1. 服务器的基本概念与初始Ajax
2. URL地址
   1. URL地址的概念

统一资源定位符，表示互联网上每一个资源的唯一存放位置。

* 1. URL地址的组成部分

由三部分组成：①客户端与服务器之间的通信协议；②存在该资源的服务器；③资源在服务器上具体的存放位置。

1. 客户端与服务器的通信过程
   1. 客户端与服务器的通信过程

通信过程：请求→处理→响应；网页中的为一个资源都是通过请求→处理→响应的方式从服务器获取回来的。

1. 服务器对外提供的资源
   1. 服务器对外提供的资源

数据

* 1. 网页中如何请求数据

使用XMLHttpRequest对象，var xhrObj = new XMLHttpRequest()

* 1. 资源的请求方式

分为get和post请求：①get请求通常用于获取服务器端资源；②post请求通常用于想服务器提交数据。

1. Ajax
   1. 什么是Ajax

异步JavaScript和XML，在网页中利用XMLHttpRequest对象和服务器进行数据交互的方式。

* 1. Ajax的作用

用于实现网页与服务器之间的数据交互。

1. jQuery中的Ajax

jQuery中的Ajax对XMLHttpRequest进行了封装，降低了Ajax的使用难度。

* 1. $.get()

$.get(‘url’, [data], [callback])

参数含义：①url为要请求的资源地址；②data为请求资源期间要携带的参数；③callback为请求成功时的回调函数。

* 1. $.post()

$.post(‘url’, [data], [callback])

参数含义：①url为提交数据的地址；②data为要提交的数据；③callback为数据提交成功时的回调函数。

* 1. $.ajax()

$.ajax({

type: ‘’, //请求的方式，get或post

url: ’’, //请求的url地址

data: {} //本次请求携带的数据

success: function(res) {} //请求成功后的回调函数

})

1. 接口
   1. 接口的概念

被请求的URL地址，即数据接口。每个接口必须有请求方式。

* 1. 接口文档

接口的说明文档，是调用接口的依据。包含了对接口URL，参数以及输出内容的说明。

* 1. 接口文档的组成部分

接口文档包含：①接口名称；②接口URL；③调用方式；④参数格式；⑤响应格式；⑥返回示例(可选)

