day04

十.form表单

4.控件

⑥新表单元素 10个

|  |
| --- |
| 1.邮箱  <input type="email" name="um">  验证输入的内容有@，@后面有字符 |
| 2.搜索类型文本框  <input type="search">  自带快速清除功能 |
| 3.url类型  <input type="url" name="url">  验证输入的内容，以http://开头+任意字符 |
| 4.电话号码  <input type="tel" name="tel">  移动设备使用，在获取焦点时，打开虚拟键盘 |
| 5.数字类型  <input type="number" name="number"  max="20" min="10" step="3">  max 最大值  min 最小值  step 步长 |
| 6.范围类型  <input type="range" name="range"  max="50" min="20" step="5">  max 最大值  min 最小值  step 步长 |
| 7.颜色类型  <input type="color"> |
| 8.日期类型，提供一个日期选择控件  <input type="date"> |
| 9.月份类型  <input type="month"> |
| 10.周类型  <input type="week"> |

**AJAX**

一.HTTP协议

1.URL

|  |
| --- |
| [http://cdn.tmooc.cn/bsfile/imgad/A9.png](http://cdn.tmooc.cn/bsfile/imgad/2A763EDADF174F26960C6A8E0972049A.png)  URL的结构：协议+主机名称+目录结构+文件名称  url的完整结构：  <scheme>://<user>:<pwd>@<host>:<port>/<path>;  <params>?<query>#<frag> |

①<scheme>://

|  |
| --- |
| scheme：方案，协议。定义，以哪种方式获取服务器资源  C:\Users\web\Desktop\web1906\web第二阶段练习效果图示\协议及端口号.png |

②<user>:<pwd>

|  |
| --- |
| 后台管理员，偶尔使用，绝大多数程序员不再使用这种方式登录 |

③<host>

|  |
| --- |
| 主机名称 localhost/127.0.0.1/www.baidu.com  ip地址/域名  域名通过DNS服务器做域名解析，把IP地址传递给我们，  之后，我们的浏览器再去访问这个ip地址 |

④:<port>

|  |
| --- |
| 端口：就是在主机上，每一个服务对应的柜台号 |

⑤<path>

|  |
| --- |
| 路径，资源在主机上具体存放的位置  [http://127.0.0.1:80/cl/nodejs/day01/](http://127.0.0.1/cl/nodejs/day01/) |

⑥<params>

|  |
| --- |
| 参数，跟踪状态参数 |

⑦?<query>

|  |
| --- |
| http://127.0.0.1:8080/login?uname=dingding&upwd=12345  get方法提交请求时，地址栏出现的查询字符串 |

⑧#<frag>

|  |
| --- |
| 锚点 |

2.HTTP协议

|  |
| --- |
| HTTP:HyperText Transfer Protocol 超文本传输协议  专门用于传输html文件，规范了数据是如何打包和传递的  C:\Users\web\Desktop\web1906\web第二阶段练习效果图示\HTTP协议历史.png |

3.web请求响应的原理

|  |
| --- |
| C:\Users\web\Desktop\web1906\web第二阶段练习效果图示\请求响应流程.png |

4.消息/报文 Message

|  |
| --- |
| 请求消息 Request Message  响应消息 Response Message  C:\Users\web\Desktop\web1906\web第二阶段练习效果图示\message.png  1. 请求消息 Request Message  请求起始行 请求头 请求主体  2. 响应消息 Response Message  响应起始行 响应头 响应主体 |

5.请求消息 Request Message

|  |
| --- |
| 客户端发送给服务器的数据库，请求消息  由三部分组成：请求起始行，请求头，请求主体 |

①请求起始行

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.请求方法   |  |  | | --- | --- | | HTTP原生的请求方法 | Restful API 规则 | | get:客户端向服务器要数据  明文，有大小限制2KB  使用查询字符串传递明文 | get:查询 select  无请求主体 | | post：客户端要传递给服务器数据的时候，使用  隐式，没有大小限制  使用请求主体传递参数  formdata就是请求主体 | post:新建 insert  有请求主体 | | put:客户端往服务器放文件的时候使用，一般被禁用 | put:修改 update  有请求主体 | | delete:客户端从服务器删除文件的时候使用，一般被禁用 | delete：删除  无请求主体 | |
| 2.协议版本号 http/1.1 |
| 3.请求的url |

②请求头

|  |
| --- |
| Host: www.tmooc.cn 客户端告诉服务器，请求的是哪一个主机  Connection: keep-alive客户端告诉服务器，开启持久连接  User-Agent: 客户端告诉服务器，我现在的浏览器的类型和版本号  Accept-Encoding: gzip 客户端告诉服务器，我能够接受的压缩文件的格式  Accept-Language: zh-CN 客户端告诉服务器，我能够接受的自然语言的类型  Referer: http://www.tmooc.cn/ 客户端告诉服务器，本次请求来自于哪个网页 |

