

今天做个简单的例子常见的前端 Ajax_Get/Post_Api 如何使用? (一)

作者: [Nice4z1](#)

原文链接: <https://ld246.com/article/1631066285584>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

写在前面

想写点好玩的东西，但又感觉要是写复杂了又比较难懂，所以还是做点简单的吧，所以，我们就简单举个例子，简单讲讲Ajax_Get/Post的使用，较为简单，做个了解即可，深入学习的的话还是建议先学Hml5+Css3然后Javascript+Es6+Jquery。

简单准备

我们就拿 “一言Api” 做一个简单Get的入门，然后再用 “彩虹云商城的Api” 做一个简单的Post请求
然后我们开始吧

一言Api

文档：<https://developer.hitokoto.cn/sentence/>

这边复制了我们要用到的接口和参数

请求地址

地址路	协议	方法	QPS 限制
v1.hitokoto.cn 球	HTTPS	Any	3.5
international.v1.hitokoto.cn 0 国外	HTTPS	Any	

请注意：一言域名已加入 HSTS 预加载计划，所有请求一律以 HTTPS 方法进行。

参数

完整参数太长可以到文档自行查看：[点我直达](#)

参数	值	可选	说明
c	默认随机	是	句子类型
encode 回编码	见官方文档	是	
charset	UTF-8	是	字符集
callback 的异步函数	如：moe	是	调
select 择器。配合encode=js 使用	默认：.hitokoto	是	
min_length 回句子的最小长度（包含）	默认：0	是	
max_length 回句子的最大长度（包含）	默认：30	是	

使用 一言Api

我们先新建个html文档 命名Demo.html

引用MDUI框架「力荐」[点我直达](#)

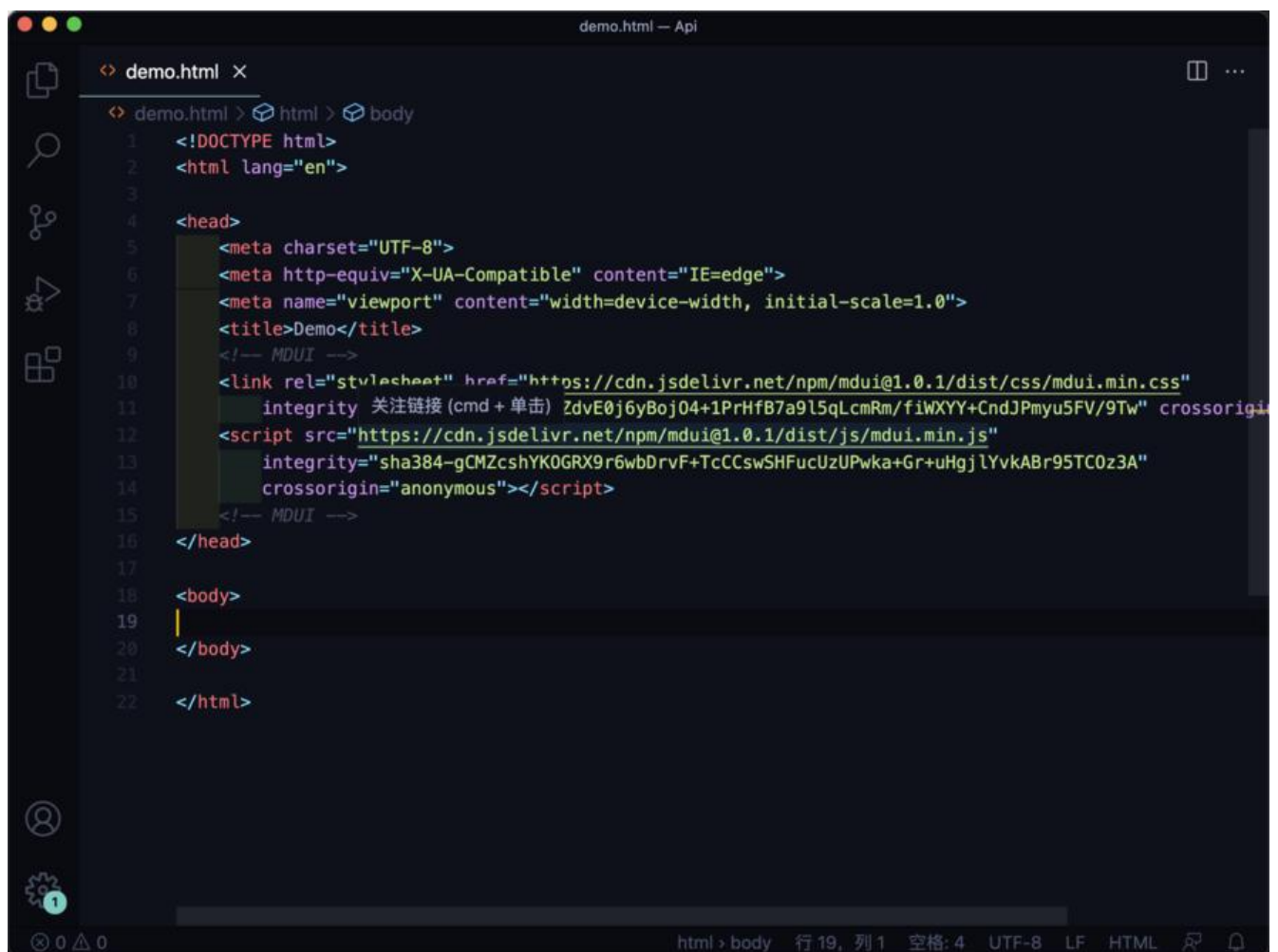
我们使用MDUI的JavaScript工具库足够帮助我们完成下面的简单开发

MDUI 内置了一个轻量的 JavaScript 工具库，它拥有和 jQuery 类似的 api 和 jQuery 风格的链式调用，但体积仅为 jQuery 的六分之一。

