



+ 学生服务器体验套餐 10元/月

· 1核2G · 1M带宽 · 50GB存储

立即抢购

专栏 / 文章详情



李小叔 RP 77

关注作者

2017-10-24 发布

春眠不觉晓，console知多少

原创 javascript 466 次阅读 · 读完需要 13 分钟

Console

console.log

首页

问答

专栏

讲堂

更多

console.log() 方法主要用来在控制台打印信息。其参数是不定的，可以传递任意多个参数，以逗号分割。

基本用法

```
console.log('hello');
console.log('hello', 'world');
console.log(obj1, obj2, document.getElementsByTagName('div'));
```

占位符

console.log() 提供了很多占位符，方便我们更加自如的定制我们的打印样式。

样式占位符	作用
%s	字符串占位符
%d 或者 %i	整数占位符
%f	浮点数占位符
%o	可展开的DOM
%O	列出DOM的属性
%c	根据提供的css样式格式化字符串（字符串格式的css样式）

这里仅以%c为例。



首页



问答



专栏

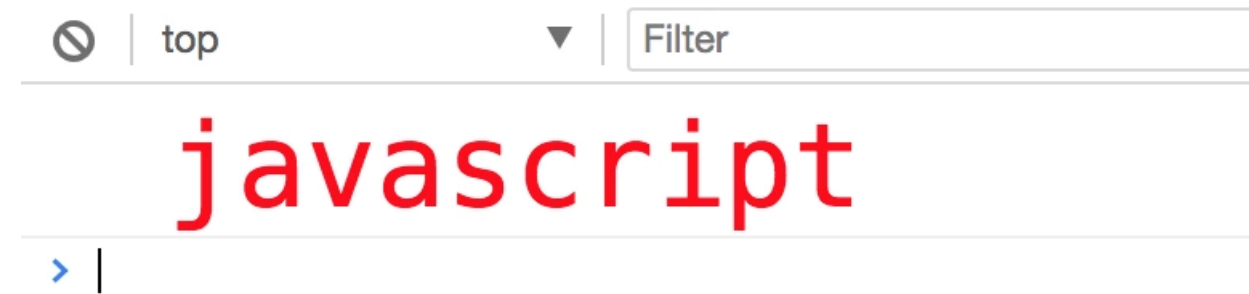


讲堂



更多

结果如下:



应用场景

制作图画

console.log 除了在控制台打印我们需要的调试信息以外，还可以制作一些图画。
如下（盗图）



 [首页](#)

专栏

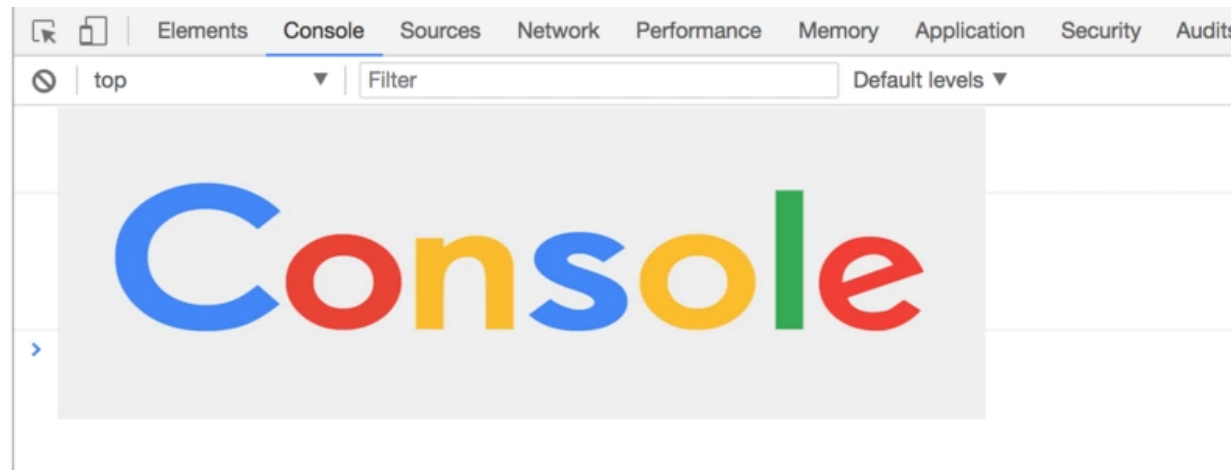
三
更多

控制台显示图片

另外，也可以使用`console.log`在控制台显示图片。

```
console.log('%c ', 'background: url(https://img.aotu.io/FiLwh8e3l--mGdsJNw3Bm-Z8o_JF) no-repeat center center;font-size:60px;padding:50px 200px;background-size:100% 100%;');
```

