这几天面试了几家中小型公司，表现差强人意，面试官们问的问题都差不多，不过我回答的都不怎么样，只要面试官去扣些细节，我就懵了，现在总结一下我没回答好的问题：

.好几个面试官都问到了url的问题，比如一个链接：http://www.baidu.com/search?name=mili&age=13&work=IT#homepage，如何获取到search后面搜索的内容，如何将其中一个参数去掉，比如去掉age=13， 如何将search转换成一个json格式的字面量等等。。。。。。

apply()与call()的作用分别是什么有什么区别？

关于闭包函数中有两个函数，如何返回这两个函数？

bootstrap中的栅格是怎么实现的？

数组处理的几个方法？pop()/push()/shift()/unshift()作用分别是什么，返回值是什么？

跨域的几种方式？

在移动端使用flex布局一个购物车

对ES6的了解，以及ES6与ES5之前的区别以及新增的一些东东？

在移动端设置 border 为1px ，有时候会显示成 3px ，该怎么解决？

call、callee 的区别是什么？

http 的状态码有哪些，分别代表的意思？

angular 中 ng-if 和 ng-show 的区别？

angular 的特点是什么，双向绑定的实现原理是什么？（感觉这个问题好变态）

对viewport了解多少，它的值有哪些，设不设置 viewport 有什么区别？

@milixie

Owner

milixie commented on 8 Nov • edited

先了解下JSON.stringify()和JSON.parse()，即json的反序列化和序列化，因为一直把这两个弄混

var obj = {name: 'mili', age: 13, work: 'IT'};

var str = JSON.stringify(obj);

console.log(str); //'{"name":"mili","age":13,"work":"IT"}'字符串形式

console.log(JSON.parse(str)); // {name: "mili", age: 13, work: "IT"} 字面量形式

之前一直以为这两个方法可以直接去处理url这类型的string，囧

获取到search后面搜索的内容

如果你使用的nodejs，那就简单多了，直接在node的命令行中，使用node的url.parse('')就可以了

$ node

> url

{ parse: [Function: urlParse],

resolve: [Function: urlResolve],

resolveObject: [Function: urlResolveObject],

format: [Function: urlFormat],

Url: [Function: Url] }

> url.parse('http://www.baidu.com/s?name=mili&age=12&work=it')

{ protocol: 'http:',

slashes: true,

auth: null,

host: 'www.baidu.com',

port: null,

hostname: 'www.baidu.com',

hash: null,

search: '?name=mili&age=12&work=it',

query: 'name=mili&age=12&work=it',

pathname: '/s',

path: '/s?name=mili&age=12&work=it',

href: 'http://www.baidu.com/s?name=mili&age=12&work=it' }

>

这样就直接可以得到你想要的

但是如果你要使用js去获取到每个key的value，则需要写一个函数：

function getParamsByKey(key, url) {

if (!url) {

url = window.location.href;

}

key= key.replace(/\[\]/g, "\\$&"); //有些href可能使酱紫的：http://www.baidu.com/s?[]=222#p; 我猜的，这行的作用就是如果是这种形式的话也可以正常使用该函数

var reg = new RegExp("[?&]" + key + "(=([^&#]\*)|&|#|$))",

res = reg.exec(url);

if (!res) {

return null;

}

if (!res[2]) {

return ' ';

}

return decodeURIComponent(res[2].replace(/\+/g, ' ')); //进行解码

}

var url = 'http://www.baidu.com/s?name=mili&age=12&work=it';

getParamsByKey('age', url); //12

获取search：写一个函数即可

function getSearch(url) {

return "?" + url.split("?")[1].split("#")[0];

}

getSearch(url);

获取到search并将其转换成json格式的字面量：

function searchTurnJson(s) {

if (!s) {

return null;

}

s = s.split('?')[1].replace(/\+/g, "").replace(/\=/g, ":");

sObj = s.split('&');

var objJson = {};

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

var objKey = sObj[i].split(':')[0],

objValue = sObj[i].split(':')[1];

objJson[objKey] = objValue;

}

return objJson;

}

var query = searchTurnJson(getSearch(url));

console.log(query);

去掉其中一个搜索条件,输出最终的url(老感觉这个方法不靠谱~)

function removeOneSearch(key, url) {

var keyV = key + "=" + getParameByName(key, url);

url = url.replace(new RegExp(keyV), "").replace("?&", "?").replace("&#", "#");

return url;

}

var removeRes = removeOneSearch('work', url);

@milixie

Owner

milixie commented on 8 Nov • edited

apply()与call()的作用分别是什么有什么区别？

作用基本相同，代替另一个对象调用一个方法，call 方法可将一个函数的对象上下文从初始的上下文改变为由 thisObj 指定的新对象

区别：参数不同

call(): Person.apply(obj, 'mili', 12);

apply：Person.call(obj, ['mili', 12]);

例子：

var anim = function(type, age) {

this.type = type;

this.age = age;

this.show = function(){

console.log('show animal information:' + this.type);

}

}

var b = {};

anim.call(b, 'cats', 12);

anim.apply(b, ['dogs', 2]);

var foo = function(arg1, arg2) {

console.log(arg1,arg2);

}

foo.call(this, 'ddd', '333');

foo.apply(this, ['333', 'ede']);

function Doctor(name, age){

this.name = name;

this.age = age;

this.info = function(){

console.log('Doctor is :' + this.name);

}

}

function DoctorType(type) {

Doctor.call(this, 'hua', 43);

this.type = type;

}

var d = new DoctorType('Doctor');

console.log(d);