你可以通过 `mdui.$` 来调用该库，但最好把 `mdui.$` 存储到一个简短的变量中以方便调用，例如：

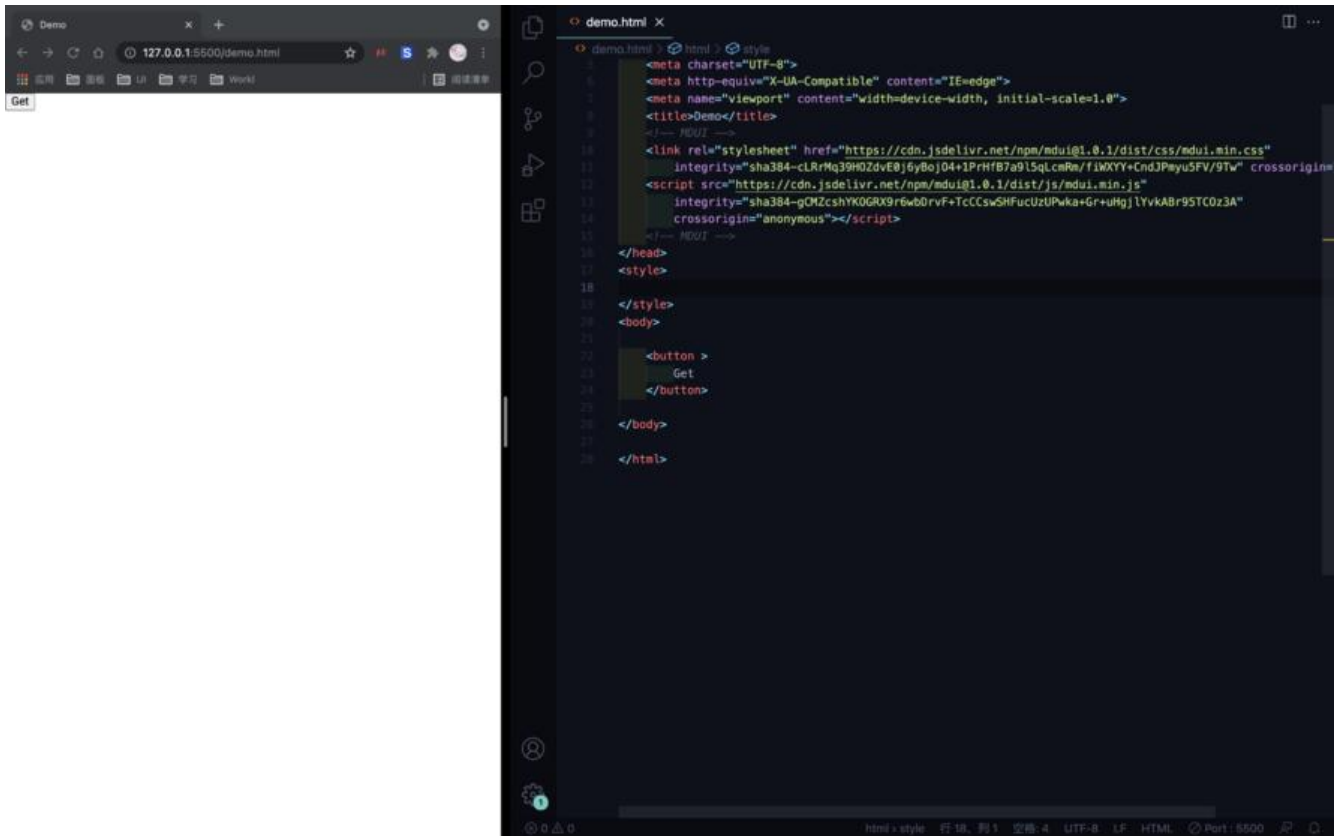
```
var $ = mdui.$;
```

在后面的文档中，都使用 `$` 来表示 `mdui.$`。

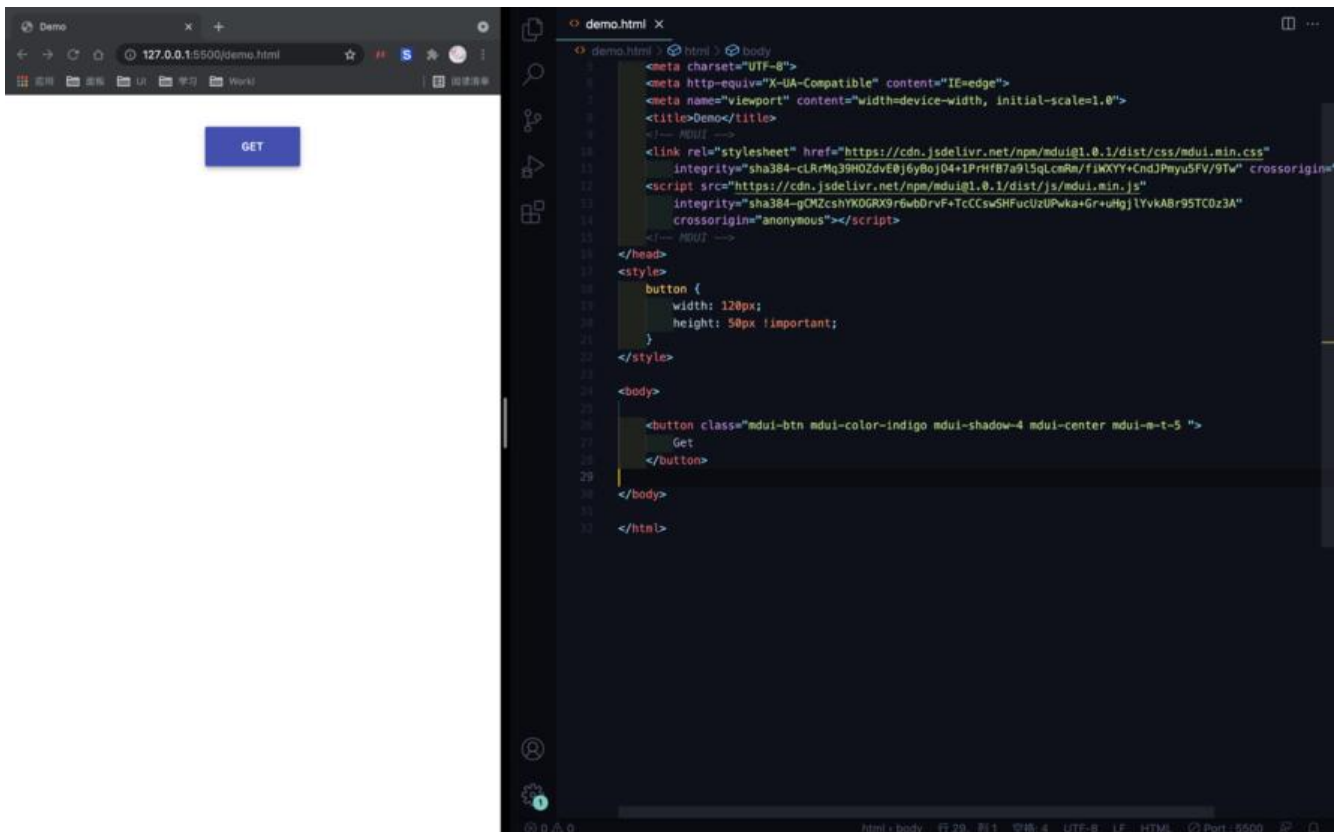


```
demo.html — Api
demo.html x
demo.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Demo</title>
9   <!-- MDUI -->
10  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/mdui@1.0.1/dist/css/mdui.min.css"
11      integrity 关注链接 (cmd + 单击) ZdvE0j6yBoj04+1PrHfB7a9L5qLcmRm/fiWXYy+CndJPmyu5FV/9Tw" crossorigin="
12  <script src="https://cdn.jsdelivr.net/npm/mdui@1.0.1/dist/js/mdui.min.js"
13      integrity="sha384-gCMZcshYK0GRX9r6wbDrvF+TcCCswSHFucUzUPwka+Gr+uHgjlYvkABr95TC0z3A"
14      crossorigin="anonymous"></script>
15  <!-- MDUI -->
16 </head>
17
18 <body>
19 |
20 </body>
21
22 </html>
```