效果如下图：



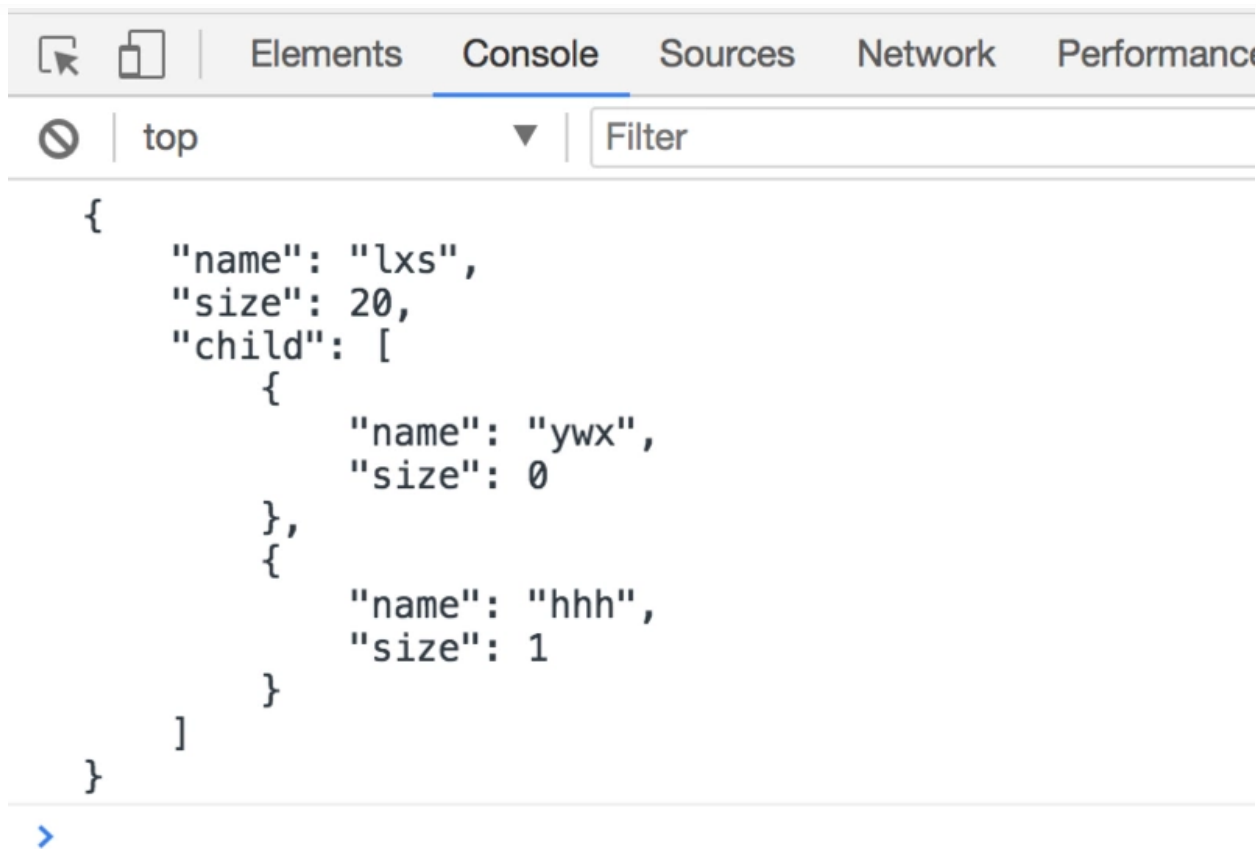
在控制台通过`console.log`打印图片，需要注意的是：

- 1、`console.log` 不能定义图片，因此借助于css背景图片；
- 2、`console.log` 不支持对图片进行`width`, `height`设置，但是我们可以通过 `font-size`, `padding`, `line-height`等属性来控制图片的大小。

最后, `console.log` 借助 `JSON.stringify` 可以对一个对象进行展开打印:

```
var obj = {  
  name: 'lxs',  
  size: 20,  
  child: [  
    {  
      name: 'ywx',  
      size: 0  
    },  
    {  
      name: 'hhh',  
      size: 1  
    }  
  ]  
};  
  
console.log(JSON.stringify(obj, null, 4));
```

