);

setTimeout(function(){input.focus();},0);

}

第一个的input无法获得焦点。因为获得焦点后，按钮又把焦点夺走。

***事件顺序：先触发onmousedown 再触发onfocus再触发onmouseup再触发onclick。***

# 62、对js Promise理解