③请求主体

|  |
| --- |
| form data  post,put有请求主体  get,delete 没有请求主体 |

6.响应消息 Response Message

|  |
| --- |
| 服务器发送给客户端的数据块  由三部分组成：响应起始行，响应头，响应主体 |

①响应起始行

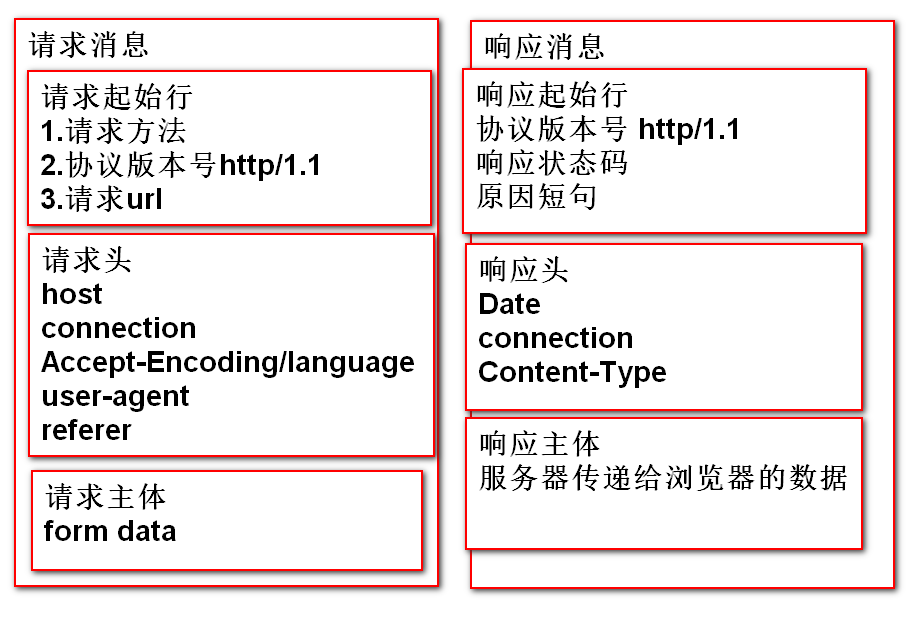
|  |  |
| --- | --- |
| 协议版本号 http/1.1  响应状态码   |  | | --- | | 1XX:正在请求，提示的信息  2XX:200 响应成功  3XX:重定向  301 永久重定向  302 临时重定向  304 命中缓存  4XX:404 请求资源不存在  403 权限不够  405 请求方法不被允许  5XX：500服务器代码出错 |   原因短句，就是响应状态码简短的解释说明 |

②响应头

|  |  |
| --- | --- |
| Date: Wed, 31 Jul 2019 07:17:08 GMT  服务器告诉浏览器，服务器响应的时间  Connection: keep-alive  服务器告诉浏览器，已经开启持久连接  Content-Type: text/html  服务器告诉浏览器，响应主体的内容是什么类型   |  | | --- | | text/html 响应回来的数据是html的文本  text/plain 响应回来的数据是普通文本  application/javascript 响应回来的是js文本  text/css 响应回来的是css文本  image/jpeg png gif 响应回来的是图片  application/xml 响应回来的是xml的字符串  application/json 响应回来的是json字符串 | |

③响应主体

|  |
| --- |
| 服务器传递给浏览器的数据 |



7.缓存

|  |
| --- |
| 客户端将服务器响应回来的数据进行自动的保存  当再次访问的时候，直接使用保存的数据 |

缓存的优点

|  |
| --- |
| 1.减少冗余的数据传输，节省客户端流量  2.节省服务器带宽  3.降低了对服务器资源的消耗，运行的要求  4.降低了由于远距离传输造成的加载延迟 |
| 缓存的新鲜度和过期  C:\Users\web\Desktop\web1906\web第二阶段练习效果图示\缓存流程.png  1.请求--无缓存--连接服务器--存缓存--客户端运行  2.请求--有缓存--够新鲜--使用缓存--客户端运行  3.请求--有缓存--不新鲜--连接服务器确认是否过期--没过期--更新缓存的新鲜度--客户端运行  4.请求--有缓存--不新鲜--连接服务器确认是否过期--已过期--连服务器获取数据--存缓存--客户端运行 |

①与缓存相关的消息头

|  |
| --- |
| Cache-Control: max-age=3600  从服务器将文档传到客户端那一刻起  此文档处于新鲜的秒数 |

②设置缓存需要修改的消息头

|  |
| --- |
| <meta http-equiv="Cache-Control"  content="max-age=3600"> |

8.HTTP性能优化

①HTTP的连接过程

|  |
| --- |
| 发起请求-->建立连接-->服务器处理请求-->访问资源-->构建响应-->发送响应--->记录日志 |

②HTTP的性能优化

|  |
| --- |
| 1.减少连接的创建次数(开启持久连接)  2.减少请求的次数  3.提高服务器运行速度  4.尽可能的减少响应数据的长度(代码压缩) 16:06~16:21休息 |

9.HTTPS，安全的http协议

|  |
| --- |
| ssl:为数据通信提供安全支持  1.客户端发送请求--->ssl加密--->服务器接收到加密文件--->ssl层解密，得到请求明文  2.服务器发送响应--->ssl加密--->客户端接收到加密文件--->ssl层解密，得到响应明文 |

二.DOM操作(简单的dom操作)

|  |
| --- |
| ajax提交请求，不需要form表单  但是，form表单自带收集整理数据的功能  ajax不使用form表单，就没有自动收集整数数据的功能  我们需要使用js的dom操作，手写代码，收集数据 |

1.完整JavaScript的构成

|  |
| --- |
| 1.js核心 ECMA Script ES6  2.DOM ：Document Object Model 文档对象模型  让js动态的操作页码上的元素  3.BOM：Browser Object Model 浏览器对象模型  让js动态的操作浏览器 |

2.使用js获取页码上某个元素

①获取元素对象

|  |
| --- |
| <input id="d1" type="text">  document.getElementById("d1") |

②获取/修改input的值

|  |
| --- |
| var u=对象.value; 获取value  对象.value="...." 修改value |

③获取双标签的内容区域

|  |
| --- |
| var u=对象.innerHTML; 获取内容区域  对象.innerHTML="...."; 修改内容区域  d1中填入的文本，在点击按钮之后，显示在div里 |

作业：

1.完成04\_ex.html

2.完成下图