打印效果如下:



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays a JSON object with the following structure:

```
{
  "name": "lxs",
  "size": 20,
  "child": [
    {
      "name": "ywx",
      "size": 0
    },
    {
      "name": "hhh",
      "size": 1
    }
  ]
}
```

Below the JSON object, there is a blue prompt character '>'.

console.time 与 console.timeEnd

`console.time` 可以开启一个定时器，来跟踪某一个（或者一组）操作所花费的时间，遇到同样参数的 `console.timeEnd` 方法时，会关闭这个定时器，并在浏览器控制台输出所花费的时间。

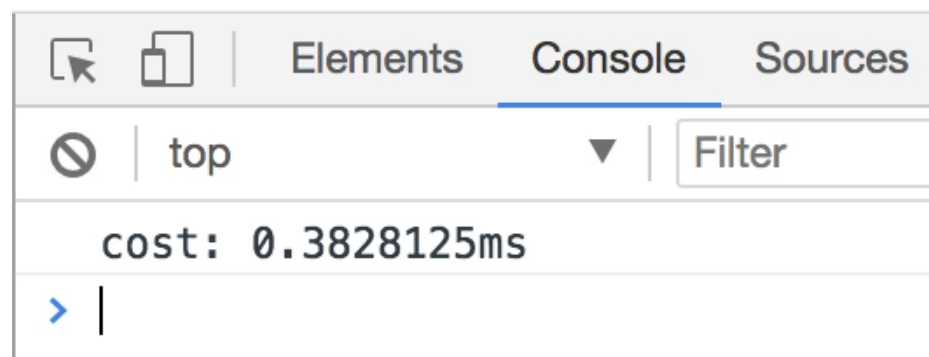
之前，我们想要知道某一些操作所花费的时间时，通常会这么做：

```
var start = new Date();  
...  
...  
...  
var end = new Date();  
console.log(end - start);
```

但是一个更方便快捷的方式就是使用 `console.time` 和 `console.timeEnd`

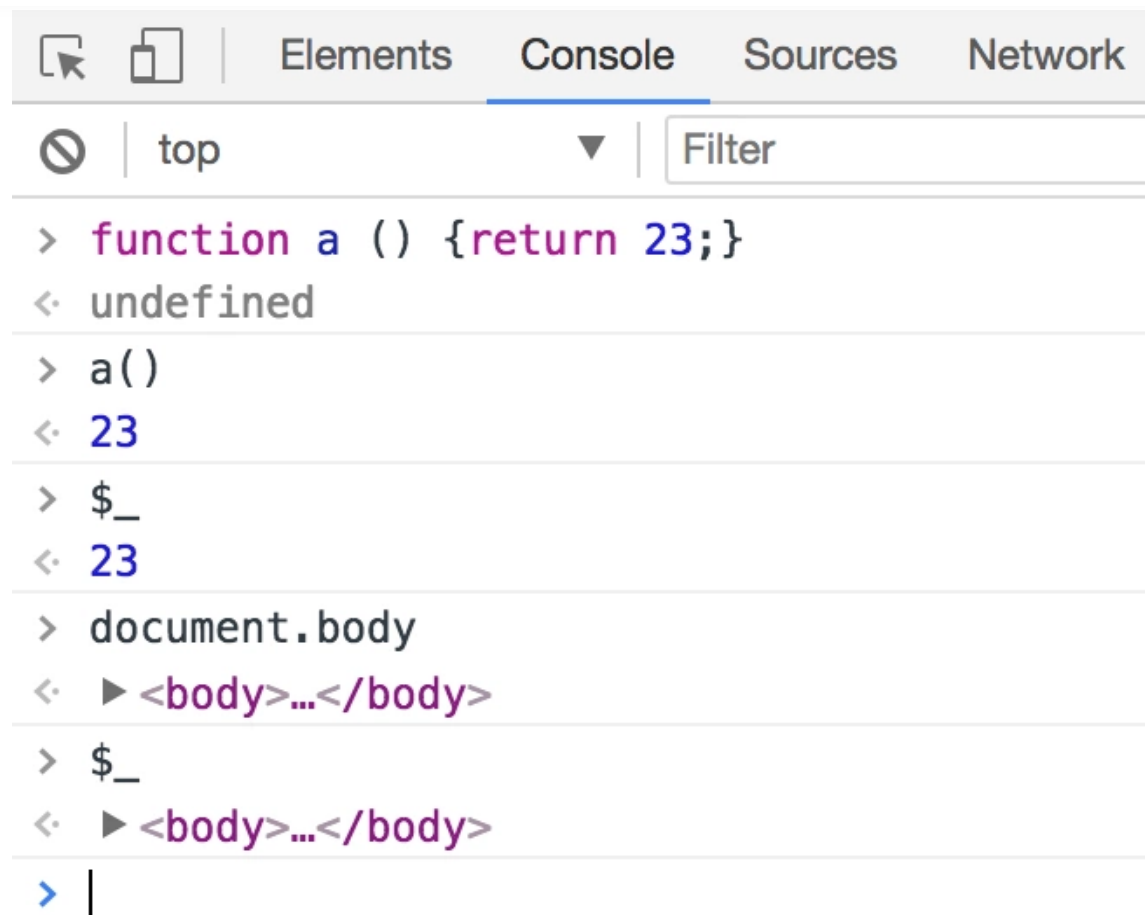
```
console.time('cost');  
for (var i = 0, j = 0; i < 10000; i++) {  
  j++;  
}  
console.timeEnd('cost');
```

