

# **console.log(JSON.parse(JSON.stringify(data))) 这样写可以让data对象每一参数值都是默认展开的 不用每次都一个一个去点**

const blob1 = new Blob(['hello randy'], { type: "text/plain" });

undefined

const blob1 = new Blob(['hello randy'], { type: "text/plain" });

console.log(blob1)

VM723:1 Uncaught SyntaxError: **Identifier 'blob1' has already been declared**

at <anonymous>:1:1

控制台不定义（var,let,const）打印也不会报错

file = new File(["文件对象"], "test", { type: "text/plain" });

console.log(file);

# **多彩的 console.log**

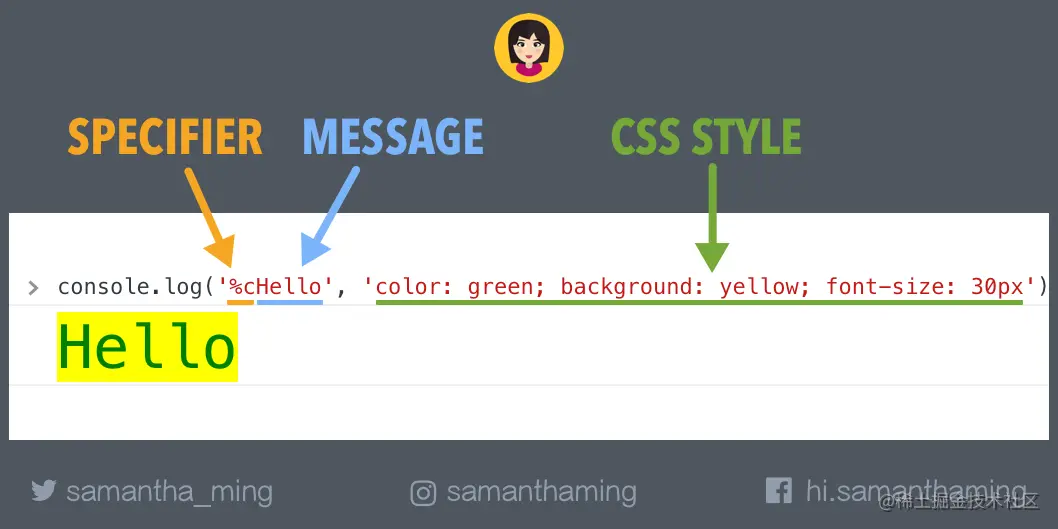
[](https://juejin.cn/user/3790771822272014)

[ylzsmallsun IMG_257](https://juejin.cn/user/3790771822272014" \t "https://juejin.cn/post/_blank)

2018年11月27日 22:35 ·  阅读 4763

关注

今天看到一篇很有意思的一篇文章，如何打印多彩的console.log? 前端的小伙伴对console.log再熟悉不过了，但是至今为止，我都是一直在用其最普通的用法，控制台中打印一条message。没想到，还能给console.log应用样式呢？不知道你们是否知道呢？



console.log('%cHello', 'color: green; background: yellow: font-size: 30px');复制代码

可以看出，上面的log语句由三部分组成： ****%c**** + message + ****style**** 其中标识符后紧跟message, 第二个参数为样式 最后输出的message的效果就如样式所定义的一致。

# 优点

当遇到一个具有大量log输出的大型应用时，如果在一些比较重要的地方输出带样式的log时，你可以在控制台中快速发现它，不至于淹没在一堆的log中难以发现查找。

# 栗子2

console.log(

'Nothing here %cHi Cat %cHey Bear', // Console Message

'color: blue', 'color: red' // CSS Style

);复制代码

效果如下图，标识符前面的文本不受影响。



# 五种类型的console message都可以添加样式

* console.log
* [console.info](https://link.juejin.cn/?target=http://console.info" \o "http://console.info" \t "https://juejin.cn/post/_blank)
* console.debug
* console.warn
* console.error

console.log('%cconsole.log', 'color: green;');

console.info('%cconsole.info', 'color: green;');

console.debug('%cconsole.debug', 'color: green;');

console.warn('%cconsole.warn', 'color: green;');

console.error('%cconsole.error', 'color: green;');复制代码

# 优雅的传递样式之实践

// 1. 将css样式内容放入数组

const styles = [

'color: green',

'background: yellow',

'font-size: 30px',

'border: 1px solid red',

'text-shadow: 2px 2px black',

'padding: 10px',

].join(';');

// 2. 利用join方法讲各项以分号连接成一串字符串

// 3. 传入styles变量

console.log('%cHello There', styles);复制代码

甚至，你还能把需要输出的message也抽离出来，保存在变量中

const styles = ['color: green', 'background: yellow'].join(';');

const message = 'Some Important Message Here';

// 3. 传入styles和message变量

console.log('%c%s', styles, message);