1. form表单与模板引擎
2. form表单的基本使用
   1. 表单的作用

表单在网页中主要负责数据采集功能。

* 1. 表单的组成部分

表单由三部分组成：①表单标签；②表单域（例：文本框、密码框、多行文本框、单选框、复选框等）；③表单按钮。

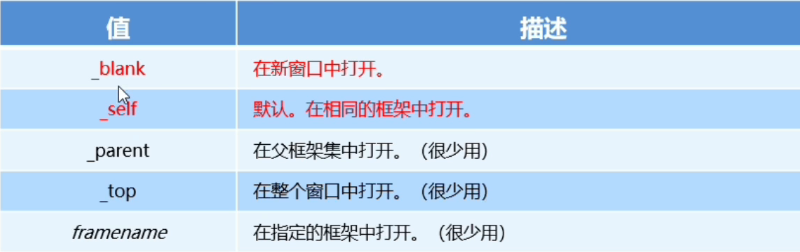
* 1. <form>标签的属性
  2. action

action属性用来规定当提交表单时，向何处发送表单数据。action属性的值应该是一个URL地址，该URL地址专门负责接收表单提交过来的数据。

提交表单后，页面会立即跳转到action属性指定的URL地址。

* 1. target

target属性用来规定在何处打开action URL。



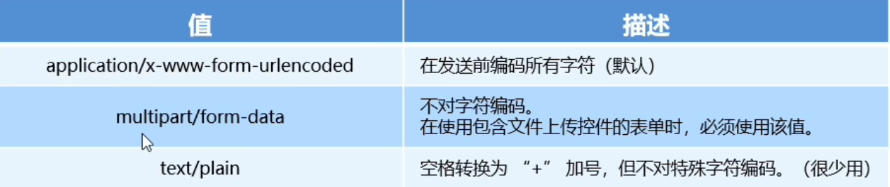
* 1. method

method属性用来规定以何种方式把表单数据提交到action URL。

方式包含：①get；②post。

* 1. enctype

enctype属性用来在发送表单数据之前如何对数据进行编码。



当涉及到文件上传的操作时，必须将enctype的值设置为multipart/form-data。

* 1. 表单的同步提交及缺点

1. 什么是表单的同步提交

通过点击submit按钮，触发表单提交的操作，从而使页面跳转到action URL的行为，叫表单的同步提交。

1. 表单同步提交的缺点
2. 表单同步提交后，页面会跳转到action URL所指的地址，用户体验差；
3. 表单同步提交后，页面之前的状态和数据会丢失。
4. 解决表单同步提交的缺点

表单只负责采集数据，Ajax负责将数据提交到服务器。

1. 通过Ajax提交表单数据
   1. 监听表单提交事件/阻止表单提交后页面跳转
   2. $(‘#form’).sumbit(function(e) {e.preventDefault();}) //阻止表单的提交和页面的跳转
   3. $(‘#form’).on(‘sumbit’, function(e) { e.preventDefault(); })
   4. 快速获取表单中的数据

$(‘selector’).serialize() 返回结果为字符串，username=值&password=值。

在使用serialize()函数快速获取表单数据时，必须为每个表单元素添加name属性。username和password即为表单设置的name属性值。

1. 模板引擎的基本概念
   1. 什么是模板引擎

模板引擎可以根据程序员指定的模板结构和数据，自动生成一个完整的HTML页面。

* 1. 模板引擎的好处

模板引擎的优点：①减少字符串的拼接操作；②使代码结构更加清晰；③使代码更易阅读与维护。

1. art-template模板引擎
   1. art-template模板引擎的基本使用

art-template的使用步骤：①导入art-template(引入template-web.js文件)；②定义数据；③定义模板(模板的HTML结构，必须定义在script标签中，type=”text/html”)；④调用template函数；⑤渲染HTML结构。

引入template-web.js文件，即模板引擎后，window全局增加template(‘模板ID’, ‘需要渲染的数据对象’)。

* 1. art-template标准语法

1. 什么是标准语法

{{…}}，在{{…}}内可以进行变量输出或循环数组等操作。

1. 标准语法-输出



1. 标准语法-原文输出

{{@value}}

value值中包含HTML标签结构，则需要使用原文输出语法在，保证value值中的HTML标签正常渲染。

1. 标准语法-条件输出

{{if value}} 满足条件输出的内容 {{/if}}

{{if v1}} 满足v1条件输出的内容 {{else if v2}} 满足v2条件输出的内容 {{/if}}

1. 标准语法-循环输出

{{each 被循环的对象}} {{$index}} {{$value}} {{/each}}

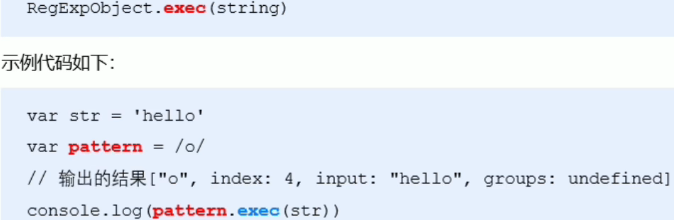
1. 标准语法-过滤器

{{value | filterName}} filterName为过滤器函数

定义过滤器函数：template.defaults.imports.filterName = function(value) {return 处理后的value;}

