```
作用:使用模板引擎提供的模板语法,可以将数据和 HTML 拼接起来。
                                                                                 〈script src="./js/template-web.js"></script> —— 1. 下载 art-template 模板引擎库文件并在 HTML 页面中引入库文件
                                                                                                               <script id="tpl" type="text/html">
                                                                                                                                                                                                                                                                       ●网速慢的情况下,页面加载时间长,用户只能等待
                                                                                                                   〈div class="box"></div> ── 2. 准备 art-template 模板
                                                                                                                                                                                                                                                                      ●表单提交后,如果一项内容不合格,需要重新填写所有表单内容 页面跳转,
                                                                                                                                                                                                                                                   传统网站中体验不足的地方 —
                                                                                                                                                                             - 客户端模板引擎
                                                                                                                                                                                                                                                                       重新加载页面,造成资源浪费,增加用户等待时间。
                                                                                var html = template('tpl', {username: 'zhangsan', age: '20'}); — 3. 告诉模板引擎将哪一个模板和哪个数据进行拼接 — 使用步骤 —
                                                                                           document.getElementById('container').innerHTML = html; ———— 4. 将拼接好的html字符串添加到页面中
                                                                                                                                                                                                                                                             它是浏览器提供的一套方法,可以实现页面无刷新更新数据,提高用户浏览网站
                                                                                     <script id="tpl" type="text/html">
                                                                                                                                                                                                                                                   Ajax概述
                                                                                                                                                                                                                                                             应用的体验。
                                                                                        <form id="form">
                                                                                                                                                                                                                                                               1. 页面上拉加载更多数据
                                                                 <input type="text" name="username" />
                                                                                                                                                                                                                                                   A.jax应用场景 —— 2.列表数据无刷新分页
                                                                 Formdata 对象不能用于 get 请求,因为对象需要被传递到 send 方法中,而
                                                                                                                                                                                                                                                               3. 表单项离开焦点数据验证 搜索框提示文字下拉列表
                                                                 <input type="button"/>
                                                                                                                       1. 模拟HTML表单,相当于将HTML表单映射成表单对象,自动将表单对象中的数据拼接
get 请求方式的请求参数只能放在请求地址的后面。
                                                                                                                      成请求参数的格式。
                                                                                                                                                                          - FormData对象的作用
                                                    ·服务器端 bodyParser 模块不能解析 formData 对象表单数据,我们需要使用
                                                                                                                                                                                                                                                                Ajax 技术需要运行在网站环境中才能生效, 当前课程会使用Node创建的服务器
formidable 模块进行解析。
                                                    var formData = new FormData(form);
                                                                                                                                                                                                                                                   ajax运行环境
                                                                                                                                                                                                                                                                作为网站服务器。
                                                                                  xhr.send(formData); —— 3. 提交表单对象
                                                                                                                                                         2. 异步上传二进制文件
                                                                                                                                                                                                                                                                Ajax 相当于浏览器发送请求与接收响应的代理人,以实现在不影响用户浏览页面
                                                                                                                             formData.get('key'); — 1. 获取表单对象中属性的值
                                                                                                                                                                                                                                                                的情况下,局部更新页面数据,从而提高用户体验。
                                                                                                                       formData.set('key', 'value'); — 2. 设置表单对象中属性的值
                                                               注意: set 方法与 append 方法的区别是,在属性名已存在的情况下,set 会覆
                                                                                                                                                                         FormData对象的实例方法
                                                                盖已有键名的值, append会保留两个值。
                                                                                                                           formData.delete('key'); — 3. 删除表单对象中属性的值
                                                                                                                      formData.append('key', 'value'); —— 4. 向表单对象中追加属性值
                                                                                                                                                                                                                                                   Ajax运行原理
                                                                                                                         // 当用户选择文件的时候
                                                                                                                          file.onchange = function () {
                                                                                                                              // 文件上传过程中持续触发onprogress事件
                                                                                                                              xhr.upload.onprogress = function (ev) {
                                                                                                                                    // 当前上传文件大小/文件总大小 再将结果转换为百分数
                                                                                                                                                                            文件上传进度展示
                                                                                                                                                                                                                                                                          开发人员不可控
                                                                                                                                                                                                                                                                                                               开发人员可控
                                                                                                                                    // 将结果赋值给进度条的宽度属性
                                                                                                                                   bar. style. width = (ev. loaded / ev. total) * 100 + '%';
                                                                                                                          Ajax 只能向自己的服务器发送请求。比如现在有一个A网站、有一个B网站, A网站中
                                                                                                                                                                                                                                                                    ar xhr = new XMLHttpRequest();
                                                                                                                          的 HTML 文件只能向A网站服务器中发送 Ajax 请求,B网站中的 HTML 文件只能向 B
                                                                                                                          网站中发送 Ajax 请求, 但是 A 网站是不能向 B 网站发送 Ajax请求的, 同理, B
                                                                                                                          网站也不能向 A 网站发送 Ajax请求。
                                                                                                                                                                                                                                                                    nr.open('get', 'http://localhost:3000/first');
                                                                                                               如果两个页面拥有相同的协议、域名和端口,那么这两个页面就属于同一个源,其中只_
                                                                                                              要有一个不相同,就是不同源。
                                                                                                                                                                                                                                                   Ajax的实现步骤
