```
exec()函数用于检索字符串中的正则表达式的匹配。
                                               如果字符串中有匹配的值,则返回该匹配值,否则返回 null。
                              exec函数
                                                  var str='hello'
                                                   var pattern=/o/
                                                   var result=pattern.exec(str)
                                                   console.log(result);
                                             正则表达式中()包起来的内容表示一个分组,可以通过分组来提取自己想要的内容,
                              分组
                                            var str=' <div>我是{{name}}</div>'
                                               var\ pattern = /\{\{([a-zA-Z]+)\}\}/
                                               var result=pattern.exec(str)
                                               console.log(result);
模板引擎 -
                 正则
                                                 用于在字符串中用一些字符替换另一些字符
                                                 var str=' <div>我是{{name}}</div>'
                                                    var pattern=/{{([a-zA-Z]+)}}/
var result=pattern.exec(str)
                                                    // 0 : "{{name}}"
                                                    // 1: "name"
                                                    str=str.replace(result[0],result[1])
                                                    console.log(str);
                              replace函数
                                                                   var str1='<div>我是{{name}}今年{{age}}岁/div>'
                                                                   var \ pattern 1 = / \{ (s*([a-zA-Z]+))s* \} \} /
                                                                   var endresult=null
                                                                       while(endresult=pattern1.exec(str1)){
                                                 多次replace -
                                                                        str1=str1.replace(endresult[0],endresult[1])
                                                                       console.log(str1);
                                         XMLHttpRequest (简称 xhr) 是浏览器提供的 Javascript 对象,通过它,可以请求服务器上的数据资源。之
                                         前所学的 jQuery 中的 Ajax 函数,就是基于 xhr 对象封装出来的。
                                                                // 1.创建xhr对象
                                                                    var xhr=new XMLHttpRequest()
                                                                    // 2.调用open函数
                                                                    xhr.open('GET','http://www.liulongbin.top:3006/api/getbooks')
                                                                    // 3.调用send函数发起请求
                                                                    xhr.send()
                                                                    // 4.监听 onreadystatechange 事件
                                                                    xhr.onreadystatechange=function(){
                                                                      // 这个是固定写法
                                        使用xhr发起get请求
                                                                      if(xhr.readyState===4\&xhr.status===200)\{\\
                                                                       // 获取服务器响应的数据
                                                                        console.log(xhr.responseText);\\
                                                                                          在调用xhr.open期间 为URL地址指定参数
                                                               _ 发起带参数的get请求 ___
                                                                                          xhr.open('GET','http://www.liulongbin.top:3006/api/getbooks?id=1')\\
Ajax高级
                 {\sf XMLHttpRequest}
                                                          是指在URL末尾加上用于向服务器发送信息的字符串(变量)
                                                           格式:将英文的?放在URL的末尾,然后再加上参数=值,想加上多个参数的话,使用&符号进行分隔。以
                                                           这个形式,可以将想要发送给服务器的数据添加到 URL 中。
                                                             URL编码 __
                                                                           _ encodeURI()
                                        URL编码与解码
                                                             URL解码 __
                                                                 var xhr=new XMLHttpRequest()
                                                                     xhr.open('POST','http://www.liulongbin.top:3006/api/addbook')
                                                                     // 3.设置Content-Type属性
                                                                     xhr.set Request Header ('Content-Type', 'application/x-www-form-urlencoded')\\
                                                                     // 4.调用send函数
                                                                     xhr.send('bookname=水浒传&&author=施耐庵&&publisher=上海图书出版社')
                                        使用xhr发起post请求 __
                                                                     xhr.onreadystatechange=function(){
    if(xhr.readyState===4&&xhr.status===200){
                                                                        console.log(xhr.responseText);
```