1. 模板引擎的实现原理
   1. 正则与字符串操作
2. 基本语法

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。



1. 分组

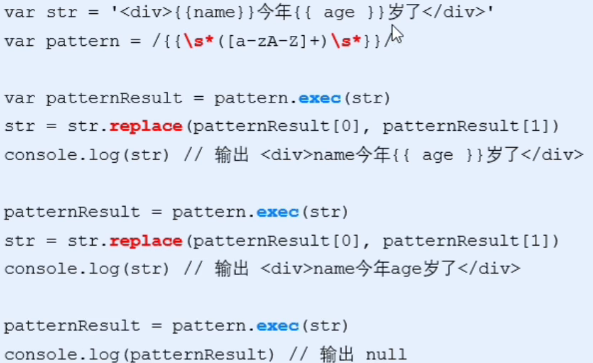
正则表达式中()抱起来的内容表示一个分组，可以通过分组来提取自己想要的内容。

1. 字符串的replace函数

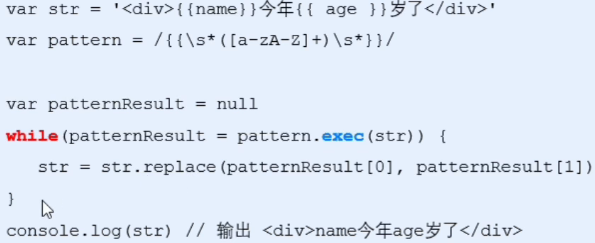
replace()函数用于在字符串中用一些字符替换另一些字符。

‘123456’.replace(‘123’, ‘abc’);

1. 多次replace

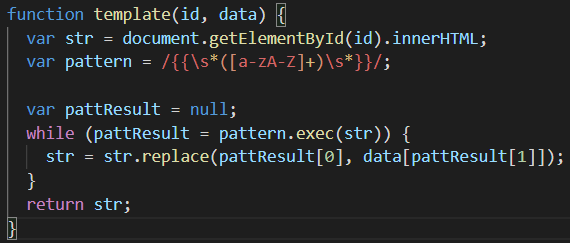


1. 使用while循环replace



* 1. 实现简易版模板引擎

实现步骤：①定义模板结构；②预调用模板引擎；③封装template函数；④导入并使用自定义的模板引擎。



三、 Ajax加强

1. XMLHttpRequest的基本使用

使用XMLHttpRequest获得的数据为JSON格式。

* 1. 什么是XMLHttpRequest

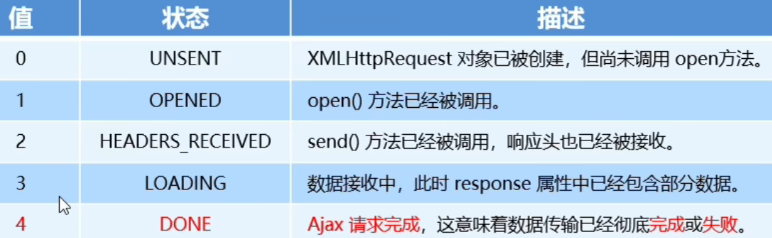
浏览器提供的Javascript对象，通过它可以请求服务器上的数据资源。

* 1. 使用xhr发起GET请求



* 1. xhr对象的readyState属性

readyState属性用来表示当前Ajax请求所处的状态。



* 1. 使用xhr发起带参数的GET请求

xhr.open('GET', 'http://www.liulongbin.top:3006/api/getbooks?id=1&name=西游记');

URL地址后面拼接的参数叫做查询字符串。

* 1. 查询字符串

定义：在URL的末尾加上用于下个服务器发送信息的字符串。

格式：URL?参数=值&参数=值

* 1. URL编码与解码

对URL中的中文字符进行编码，即使用英文字符表示非英文字符。

encodeURI()编码函数；decodeURI()解码函数。三个百分号表示一个中文字符。

* 1. 使用xhr发起POST请求



1. 数据交换格式
   1. 什么是数据交换格式

服务器端与客户端之间进行数据传输与交换的格式。

数据交换格式分类：①XML(不常用)；②JSON。

* 1. JSON
  2. 什么是JSON

JSON是Javascript对象和数组的字符串表示法，它使用文本表示一个JS对象或数组的信息，所以其本质是字符串。

* 1. JSON的两种结构与语法注意事项

对象结构：{“key”: value, “key”: value }

key必须是使用英文双引号包裹的字符串，value的诗句类型可以是数字，”字符串”，布尔值，null，数组，对象6种类型。

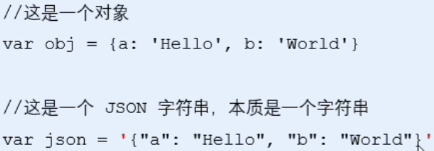
数组结构：[数字, “字符串”, 布尔值, null, 数组, 对象]