                                                                                         http://www.example.com/dir2/other.html: 同源
                                                                                                                                                                   一 什么是同源
                                                                                         http://example.com/dir/other.html:不同源(域名不同)
                                                                                         http://v2.www.example.com/dir/other.html:不同源(域名不同) — http://www.example.com/dir/page.html
                                                                                                                                                                                                                                                                    nr.onload = function (){
                                                                                         http://www.example.com:81/dir/other.html:不同源(端口不同)
                                                                                                                                                                                                                                                                     console.log(xhr.responseText)
                                                                                         https://www.example.com/dir/page.html:不同源(协议不同)
                                                                                                             同源政策是为了保证用户信息的安全,防止恶意的网站窃取数据。最初的同源政策是指_
                                                                                                             A 网站在客户端设置的 Cookie, B网站是不能访问的。
                                                                                                                                                                 同源政策的目的 -
                                                                                                             随着互联网的发展,同源政策也越来越严格,在不同源的情况下,其中有一项规定就是___
                                                                                                                                                                                                                                                                      在真实的项目中,服务器端大多数情况下会以 JSON 对象作为响应数据的格式。
                                                                                                             无法向非同源地址发送Ajax 请求,如果请求,浏览器就会报错。
                                                                                                                                                                                                                                                                      当客户端拿到响应数据时,要将 JSON 数据和 HTML 字符串进行拼接,然后将
                                                                                                      jsonp 是 json with padding 的缩写,它不属于 Ajax 请求,但它可以模拟 Ajax
                                                                                                                                                                                                                                                                      拼接的结果展示在页面中。
                                                                                                                                                                                                                                                   服务器端响应的数据格式
                                                                              <script src="www.example.com"></script>
                                                                                                                                                                                                                                                                     在 http 请求与响应的过程中,无论是请求参数还是响应内容,如果是对象类
                                                                                                          一 1. 将不同源的服务器端请求地址写在 script 标签的 src 属性中
                                                         <script src= "https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js">
                                                                                                                                                                                                                                                                     型,最终都会被转换为对象字符串进行传输。
                                                       script>
                                                                                                                                                          使用JSONP解决同源限制问题
                                                                 const data = 'fn({name: "张三", age: "20"})'; ____ 2. 服务器端响应数据必须是一个函数的调用,真正要发送给客户端的数据需要作
                                                                                                                                                                                                                                                                     JSON.parse() // 将 json 字符串转换为json对象
                                                                 res.send(data);
                                                                                                                                                                                                                                                                get请求方式 ____ xhr.open('get', 'http://www.example.com?
                                                                                                        function fn(data){} —— 3. 在客户端全局作用域下定义函数 fn
                                                                                                                                                                                                                                                                            name=zhangsan&age=20);
                                                                                    function fn (data) { console.log(data); } —— 4. 在 fn 函数内部对服务器端返回的数据进行处理
                                                                                                           CORS: 全称为 Cross-origin resource sharing, 即跨域资源共享, 它允许浏览器
                                                                                                                                                                                                                                                                            xhr.setRequestHeade1('Content-Type', 'application/x-www-form-
                                                                                                                                                            -- CORS跨域资源共享
                                                                                                           向跨域服务器发送 Ajax 请求,克服了 Ajax 只能同源使用的限制。
                                                                                                                                                                                                                                                                            urlencoded') xhr.send('name=zhangsan&age=20);
                                                                                                                                                                                                                                                   请求参数的传递
                                                                                                                                                                                                                                                                post请求方式
                                                                                                                                在跨域请求的过程中不会自动的携带cookie -
                                                                                                                                                                                                                                                                                       在 HTTP 请求和响应的过程中传递的数据块就叫报文,包括要传送的数据和一些
                                                                                                        withCredentials: 指定在涉及到跨域请求时,是否携带cookie信息,默认值为false —— 跨域请求中有关cookie
                                                                                                                                                                                                                                                                            请求报文
                                                                                                                                                                                                                                                                                      附加信息,这些数据和信息要遵守规定好的格式。
                                                                                                                                                                               Ajax请求限制 -
                                                                                                服务器端 —— Access-Control-Allow-Credentials: true 允许客户端发送请求时携带cookie
                                                                                                                                                作用:发送Ajax请求。
                                                                                                                                                                                                                                                                                                                           E请求头中指定 Content-Type 属性的值是 application/json,告诉服务器端当
                                                                                                                                                                                                                                                                - 1. application/x-www-form-urlencoded ---- name=zhangsan&age=20&sex=男
                                                                                                                                                                                                                                                                                                                           前请求参数的格式是 json。 JSON.stringify() // 将json对象转换为json
                                                                                                                      $.ajax({
                                                                                                                                                                                                                                                   请求参数的格式
                                                                                                                                                                                                                                                                                                                           <sup>2符串</sup> 注意:get 请求是不能提交 json 对象数据格式的,传统网站的表单提交。
                                                                                                                           type: 'get',
                                                                                                                                                                                                                                                                2. application/json —— {name: 'zhangsan', age: '20', sex: '男'}