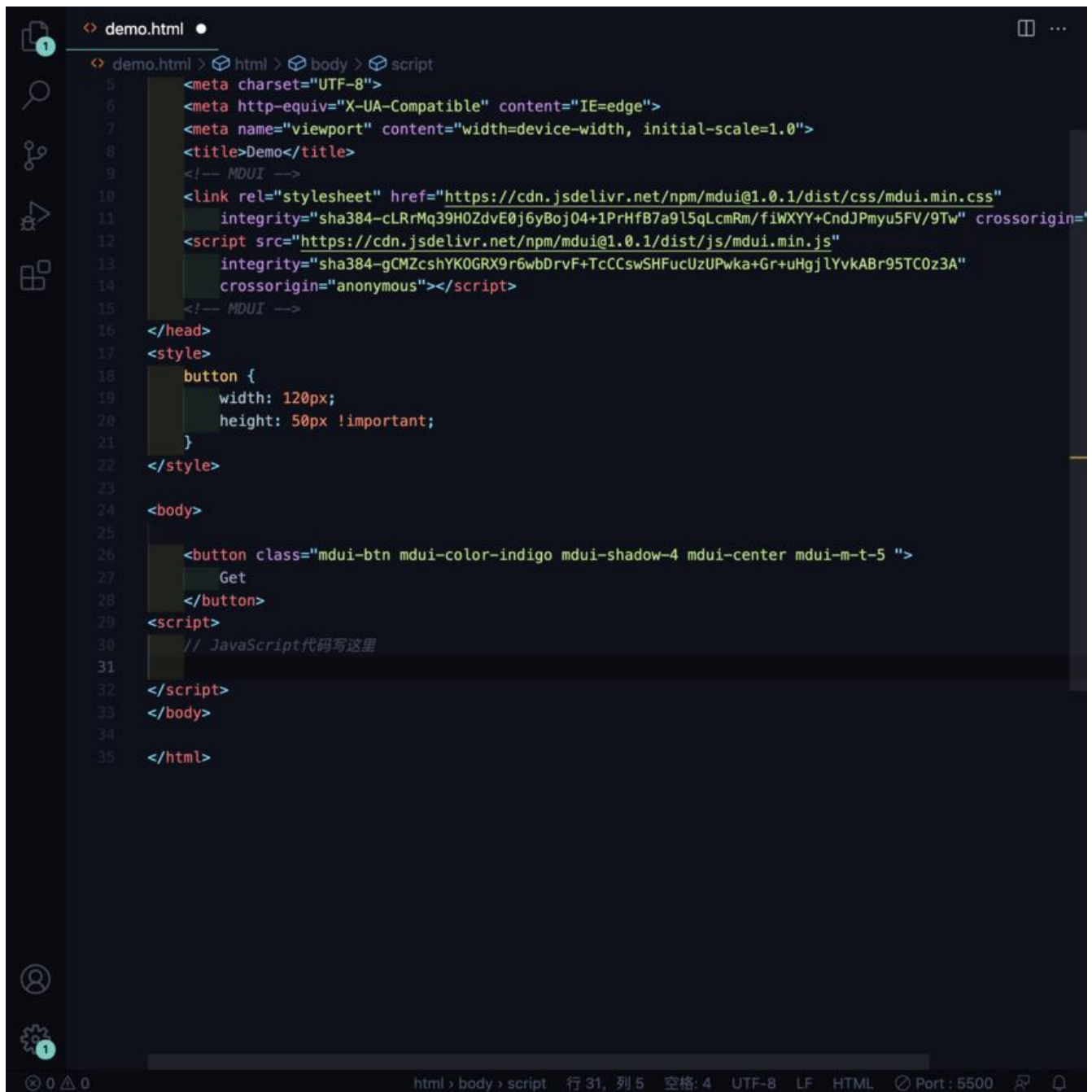
我们先给页面写一个按钮当作触发器，



然后加上简单的css让它稍微赏心悦目一点



然后我们开始写上JavaScript代码 但是前提是你加上 **script**标签



```
demo.html
demo.html > html > body > script
5 <meta charset="UTF-8">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
8 <title>Demo</title>
9 <!-- MDUI -->
10 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/mdui@1.0.1/dist/css/mdui.min.css"
11 integrity="sha384-cLRrMq39H0ZdvE0j6yBoj04+1PrHfB7a9l5qLcmRm/fiWXY+Y+CndJPmyu5FV/9Tw" crossorigin="
12 <script src="https://cdn.jsdelivr.net/npm/mdui@1.0.1/dist/js/mdui.min.js"
13 integrity="sha384-gCMZcshYK0GRX9r6wbDrvF+TcCCswSHFucUzUPwka+Gr+uHgjlYvkaBr95TC0z3A"
14 crossorigin="anonymous"></script>
15 <!-- MDUI -->
16 </head>
17 <style>
18 button {
19 width: 120px;
20 height: 50px !important;
21 }
22 </style>
23
24 <body>
25
26 <button class="mdui-btn mdui-color-indigo mdui-shadow-4 mdui-center mdui-m-t-5 ">
27 Get
28 </button>
29 <script>
30 // JavaScript代码写这里
31
32 </script>
33 </body>
34
35 </html>
```

实现思路

1. 点击按钮后，我需要得到 一言的 游戏类的句子
2. 然后我们要让它显示在html页面上

我们有了思路，就好实现了。

开始写Javascript，但是我们前面要按照MDUI提示的，

你可以通过 `mdui.$` 来调用该库，但最好把 `mdui.$` 存储到一个简短的变量中以方便调用，例如：

```
var $ = mdui.$;
```

在后面的文档中，都使用 `$` 来表示 `mdui.$`。

```
demo.html
demo.html > html > body
5 <meta charset="UTF-8">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
8 <title>Demo</title>
9 <!-- MDUI -->
10 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/mdui@1.0.1/dist/css/mdui.min.css"
11 integrity="sha384-cLRrMq39H0ZdvE0j6yBoj04+1PrHfB7a9l5qLcmRm/fiWXY+Y+CndJPmyu5FV/9Tw" crossorigin="
12 <script src="https://cdn.jsdelivr.net/npm/mdui@1.0.1/dist/js/mdui.min.js"
13 integrity="sha384-gCMZcshYK0GRX9r6wbDrvF+TcCCswSHFucUzUPwka+Gr+uHgjlYvkaBr95TC0z3A"
14 crossorigin="anonymous"></script>
15 <!-- MDUI -->
16 </head>
17 <style>
18 button {
19 width: 120px;
20 height: 50px !important;
21 }
22 </style>
23
24 <body>
25
26 <button class="mdui-btn mdui-color-indigo mdui-shadow-4 mdui-center mdui-m-t-5 ">
27 Get
28 </button>
29 <script>
30 // JavaScript代码写这里
31 var $ = mdui.$; //把mdui的JavaScript工具库存到这个简短的变量中
32 </script>
33 </body>
34
35 </html>
```