* 1. JSON语法的注意事项

key必须是使用英文双引号包裹的字符串，value的数据类型可以是数字，”字符串”，布尔值，null，数组，对象6种类型，JSON中无注释，不可使用单引号。

* 1. JSON和JS对象的关系

JSON是JS对象的字符串表示法。



* 1. JSON和JS对象之间的转换

JSON.parse(JSON字符串) //JSON转换为JS

JSON.stringify(JS对象) //JS转换为JSON，输出为字符串类型

* 1. 序列化和反序列化

把数据对象转换为字符串的过程叫做序列化；例如JSON.stringify(JS对象)

把字符串转换为数据对象的过程叫做反序列化；例如：JSON.parse(JSON字符串)

1. XMLHttpRequest Level2新特性
   1. 认识XMLHttpRequest Level2
   2. 旧版XMLHttpRequest的缺点
2. 只支持文本数据的传输，无法用来读取数据和上传文件；
3. 传送和接收数据时，没有进度信息，只能提示是否完成。
   1. XMLHttpRequest Level2的新功能
4. 可以设置 HTTP 请求的时限；
5. 可以使用 FormData 对象管理表单数据；
6. 可以上传文件；
7. 可以获得数据传输的进度信息。
   1. 设置HTTP请求超时

xhr.timeout = 1000;

xhr.ontimeout = function(event) { //请求超时后的回调函数

alert(“请求超时！”);

}

* 1. FormData对象管理表单数据



FormData对象可以用来获取网页表单的值, var data = $('#form1').serialize();此方法也可获取表单中的数据，是jQuery中的Ajax，在原生js中无法使用



1. jQuery高级用法
   1. jQuery实现文件上传
   2. ajaxStart(callback) Ajax请求开始时，执行此函数。可以在callback中显示loading效果。

$(document).ajaxStart()函数会监听当前文档内所有的Ajax请求。

* 1. ajaxStop(callback) Ajax请求结束时，执行此函数。可以在callback中隐藏loading效果。

1. axios
   1. 什么是axios

Axios是专注于网络数据请求的库。比原生的XMLHttpRequest对象简单易用，比jQuery轻量化，只专注于网络数据请求。

* 1. axios发起get请求

使用axios需导入axios.js包。

axios.get('url', { params: { /\*参数\*/ } }).then(callback)

* + 1. url为请求的url地址，可赋值给变量后引用；
    2. { params: { /\*参数\*/ } }为请求的参数对象，可将请求的参数对象赋值给变量后引用；
    3. callback为请求成功后的返回函数，返回的res对象包含其他数据，res.data是才是服务器返回的数据。
  1. axios发起post请求

axios.post('url', { /\*参数\*/ }).then(callback)

* 1. 直接使用axios发起请求

axios({

method: '请求类型',

url: '请求的URL地址',

data: { /\* POST数据 \*/ },或params: { /\* GET参数 \*/ }

}) .then(callback)

四、跨域与JSONP

* + - 1. 了解同源策略和跨域
  1. 同源策略
  2. 什么是同源

两个页面的协议、域名和端口都相同，则两个页面具有相同的源。端口号默认为80。

* 1. 什么是同源策略

同源策略是浏览器提供的一个安全功能。同源策略限制了从同一个源加载的文档或脚本如何与来自另一个源的资源进行交互。这是一个用于隔离潜在恶意文件的重要安全机制。

1. 无法读取非同源网页的 Cookie、LocalStorage 和 IndexedDB;
2. 无法接触非同源网页的 DOM;
3. 无法向非同源地址发送 Ajax 请求。
   1. 跨域
   2. 什么是跨域