@milixie

Owner

milixie commented on 9 Nov

关于闭包函数中有两个函数，如何返回这两个函数？

function foo(){

var str = "ncy";

var obj = {

a: function(){

console.log(str + '---a');

},

b: function(){

console.log(str + '---b');

}

};

return obj;

}

var f = foo();

f.a();

f.b()

@milixie

Owner

milixie commented on 9 Nov • edited

bootstrap中的栅格是怎么实现的？

关于这个问题，我说是每个col-sm-x的宽度是百分比形式，使用float:left实现，但面试官很肯定的说不是，我也就懵了，但是回来查看了bootstrap的源码，bootstrap根据浏览器兼容性使用了两种形式去实现的，源码如下：

//第一种

.col-xs-1 {

float: left;

width: 8.333333%;

}

.col-xs-2 {

float: left;

width: 16.666667%;

}

.col-xs-3 {

float: left;

width: 25%;

}

//第二种：

.col-xs-1 {

-webkit-box-flex: 0;

-webkit-flex: 0 0 8.333333%;

-ms-flex: 0 0 8.333333%;

flex: 0 0 8.333333%;

max-width: 8.333333%;

}

.col-xs-2 {

-webkit-box-flex: 0;

-webkit-flex: 0 0 16.666667%;

-ms-flex: 0 0 16.666667%;

flex: 0 0 16.666667%;

max-width: 16.666667%;

}

.col-xs-3 {

-webkit-box-flex: 0;

-webkit-flex: 0 0 25%;

-ms-flex: 0 0 25%;

flex: 0 0 25%;

max-width: 25%;

}

//scss源码：

@mixin make-col($size, $columns: $grid-columns) {

@if $enable-flex {

flex: 0 0 percentage($size / $columns);

// Add a `max-width` to ensure content within each column does not blow out

// the width of the column. Applies to IE10+ and Firefox. Chrome and Safari

// do not appear to require this.

max-width: percentage($size / $columns);

} @else {

float: left;

width: percentage($size / $columns);

}

}

//如果支持flex，则使用flex布局的方式，如果不支持，则使用float加百分比宽度的布局

@milixie

Owner

milixie commented on 9 Nov • edited

数组处理的几个方法？pop()/push()/shift()/unshift()作用分别是什么，返回值是什么？

var arr = ['mili', 23, 'girl', 'IT'];

console.log(arr.unshift('xie')); //5

console.log(arr); // ['xie', 'mili', 23, 'girl', 'IT']

console.log(arr.shift()); //xie

console.log(arr); // ['mili', 23, 'girl', 'IT']

console.log(arr.push('beauty')); //5

console.log(arr); // ['mili', 23, 'girl', 'IT', 'beauty']

console.log(arr.pop()); //beauty

console.log(arr); // ['mili', 23, 'girl', 'IT']

从上面代码上可以看出，增加元素的push()/unshift()返回的都是增加后的arr的length，删除元素的pop()/shift()返回的都是删除的元素

@milixie

Owner

milixie commented on 9 Nov • edited

跨域的几种方式？

对于跨域，其实工作中没怎么遇到过，只是知道jsonp、iframe等方式都可以跨域，但是具体怎么实现的，就懵了

先了解下为什么会跨域，就是js出于安全性考虑，不允许跨域去调用其他页面的对象，也就是同源策略，即a.com域名下的js无法调用b.com或c.com域名下的文件，只有同一域名并且端口号、协议都相同时才可以允许通信，详情参考：允许通信表

跨域的具体方法如下：

1.document.domain+iframe的设置

适用于主域相同子域不同，例如：

www.baidu.com/a.html与yun.baidu.com/b.html

在a.html中创建一个iframe，去控制iframe的contentDocument

代码如下：

//www.baidu.com上的a.html

document.domain = 'baidu.com'; //domain只能设置主域名

var ifr = document.createElement('iframe');

ifr.src = "http://yun.baidu.com/b.html";

ifr.style.display = 'none';

document.body.appendChild(ifr);

ifr.onload = function(){

var doc = ifr.conentDocument || ifr.contentWindow.document;

//在这里操纵b.html

console.log(doc.getElementsByTagName('div')[0].innerHTML);

}

//yun.baidu.com/b.html

document.domain = 'baidu.com'

2.//jsonp方式：动态创建script

var oScript = document.createElement('script');

oScript.src = "http://yun.baidu.com/b.html?callback=foo";

oScript.onload = oScript.onreadystatechange = function(){

//加载完毕后服务端返回的是"foo(data)"所以，可以直接调用foo了

if (!this.readyState || this.readyState === 'loaded' || this.readyState === 'complete') {

// callback在此处执行

function foo(data){

console.log(data);

}

oScript.onload = oScript.onreadystatechange = null;

}

}

3.使用html5 postMessage：跨文档消息传输

otherWindow.postMessage(message, targetOrigin);

otherWindow:对接收信息页面的window的引用

message:所要发送的数据，string类型

targetOrigin:用于限制otherWindow，\*表示不做限制

a.com/index.html中：

<iframe id="ifr" src="http://b.com/index.html"></iframe>

<script type="text/javascript">

window.onload = function() {

var ifr = document.getElementById('ifr');

var targetOrigin = 'http://b.com'; // 若写成'http://b.com/c/proxy.html'效果一样

// 若写成'http://c.com'就不会执行postMessage了

ifr.contentWindow.postMessage('I was there!', targetOrigin);

};

</script>

b.com/index.html中：

<script type="text/javascript">

window.addEventListener('message', function(event){

//通过origin属性判断消息来源地址

if(event.origin == "http://a.com") {

alert(event.data); // I was there!

alert(event.source);

}

}, false)

</script>