下面的代码我就——放图了，我直接代码贴在这里方便大家阅读。

忘了，我们得先给我们的按钮起个 id名字 这边以 `getData` 来命名

```
<button id="getData" class="mdui-btn mdui-color-indigo mdui-shadow-4 mdui-center mdui-m-t-5 ">
  Get
</button>
```

ok接下来，我们开始写JavaScript，可以认真看看。

我们再次回顾一下思路

点击按钮后，我需要得到 一言的 游戏类的句子

既然要点击按钮，我们就需要获取到这个按钮，那么

在MDUI中。有着和Jquery类似的写法

我们先把id叫 **getData**的按钮 获取过来

```
<script>
  // JavaScript代码写这里
  var $ = mdui.$; //把mdui的JavaScript工具库存到这个简短的变量中

  //正式代码
  $('#getData'); //获取名字为 getData 的按钮为什么要加 # 因为是id元素

</script>
```

如上面写的，但是呢，我们并不知道有没有成功获取，怎么办？

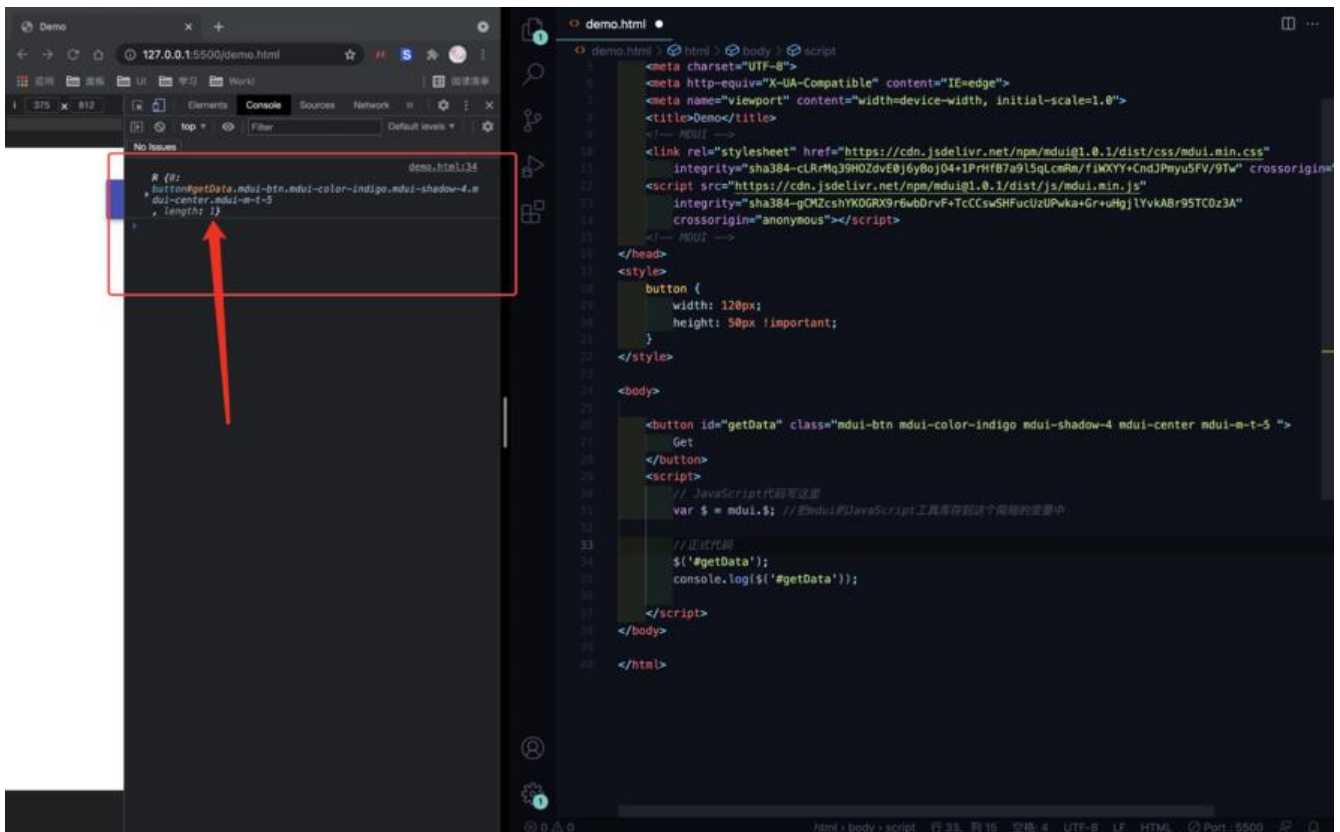
把它打印出来， **console.log**

```
<script>
  // JavaScript代码写这里
  var $ = mdui.$; //把mdui的JavaScript工具库存到这个简短的变量中

  //正式代码
  $('#getData');//获取名字为 getData 的按钮为什么要加 # 因为是id元素
  console.log($('#getData')); //打印验证一下 看看能不能获取

</script>
```

然后我们如何看有没有获取呢？ F12 浏览器**console**



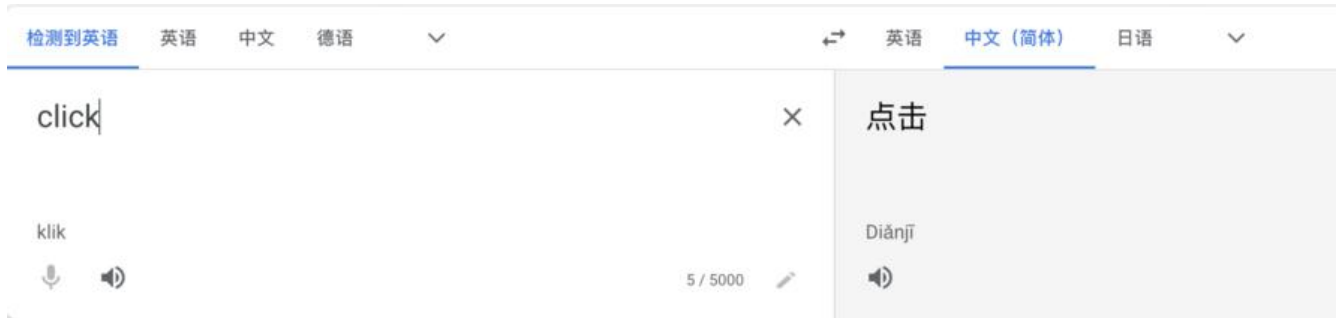
可以看到，我们获取到了，然后，我们给按钮绑定事件。也就是 `.on( )`;

