

学号	姓名	提交时间	总分	成绩
20225102010129	杨嘉钰	2024-05-22 11:22:43	100	-

1. 【简答题】 写出jquery ajax

得分：0/100

学生答案：引入 jQuery 库：在使用 jQuery 进行 AJAX 请求之前，需要确保在 HTML 文档中引入了 jQuery 库。

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

编写 AJAX 请求：使用 jQuery 的 \$.ajax() 方法来发送请求。

```
$.ajax({
  url: 'url_to_request', // 服务器请求的URL
  type: 'GET', // 请求方式，例如GET或POST
  dataType: 'json', // 期望的服务器响应的数据类型
  success: function(data) {
    // 请求成功，处理响应数据
    console.log(data);
  },
  error: function(error) {
    // 请求失败，处理错误
    console.error('An error occurred:', error);
  }
});
```

配置请求参数（如果是POST请求）：可以在 \$.ajax() 方法中配置 data 属性来发送数据。

```
$.ajax({
  url: 'url_to_request',
  type: 'POST',
  data: {
    key1: 'value1',
    key2: 'value2'
  },
  success: function(data) {
    console.log(data);
  },
  error: function(error) {
    console.error(error);
  }
});
```

设置请求头：如果需要自定义请求头，可以使用 beforeSend 属性。

```
$.ajax({
  url: 'url_to_request',
  type: 'POST',
  beforeSend: function(xhr) {
    xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
  },
  success: function(data) {
    console.log(data);
  },
  error: function(error) {
    console.error(error);
  }
});
```

处理异步：jQuery 的 AJAX 请求默认是异步的，如果你需要同步请求，可以设置 async: false。

```
$.ajax({
  url: 'url_to_request',
  async: false, // 同步请求
  success: function(data) {
    console.log(data);
  },
  error: function(error) {
    console.error(error);
  }
});
```

使用简写方法：jQuery 还提供了一些简写方法，如 \$.get() 和 \$.post()，用于快速发送 GET 和 POST 请求。

GET 请求示例：

```
$.get('url_to_request', function(data) {
  console.log(data);
});
```

参考答案：1