结果如下：



`$_`

`$_` 代表上一个语句的执行结果：



```
> function a () {return 23;}
< undefined
> a()
< 23
> $_
< 23
> document.body
< ▶ <body>...</body>
> $_
< ▶ <body>...</body>
> |
```

快捷引用：\$i

代表的是inspect最近选中的5个DOM元素，

\$0是最近选中的一次，\$1...\$4以此类推。这个很有用，在elements面板中单击某个DOM后，在控制台直接输入\$0，就是对刚才单击元素的引用。

假设有如下DOM结构：

```
Elements Console Sources Network Performance Memory Application Security Audits
...<!DOCTYPE html> == $0
<html lang="en">
  <head>...</head>
  <body>
    <div id="J_content0" name="formContainer" class="wrap" data-price="1234">...</div>
    <div id="J_content1" name="formContainer" class="wrap" data-price="1234">...</div>
    <div id="J_content2" name="formContainer" class="wrap" data-price="1234">...</div>
    <div id="J_content3" name="formContainer" class="wrap" data-price="1234">...</div>
    <div id="J_content4" name="formContainer" class="wrap" data-price="1234">...</div>
    <div id="J_content5" name="formContainer" class="wrap" data-price="1234">...</div>
    <script>...</script>
  </body>
</html>
```

我们依次单击

div#J_content0, div#J_content1, div#J_content2, div#J_content3, div#J_content4, div#J_content5

再看控制台：

控制台中的\$0引用的是 div#J_content5，\$1引用的是倒数第二次点击的div#J_content4，... 到\$5 的时候报错了，是因为目前支持5个。

```
Elements Console Sources Network Performance Memory Application Security Audits
top Filter Default levels ▼
> $0
< > <div id="J_content5" name="formContainer" class="wrap" data-price="1234">...</div>
> $1
< > <div id="J_content4" name="formContainer" class="wrap" data-price="1234">...</div>
> $2
< > <div id="J_content3" name="formContainer" class="wrap" data-price="1234">...</div>
> $3
< > <div id="J_content2" name="formContainer" class="wrap" data-price="1234">...</div>
> $4
< > <div id="J_content1" name="formContainer" class="wrap" data-price="1234">...</div>
> $5
✖ > Uncaught ReferenceError: $5 is not defined
  at <anonymous>:1:1
```