里面有两个属性**`.on(怎么样触发它,触发了的返回值)`**

这么说可能不太清楚，我们用代码来表示一次

`$('#getData').on(怎么样触发它或者理解为触发条件,触发后返回什么或者做什么理解为);`

然后，我们要做的事点击事件，那么就是 `click`



ok,那么我们写上。

`$(selector).on('click', 触发后返回什么或者做什么理解为);`

我们写上了点击，那么，点击后该干什么呢？我们写上一个方法，`function () {}`

那么完整的写法，我们基本了解了，我们先写上，方便代码阅读，我们可以删掉之前的代码。

```
<script>
// JavaScript代码写这里
var $ = mdui.$; //把mdui的JavaScript工具库存到这个简短的变量中

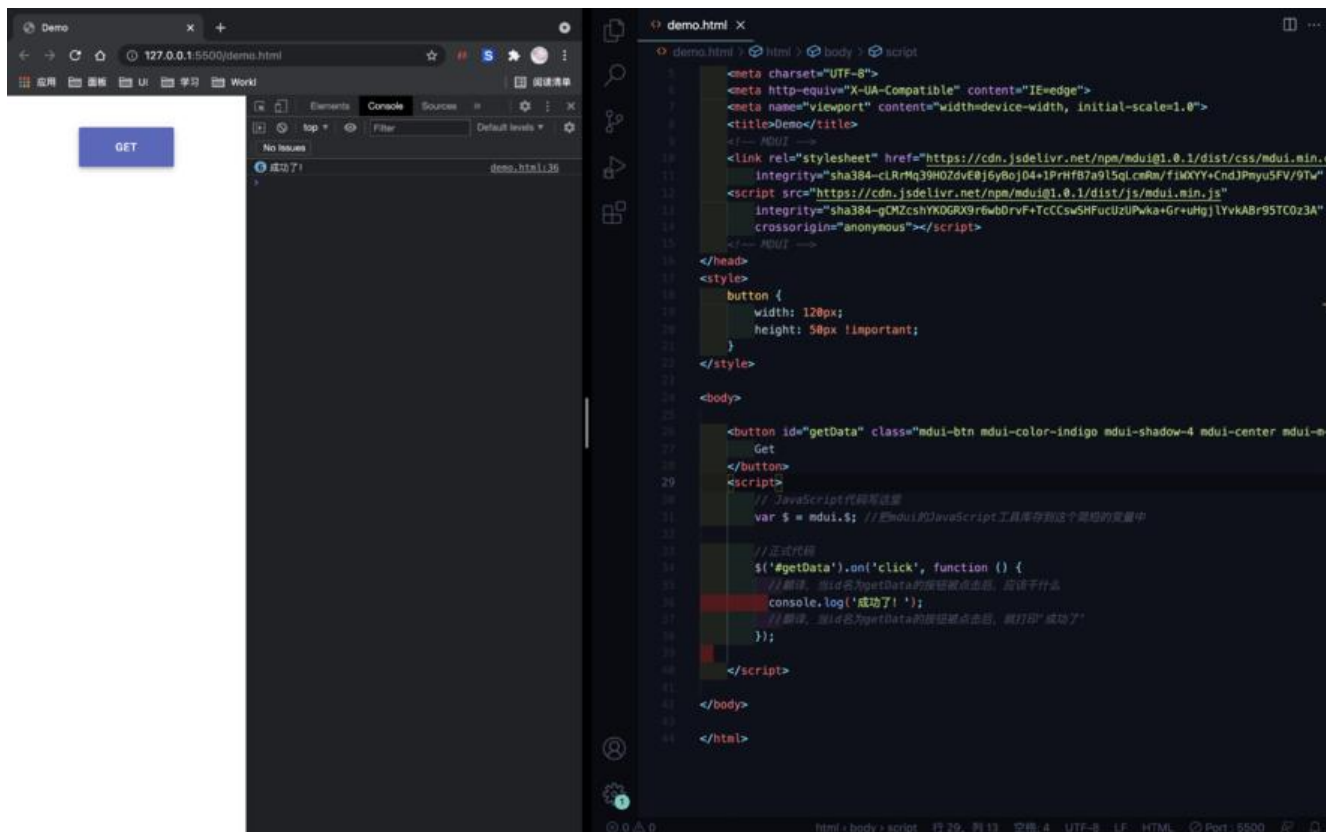
//正式代码
$('#getData').on('click', function () {
//翻译，当id名为getData的按钮被点击后，应该干什么
});
</script>
```

那么，我们也不知道有没有写对，怎么办？执行打印一下

```
<script>
// JavaScript代码写这里
var $ = mdui.$; //把mdui的JavaScript工具库存到这个简短的变量中

//正式代码
$('#getData').on('click', function () {
//翻译，当id名为getData的按钮被点击后，应该干什么
console.log('成功了! ');
//翻译，当id名为getData的按钮被点击后，就打印 '成功了'
});
</script>
```

然后 我们去点击按钮 试试。



证明是没有问题了，然后我们下一步吧

我们需要得到一言的游戏类型的句子，那么我们就需要用到ajax了，然后MDUI也是有Ajax的具体参，我们可以使用。

关于Ajax的介绍，可以百度百度，我们直只使用。

我们先去MDUI看看ajax的文档,太长，完整的建议去看看MDUI提供的文档



然后我们回到打印‘成功了！’那一块，

先删除点多余的打印代码和注释

```
<script>
  // JavaScript代码写这里
  var $ = mdui.$; //把mdui的JavaScript工具库存到这个简短的变量中
  //正式代码
  $('#getData').on('click', function () {
    //Ajax要放在这里面
  });
</script>
```

我们是做的事 按钮点击后执行，所以，Ajax需要放在这里面。

```
<script>
  // JavaScript代码写这里
  var $ = mdui.$; //把mdui的JavaScript工具库存到这个简短的变量中
  //正式代码
  $('#getData').on('click', function () {
    //Ajax要放在这里面
    $.ajax({
      method: 'POST', //请求方式 我们改为GET
      url: './test.php', //请求地址 我们改成一言的
      data: {           //请求时，我们需要携带的参数
        key1: 'val1',
        key2: 'val2'
      },
      success: function (data) { //请求成功后，返回的数值，或执行什么事件
        console.log(data); //打印成功后的返回值
      }
    });
  });
</script>
```

通过我上面写的信息，应该都简单的了解了Ajax，但是其实Ajax远远不止这些，但是本次文章讲的事单的用法。

ok我们对着一言文档改改。

句子类型（参数）

参数	说明
a	动画
b	漫画
c	游戏
d	文学
e	原创
f	来自网络
g	其他
h	影视

i	诗词
j	网易云
k	哲学
l	抖机灵
其他	作为 动画 类型处理

可选择多个分类，例如： `?c=a&c=c`

返回编码（参数）

参数	说明
text	返回纯洁文本
json	返回格式化后的 JSON 文本
js	返回指定选择器的同步调用函数。默认选择器
: .hitokoto	
其他	返回格式化后的 JSON 文本