两个URL之间的协议、域名和端口任意一项或多项不一致，则为跨域。

* 1. 浏览器对跨域请求的拦截

浏览器允许发起跨域请求，服务器也会响应，跨域请求回来的诗句会被浏览器拦截，无法被页面获取。

* 1. 如何实现跨域数据请求

方法：JSONP和CORS

JSONP：兼容性好，但不是官方提供的解决方法，只能支持GET请求；

CORS：官方提供的解决方法，支持GET和POST请求，但兼容性差。

* + - 1. JSONP
  1. 什么是JSONP

JSONP是JSON的一种使用模式，可以用于解决主流浏览器的跨域数据访问问题。

* 1. JSONP的实现原理

<script>标签不受浏览器同源策略的影响，可以通过src属性，请求非同源的js脚本。因此，JSONP的实现原理，就是通过<script>标签的src属性，请求跨域的数据接口，并通过函数调用的形式，接收跨域接口响应回来的数据。

* 1. 实现简单的JSONP

1. 在<script>标签中定义函数(例如：function abc(data) {…})
2. 再新建一个<script>标签，<script src="http://ajax.frontend.itheima.net:3006/api/jsonp?callback=abc&参数"></script>
   1. JSONP的缺点

JSONP之指出GET数据请求，不支持POST请求。

JSONP和Ajax之间没有任何关系，因为JSONP没有用到XMLHttpRequest对象。

* 1. jQuery中的JSONP

$.ajax({

url: 'http://ajax.frontend.itheima.net:3006/api/jsonp?name=zs&age=20',

dataType: 'jsonp', // 如果要使用 $.ajax() 发起 JSONP 请求，必须指定 datatype 为 jsonp

success: function(res) {

console.log(res)

}

})

使用jQuery发起JSONP请求，会自动携带一个callback=jQueryxxx的参数，jQueryxxx是随机生成的一个回调函数名称。

* 1. 自定义参数及回调函数名称

$.ajax({

url: 'http://ajax.frontend.itheima.net:3006/api/jsonp?name=zs&age=20',

dataType: 'jsonp',

jsonp: 'callback', // 发送到服务端的参数名称，默认值为 callback

jsonpCallback: 'abc', // 自定义的回调函数名称，默认值为 jQueryxxx 格式

success: function(res) {

console.log(res)

}

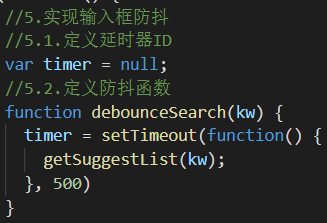
})

* 1. jQuery中JSONP的实现过程

jQuery中实现JSONP采用的是动态创建和移除<script>标签的方式。

1. 在发起JSONP请求的时候，动态向<header>中append一个<script>标签；
2. 在 JSONP 请求成功以后，动态从 <header> 中移除刚才 append 进去的 <script> 标签。
3. 防抖和节流
   1. 什么是防抖

防抖策略是当事件被出发后，延迟n秒后再执行回调函数，如果在n秒内事件被再次触发，则重新计时。



* 1. 防抖的应用

在输入框中输入完成后，再执行查询的请求，可有效减少请求次数，节约请求资源。

* 1. 什么是节流

节流策略可以减少一段时间内事件的触发频率。



* 1. 节流的应用

1. 鼠标连续不断的触发某事件，只在单位时间内触发一次；
2. 懒加载时要监听计算滚动条的位置，但不必每次滑动都触发，可以降低计算的频率，而不必去浪费 CPU 资源。
   1. 防抖和节流的区别
3. 如果事件被频繁触发，发抖能保证只有最后一次触发生效；
4. 如果事件被频繁触发，节流能够减少事件触发的频率。