快捷选择器：

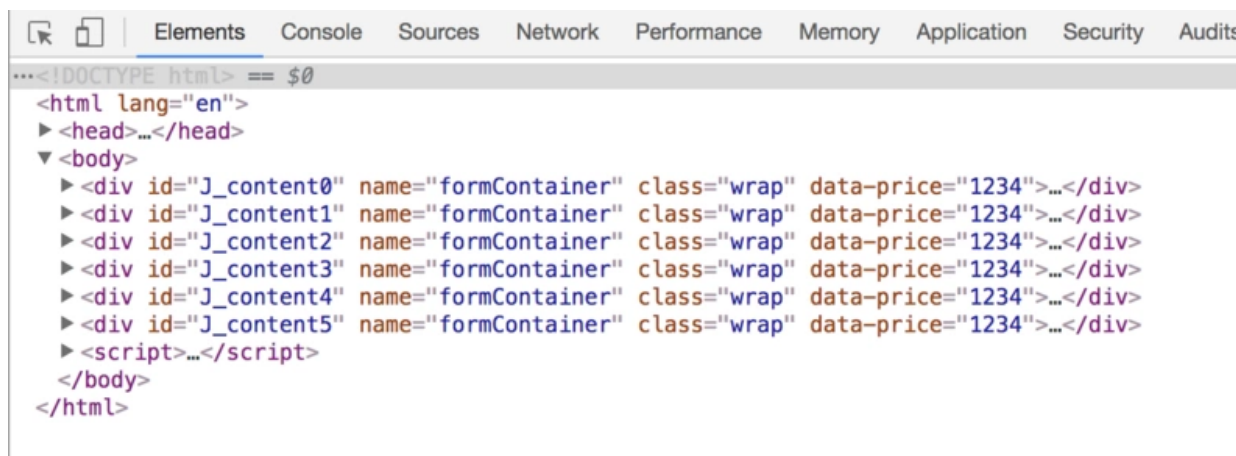
`$(selector)` 与 `$$ (selector)`

这里的\$并不是jquery的那个\$, 而是浏览器提供的一个 Command Line API.

`$(selector)`只匹配一个元素，其实是`document.querySelector`（如果支持）的别名；

`$$ (selector)`返回匹配的所有元素，其实是`document.querySelectorAll`（如果支持）的别名；

还是上面的DOM结构：



```
...<!DOCTYPE html> == $0
<html lang="en">
  <head>...</head>
  <body>
    <div id="J_content0" name="formContainer" class="wrap" data-price="1234">...</div>
    <div id="J_content1" name="formContainer" class="wrap" data-price="1234">...</div>
    <div id="J_content2" name="formContainer" class="wrap" data-price="1234">...</div>
    <div id="J_content3" name="formContainer" class="wrap" data-price="1234">...</div>
    <div id="J_content4" name="formContainer" class="wrap" data-price="1234">...</div>
    <div id="J_content5" name="formContainer" class="wrap" data-price="1234">...</div>
    <script>...</script>
  </body>
</html>
```

`$('.div')`

`$$ ('div')`

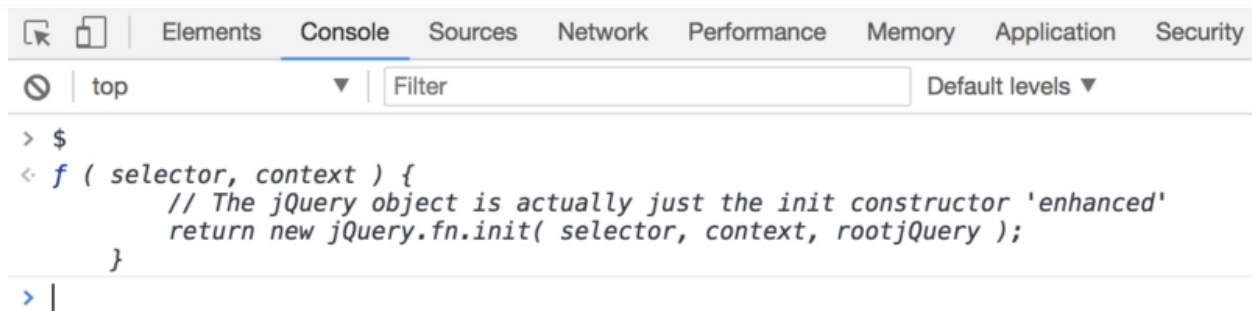
结果如下：

```
Elements Console Sources Network Performance Memory Application Security Audits
top Filter Default levels ▼
> $('div')
< ▶ <div id="J_content0" name="formContainer" class="wrap" data-price="1234">...</div>
> $$('div')
< ▼ (12) [div#J_content0.wrap, div.list, div#J_content1.wrap, div.list, div#J_content2.wrap, div
]
  ▶ 0: div#J_content0.wrap
  ▶ 1: div.list
  ▶ 2: div#J_content1.wrap
  ▶ 3: div.list
  ▶ 4: div#J_content2.wrap
  ▶ 5: div.list
  ▶ 6: div#J_content3.wrap
  ▶ 7: div.list
  ▶ 8: div#J_content4.wrap
  ▶ 9: div.list
  ▶ 10: div#J_content5.wrap
  ▶ 11: div.list
  length: 12
  __proto__: Array(0)
>
```