然后看看我们改的

```

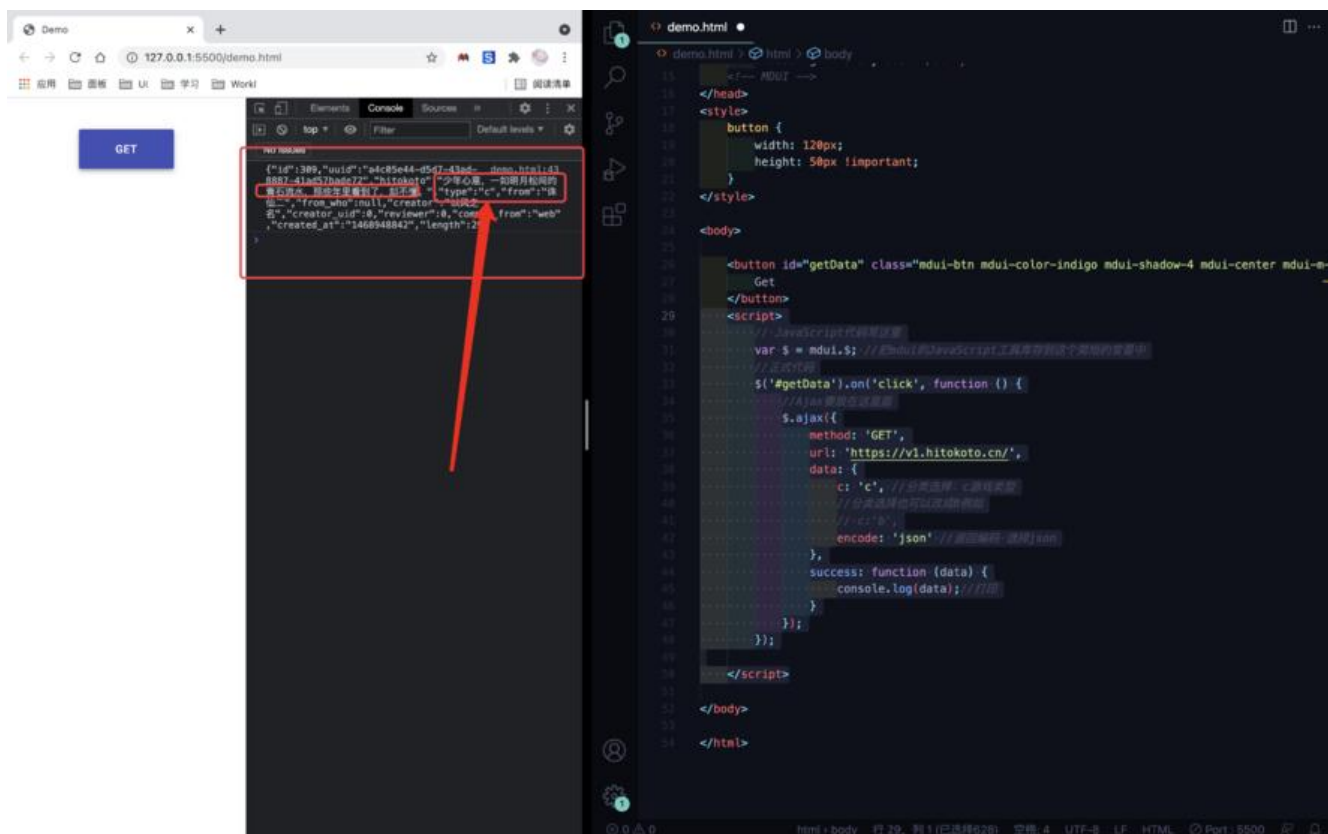
<script>
  // JavaScript代码写这里
  var $ = mdui.$; //把mdui的JavaScript工具库存到这个简短的变量中
  //正式代码
  $('#getData').on('click', function () {
    //Ajax要放在这里面
    $.ajax({
      method: 'GET',
      url: 'https://v1.hitokoto.cn/',
      data: {
        c: 'c', //分类选择: c游戏类型
        //分类选择也可以改成B例如
        // c:'b',
        encode: 'json' //返回编码 选择json
      },
      success: function (data) { //成功后执行
        console.log(data); //打印
      }
    });
  });
</script>