                                                                                                                                                                                                                                                                                                                          也是不支持 json 对象数据格式的。
                                                                                                                            url: 'http://www.example.com',
                                                                                              data: 'name=zhangsan&age=20'
                                                                                                                           data: { name: 'zhangsan', age: '20' },
                                                                                                                                                                                                                                                                   已经过时但是任然有人在使用, 此知识点作为了解
                                                                                                                            contentType: 'application/x-www-form-urlencoded',
                                                                                                                            beforeSend: function () {
                                                                                                                                                                 ─ $.ajax()方法
                                                                                                                                                                                                                                                                              在创建ajax对象,配置ajax对象,发送请求,以及接收完服务器端响应数据,这
                                                                                                                                                                                                           Class12:前后端交互Ajax
                                                                                           contentType: 'application/json'
                                                                                                                                 return false
                                                                                                                                                                                                                                                                              个过程中的每一个步骤都会对应一个数值,这个数值就是ajax状态码。
                                                                                                                            success: function (response) {},
                                                                                   SON.stringify({name: 'zhangsan', age: '20'})
                                                                                                                           error: function (xhr) {}
                                                                                                                                                                                                                                                                              0:请求未初始化(还没有调用open())
                                                                                                                                                                                                                                                   - 获取服务器端的响应 —
                                                                                                                                                                                                                                                                              1:请求已经建立,但是还没有发送(还没有调用send())
                                                                                                                                                                                                                                                                 Ajax状态码
                                                                                                                                                                                                                                                                             - 2:请求已经发送
                                                                                              在封装的$.ajax()方法中有个dataType属性设置属性的值为 字符串的 jsonp —— $.ajax方法发送jsonp请求
                                                                                                                                                                                                                                                                              3:请求正在处理中,通常响应中已经有部分数据可以用了
                                                                                                                             作用:将表单中的数据自动拼接成字符串类型的参数
                                                                                                                                                                                                                                                                              4:响应已经完成,可以获取并使用服务器的响应了
                                                                                                                                                                 serialize方法
                                                                                                                                var params = $('#form').serialize();
                                                                                                                                                                                                                                                                                nr.readyState // 获取Ajax状态
                                                                                                                                // name=zhangsan&age=30
                                                                                                                                                                                                                                                                              onreadstatechange()事件 — 当Ajax对象的状态(即状态码)发生变化时触发该事件。
                                                                                                              作用: $.get方法用于发送get请求, $.post方法用于发送post请求。
                                                                                                                                                                                                                                                               1. 网络畅通,服务器端能接收到请求,服务器端返回的结果不是预期结果。 可
                                                                                                                                                                                                                                                               以判断服务器端返回的状态码,分别进行处理。xhr.status 获取http状态码
                                                                                                       $.get('http://www.example.com, {name: 'zhangsan', age: 30},
                                                                                                                                                            - $.get()和$.post()方法 -
                                                                                                       function (response) {}) $.post('http://www.example.com',
                                                                                                                                                                                                                                                               2. 网络畅通,服务器端没有接收到请求,返回404状态码。
                                                                                                       {name: 'lisi', age: 22}, function (response) {})
                                                                                                                                                                                                                                                                检查请求地址是否错误。
                                                                                                                                                                                                                                                   Ajax错误处理
                                                                                                                                 只要页面中有Ajax请求被发送,对应的全局事件就会被触发
                                                                                                                                                                                                                                                               3. 网络畅通,服务器端能接收到请求,服务器端返回500状态码。 服务器端错
                                                                                                                                                                                                                                                               误,找后端程序员进行沟通。
                                                                                                                                        .ajaxStart()   // 当请求开始发送时触发
                                                                                                                                         .ajaxComplete() // 当请求完成时触发
                                                                                                                                                                          jQuery中Ajax全局事件
                                                                                                                                                                                                                                                               4. 网络中断,请求无法发送到服务器端。
                                                                                                                                                                                                                                                                会触发xhr对象下面的onerror事件,在onerror事件处理函数中对错误进行处
                                                                                                         document).on('ajaxStart', function () {
                                                                                                                                          一 Ajax全局事件要求绑定在document对象的身上 -
                                                                                                                                        官宣: 纳米进度条, 使用逼真的涓流动画告诉用户正在发生的事情一
                                                                                                                                                                                                                                                                         问题: 在低版本的 IE 浏览器中, Ajax 请求有严重的缓存问题, 即在请求地址不
                                                                                                                                        <link rel='stylesheet' href='nprogress.css'/>
                                                                                                                                                                                                                                                                         发生变化的情况下,只有第一次请求会真正发送到服务器端,后续的请求都会从