刚才说过，这里的\$并不是 jquery，zepto等库的\$，那是在页面没有引用这些库的情况下，如果页面使用了这些库，那么\$就会被覆盖。

未使用jquery，zepto等库：

```
Elements Console Sources Network Performance Mer
top Filter
> $
< f $(selector, [startNode]) { [Command Line API] }
>
```



The screenshot shows the Chrome DevTools console with the 'Console' tab selected. The top bar includes icons for back, forward, and search, along with tabs for Elements, Console, Sources, Network, Performance, Memory, Application, and Security. Below the tabs, there's a filter input set to 'top' and a 'Default levels' dropdown. The console log shows the following code:

```
> $  
< f ( selector, context ) {  
    // The jQuery object is actually just the init constructor 'enhanced'  
    return new jQuery.fn.init( selector, context, rootjQuery );  
}
```

clear

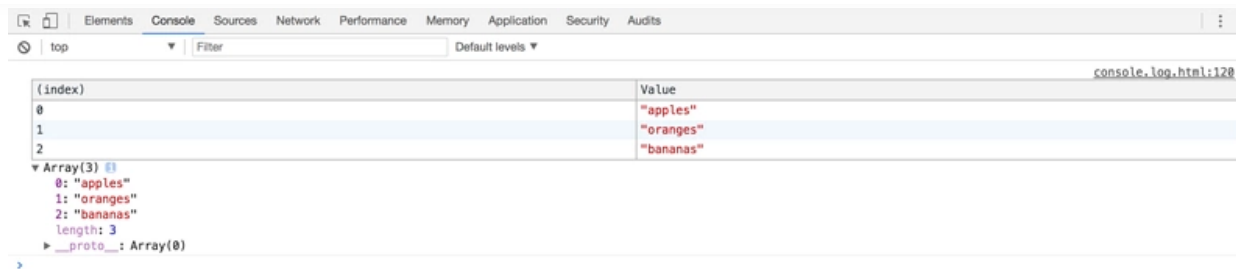
清除控制台的所有输出

table

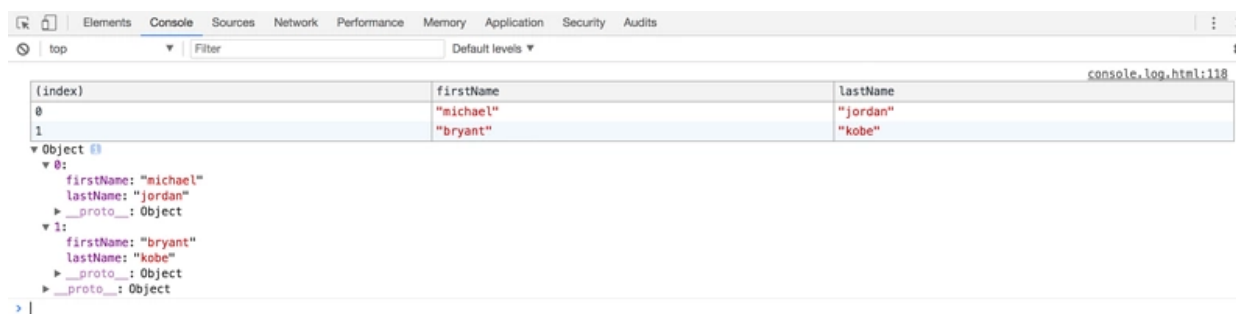
console.table 用于将数据以表格的形式输出。

```
console.table(["apples", "oranges", "bananas"]);
```

在控制台输出如下：



```
var names = {  
  0: {  
    firstName: 'michael',  
    lastName: 'jordan'  
  },  
  1: {  
    firstName: 'bryant',  
    lastName: 'kobe'  
  }  
};  
  
console.table(names, ['firstName', 'lastName']);
```



console.group 与 console.groupEnd

在控制台上新建一个分组，随后打印的内容都会被添加一个缩进，表示该内容属于当前分组，直到遇到 `console.groupEnd()` 之后，当前分组结束。

```
console.group();  
console.log(1);  
console.log(2);  
console.log(3);  
console.groupEnd();
```

```
console.group();  
console.log(4);  
console.log(5);  
console.log(6);  
console.groupEnd();
```