```

整体大概就是这样，

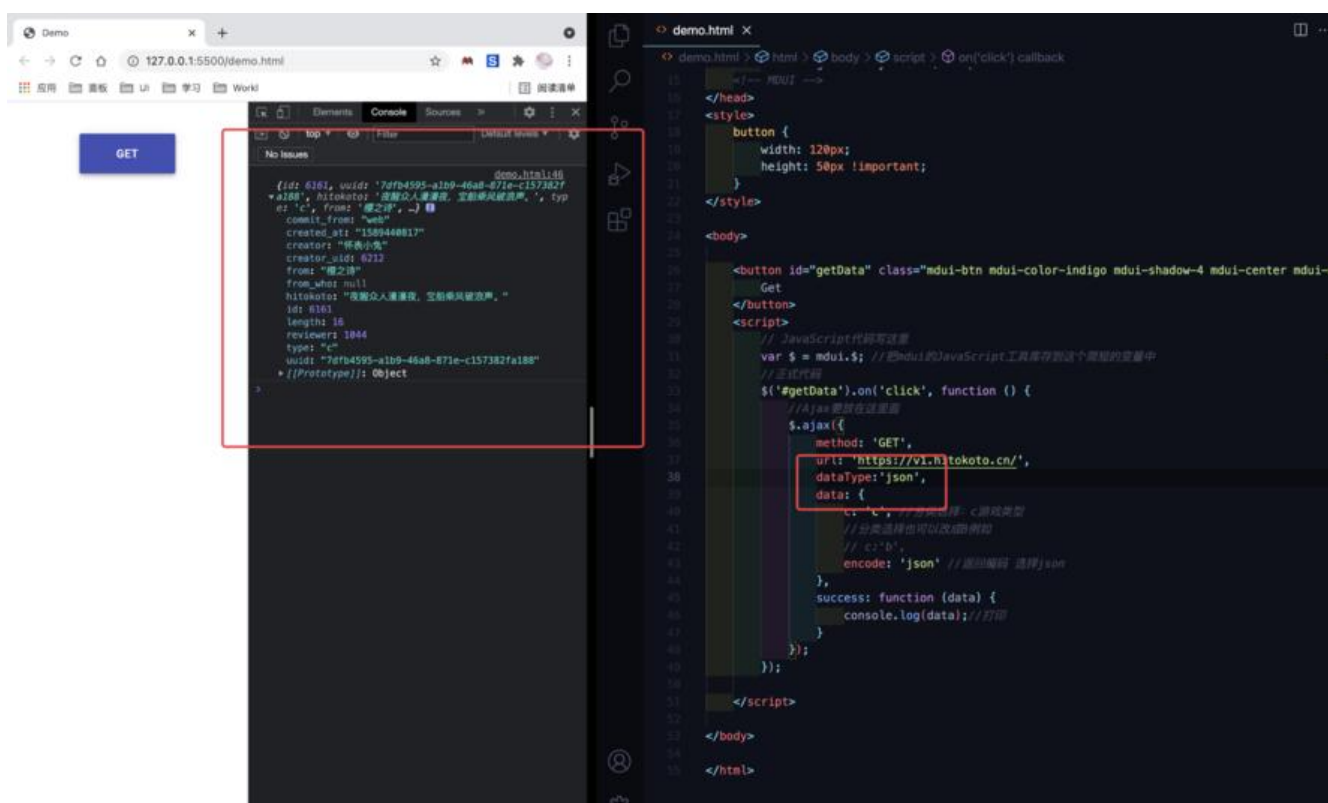
获取按钮，点击按钮，执行Ajax事件，得到成功返回值

那么ok，改完我们直接点击按钮执行一次。



我们确实得到了，但是非常混乱，没有得到我想要的句子，那么我们就需要进行再次转为json

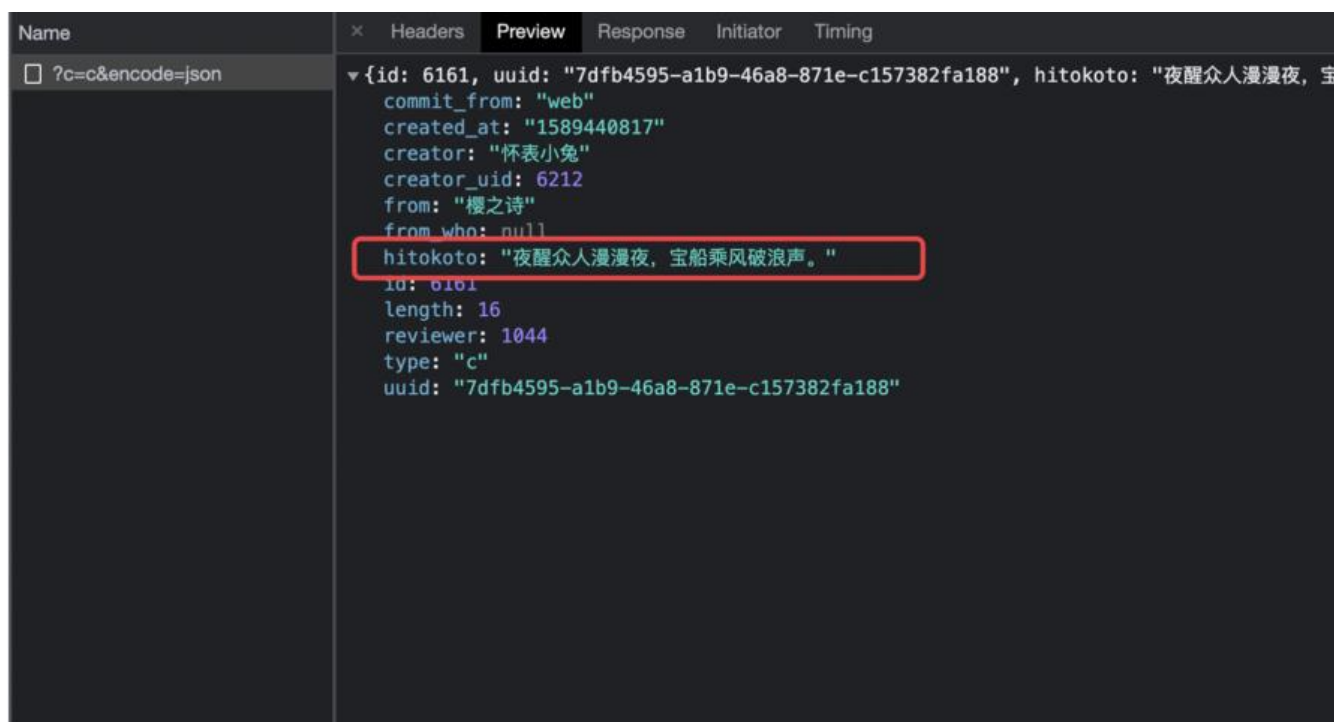
dataType:'json' 然后再执行一下，就清晰多了



但是这并没有解决我们的问题，我们要的是句子，显示这么多不该显示的，还是太乱了，怎么办？

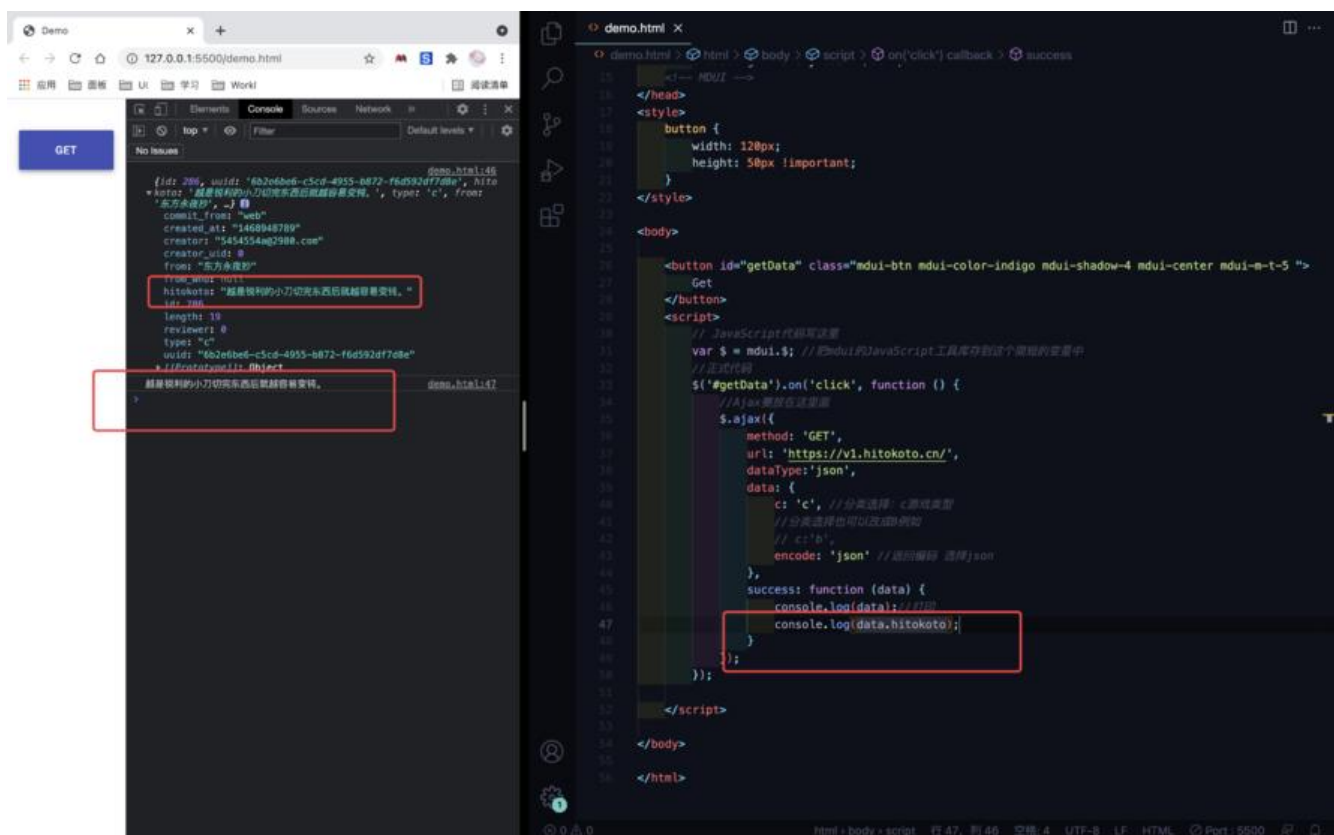
那么就到返回值这一块处理了

返回值是返回给我们一整个 **data**



但是我们不需要啊，我们需要的是 **data**里面的hitokoto 也就是句子

那么，我们就在返回值 打印出 **data.hitokoto**



这样 就干净多了~ 思路

1.点击按钮后，我需要得到 一言的 游戏类的句子

2.然后我们要让它显示在html页面上

我们已经实现了第一步，那么ok，如何第二步，如何让它显示在html页面上？

很简单，我们新建个标签 **span** 取个类名，就叫 text吧

阅读MDUI提供的文档后

`.text()`

获取集合中所有元素的文本内容（包含它们的后代元素）

```
// 获取所有 .box 元素的文本
$('.box').text();
```

也可以是设置集合中所有元素的文本。

设置的值可以是字符串、数值、布尔值、回调函数。

若为回调函数，第一个参数为元素的索引位置，第二个参数为元素原有的文本内容，函数内的 `return`

若设置的值、或回调函数返回值为 `undefined`，则不修改对应元素的文本。

```
// 设置选中元素的文本内容
$('.box').text('text content');

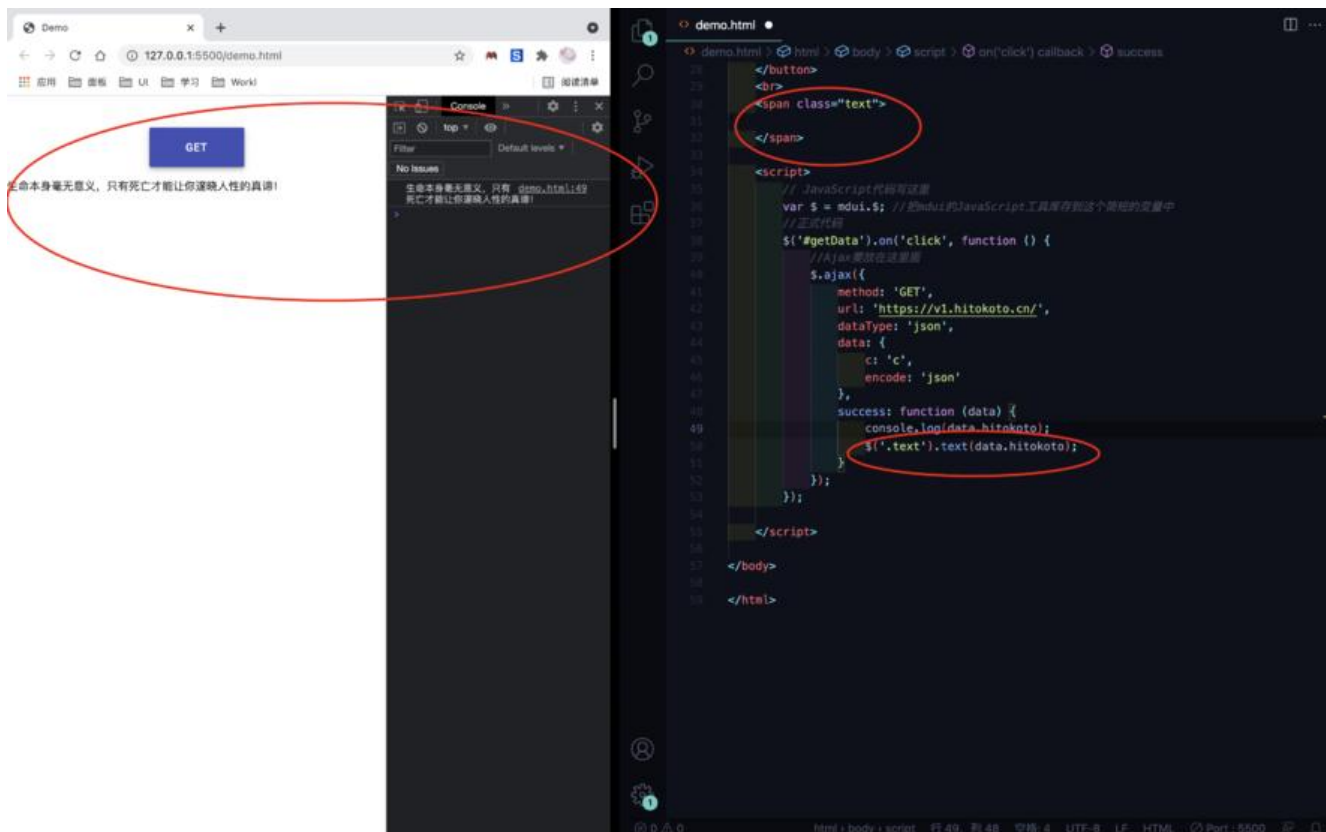
// 通过回调函数的返回值设置元素的文本内容
$('.box').text(function (index, oldTextContent) {
    return 'new text content';
});
```

使用`.text()`就能实现了！

上面学了怎么获取元素，那么道理相同

获取元素类名为text的元素，然后把它的text的值改为 data.hitokoto 即可。

我们来实现吧~



至此，一言Api的简单例子就做完了，然后如果认真看了的话 那么你就比较简单的掌握了ajax的最为单的用法，还顺带学了点js。

下面附上完整JavaScript代码