五、HTTP协议加强

* + - 1. HTTP协议简介
  1. 什么是通信

通信是信息的传递和交换。

通信三要素：①通信的主体；②内容；③方式。

* 1. 什么是通信协议

通信协议是指通信的双方完成通信所必须遵守的规则和约定。

* 1. 什么是HTTP协议

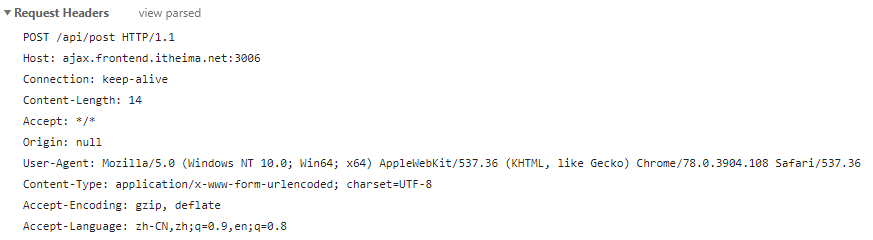
HTTP协议即超文本传送协议，它规定了客户端与服务器之间进行网页内容传输时所必须遵守的传输格式。

1. HTTP请求消息
   1. 什么是HTTP请求消息

由于 HTTP 协议属于客户端浏览器和服务器之间的通信协议。因此，客户端发起的请求叫做 HTTP 请求，客户端发送到服务器的消息，叫做 HTTP 请求消息/HTTP请求报文。

* 1. HTTP请求消息的组成部分

HTTP请求消息由请求行、请求头部、空行、请求体组成。



1. 请求行由3部分组成：请求方式、URL、协议版本；
2. 请求头部：用来描述客户端的基本信息，把客户端相关的信息告知服务器，请求头部由键:值对组成。
3. 空行：用来分隔请求头部和请求体；
4. 请求体：请求体中存放的是通过POST方式提交到服务器的数据。

注意：GET请求没有请求体；只有POST请求才有请求体。

1. HTTP相应消息
   1. 什么是HTTP响应消息

服务器响应给客户端的消息内容就是HTTP响应消息/HTTO响应报文。

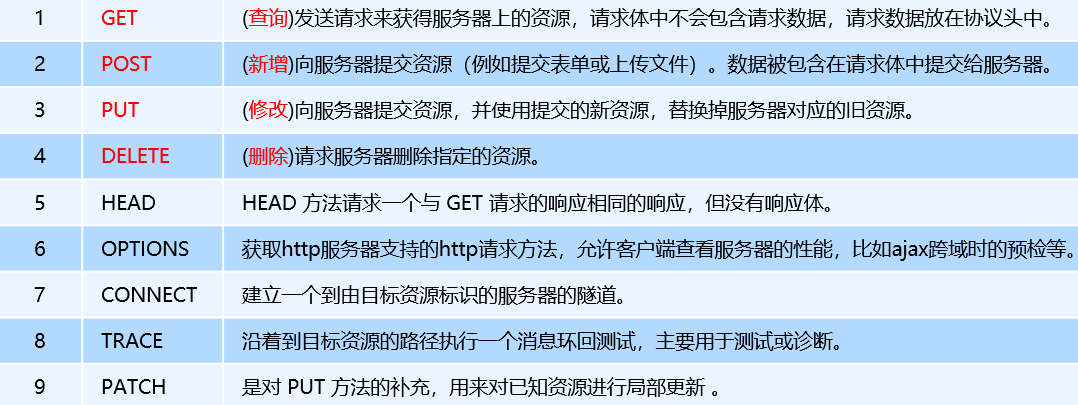
* 1. HTTP响应消息的组成部分

HTTP响应消息由状态行、响应头部、空行、响应体组成。

1. 状态行由3部分组成：HTTP协议版本、状态码、状态码的描述文本。
2. 响应头部：用来描述服务器的基本信息；
3. 空行；
4. 响应体：响应体中存放的是服务器相应给客户端的资源内容。
5. HTTP请求方法
   1. 什么是HTTP请求方法

对服务器上的资源执行的操作。

* 1. HTTP请求方法



1. HTTP响应状态码
   1. 什么是HTTP响应状态码

HTTP响应状态码用来表示响应的状态。

* 1. HTTP响应状态码的组成和分类