首页



问答



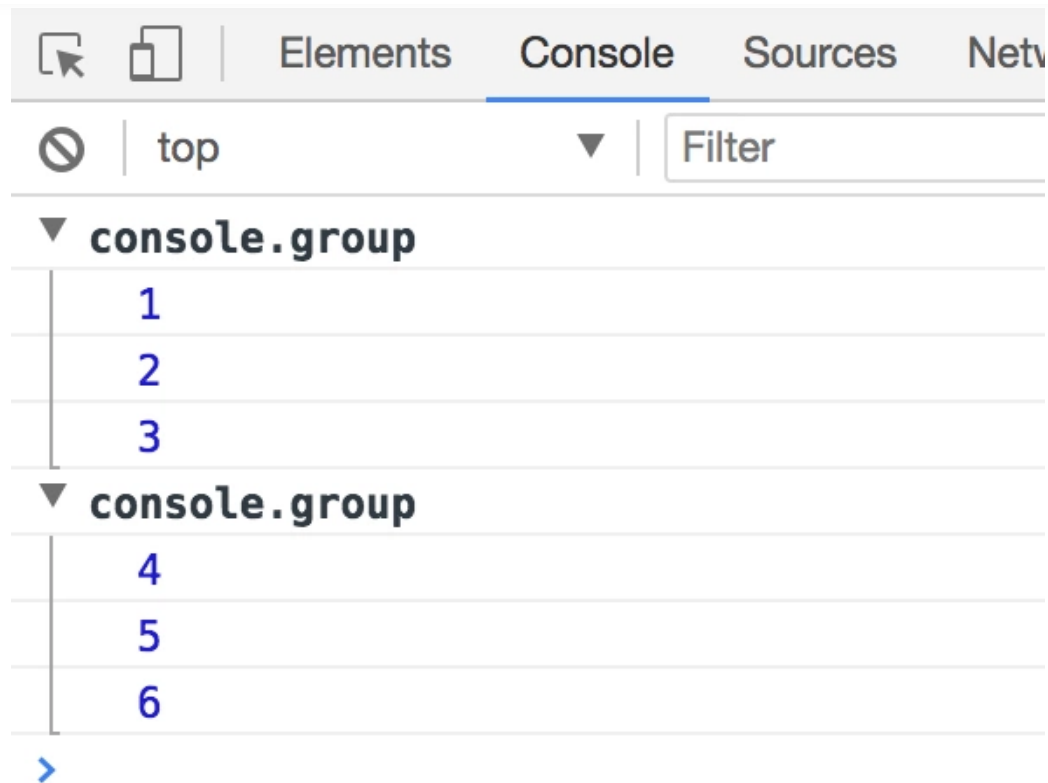
专栏



讲堂



更多



console.info

在控制台输出通知信息，用法和 `console.log` 一样。

console.warn

console.debug

console.log 的别名。

console.error

在控制台输出错误信息，用法和 console.log 一样。



赞 | 3

收藏 | 1



3分钟搭建微信小程序

1元起 零开发基础 轻松DIY

广告 X

你可能感兴趣的

- [如何使用JavaScript控制台改进工作流程](#) songfens typescript sass html5 html javascript
- [JavaScript Debug 的 Console](#) 极光推送 console debug javascript
- [javascript中console命令详解](#) specialCoder console javascript

- [JS进阶篇--JS之console.log详解以及兄弟姐妹们邻居方法扩展](#) 风雨后见彩虹 javascript console.log
- [解读生产环境为何避免使用console.log](#) 小皇帝James javascript
- [详解Object.create\(null\)](#) 司想君 javascript
- [console 非常实用的方法](#) hank javascript

评论

默认排序 时间排序



文明社会，理性评论

发布评论



3分钟搭建微信小程序

1元起 零开发基础 轻松DIY



产品

热门问答
热门专栏
热门课程
最新活动
技术圈
酷工作
移动客户端

课程

Java 开发课程
PHP 开发课程
Python 开发课程
前端开发课程
移动开发课程

资源

每周精选
用户排行榜
徽章
帮助中心
声望与权限
社区服务中心

合作

关于我们
广告投放
职位发布
讲师招募
联系我们

关注

产品技术日志
社区运营日志
市场运营日志
团队日志
社区访谈

条款

服务条款
内容许可



扫一扫下载 App



首页



问答



专栏



讲堂



更多

CDN 存储服务由 又拍云 赞助提供



首页



问答



专栏



讲堂



更多