                                                                                                                                                                                                                                                                         浏览器的缓存中获取结果。即使服务器端的数据更新了,客户端依然拿到的是缓
                                                                                                                                         <script src='nprogress.js ></script>
                                                                                                                                                                                 NProgress
                                                                                                                                                                                                                                                                        存中的旧数据。
                                                                                                                                                NProgress.start(); // 进度条开始运动
                                                                                                                                                                                                                                                   低版本IE浏览器中的缓存问题
                                                                                                                                                 NProgress.done(); // 进度条结束运动
                                                                                                                                                                                                                                                                        解决方案: 在请求地址的后面加请求参数, 保证每一次请求中的请求参数的值不
                                                                                                                                                                                                                                                                        相同。
                                                                                                                                                       一套关于设计请求的规范。
                                                                                                                                                                                                                                                                        xhr.open('get', 'http://www.example.com?t='+ Math.random());
                                                                                                                                                           GET: 获取数据
                                                                                                                                                           POST:
                                                                                                                                                                 添加数据
                                                                                                                                                                            RESTful风格的API
                                                                                                                                                                                                                                                                                   ● 一个人同一时间只能做一件事情,只有一件事情做完,才能做另外一件事情。
                                                                                                                                                                                                                                                                                  ● 落实到代码中,就是上一行代码执行完成后,才能执行下一行代码,即代码逐
                                                                                                                                                           PUT: 更新数据
                                                                                                                                                           DELETE: 删除数据
                                                                                                                        GET: <a href="http://www.example.com/users">http://www.example.com/users</a>
                                                                                                                                                   获取用户列表数据
                                                                                                                                                                                                                                                                                  ●一个人一件事情做了一半,转而去做其他事情,当其他事情做完以后,再回过
                                                                                                                                                                                                                                                   Ajax异步编程 —— 同步异步概念
                                                                                                                        POST: <a href="http://www.example.com/users">http://www.example.com/users</a> 创建(添加)用户数据
                                                                                                                                                                                                                                                                                   头来继续做之前未完成的事情。
                                                                                                                         ●落实到代码上,就是异步代码虽然需要花费时间去执行,但程序不会等待异步
                                                                                                                        PUT: <a href="http://www.example.com/users/1">http://www.example.com/users/1</a> 修改用户ID为1的用户信息
                                                                                                                                                                                                                                                                                   代码执行完成后再继续执行后续代码,而是直接执行后续代码,当后续代码执行
                                                                                                                        DELETE: <a href="http://www.example.com/users/1">http://www.example.com/users/1</a> 删除用户ID为1的用户信息
                                                                                                                                                                                                                                                                                   完成后再回头看异步代码是否返回结果,如果已有返回结果,再调用事先准备好
                                                                                                                                                                                                                                                                                   的回调函数处理异步代码执行的结果。
                                                                                                                             XML 的全称是 extensible markup language,代表可扩展标记语言,它的作用
                                                                                                                             是传输和存储数据。
                                                                                                                                                                                                                                                            - 问题: 发送一次请求代码过多,发送多次请求代码冗余且重复。
                                                                                                                                                         <students>
                                                                                                                                                                                                                                                            一解决方案:将请求代码封装到函数中,发请求时调用函数即可。
                                                                                                                                                            <student>
                                                                                                                                                               <sid>001</sid>
                                                                                                                                                              <name>张三</name>
                                                                                                                                                               </student>
                                                                                                                                                                                  XML基础
                                                                                                                                      保存了两个学生的信息 -
                                                                                                                                                                                                                                                  - Ajax封装
                                                                                                                                                           <student>
                                                                                                                                                                                                                                                                type: 'get',
                                                                                                                                                                                                                                                                 url: 'http://www.example.com;'
                                                                                                                                                               <sid>002</sid>
                                                                                                                                                                                                                                                                 success function (data) {
                                                                                                                                                               <name>#==Y</name>
                                                                                                                                                                                                                                                                    consolelog(data);
                                                                                                                                                               </student>
                                                                                                                                                         </students>
                                                                                                                    XML DOM 即 XML 文档对象模型,是 w3c 组织定义的一套操作 XML 文档对象
```

---- XML DOM -

的API。浏览器会将 XML 文档解析成文档对象模型。