```
<script>
// JavaScript代码写这里
var $ = md5.$; //把md5的JavaScript工具库存到这个简短的变量中
//正式代码
$('#getData').on('click', function () {
    //Ajax要放在这里面
    $.ajax({
        method: 'GET',
        url: 'https://v1.hitokoto.cn/',
        dataType: 'json',
        data: {
            c: 'c',
            encode: 'json'
        },
        success: function (data) {
            console.log(data.hitokoto);
            $('#text').text(data.hitokoto);
        }
    });
});
</script>
```

其实一言api提供了不少例子，对多少学过JavaScript的还是简单的，这次写这个，只是抛砖引玉罢了。

有更多更复杂的api等着你使用。

下一个例子，有空再更新。

如果你学会了如何使用，那么你就可以自己写一个属于自己的个人主页什么的，就可以去网上自己找get的api玩。

对了，如果你删掉了下面红色方框框住的，那么就不用点击按钮，直接执行ajax哦~

```
<script>
  // JavaScript代码写这里
  var $ = mdui.$; //把mdui的JavaScript工具库存到这个简短的变量中
  //正式代码
  $('#getData').on('click', function () {
    //Ajax要放在这里面
    $.ajax({
      method: 'GET',
      url: 'https://v1.hitokoto.cn/',
      dataType: 'json',
      data: {
        c: 'c',
        encode: 'json'
      },
      success: function (data) {
        console.log(data.hitokoto);
        $('#.text').text(data.hitokoto);
      }
    });
  });
</script>
```

其实一言api提供了不少例子，对多少学过JavaScript的还是简单的，这次写这个，只是抛砖引玉罢了。

这边就不放成品了，好好学习，天天向上~