

前置知识

1 软件的架构

C/S架构

C/S即Client/Server(客户端/服务器)结构,它可以分为客户端和服务端两层:第一层是在客户机系统上结合了界面显示与业务逻辑,第二层是通过网络结合了数据库服务器。

C/S架构的优点

1. C/S架构的界面和操作可以很丰富。(客户端操作界面可以随意排列,满足客户的需要)
2. 安全性能可以很容易保证。(它可以对权限进行多层次校验,提供了更安全的存取模式,对信息安全的控制能力很强。一般高度机密的信息系统采用C/S结构适宜)
3. 由于只有一层交互,因此响应速度较快。(直接相连,中间没有什么阻隔或岔路,比如QQ,每天那么多人在线,也不觉得慢)

C/S架构的缺点

可以将QQ作为类比:

1. 一般建立在专用的网络上,小范围里的网络环境,局域网之间再通过专门服务器提供连接和数据交换服务。
2. 用户群固定。由于程序需要安装才可使用,因此不适合面向一些不可知的用户。
3. 维护成本高,发生一次升级,则所有客户端的程序都需要改变。

例如qq, 微信, 王者荣耀...

B/S架构

B/S即Browser/Server(浏览器/服务器)结构,通过访问一个网页的形式来使用,将一些运算等操作放到远端的服务器上。

B/S架构的优点

1. 客户端无需安装,有Web浏览器即可。
2. BS架构可以直接放在广域网上,通过一定的权限控制实现多客户访问的目的,交互性较强。
3. BS架构无需升级多个客户端,升级服务器即可。可以随时更新版本,而无需用户重新下载。

B/S架构的缺点

1. 在跨浏览器上,BS架构不尽如人意。
2. 表现要达到CS程序的程度需要花费不少精力。
3. 在速度和安全性上需要花费巨大的设计成本,这是BS架构的最大问题。
4. 客户端服务器端的交互是请求-响应模式,通常需要刷新页面,这并不是客户乐意看到的。(在Ajax

风行后此问题得到了一定程度的缓解)

例如, 京东, 淘宝。。。

2 HTTP协议

HTTP是超文本传输协议（HyperText Transport Protocol）的缩写，它用于传送WWW方式的数据。HTTP协议采用了请求/响应模型。客户端向服务器发送一个请求报文，请求头包含请求的方法、URL、协议版本、以及包含请求修饰符、客户信息和内容的类似于MIME的消息结构。服务器以一个状态行作为响应，响应的内容包括消息协议的版本，成功或者错误编码加上包含服务器信息、实体元信息以及可能的实体内容。

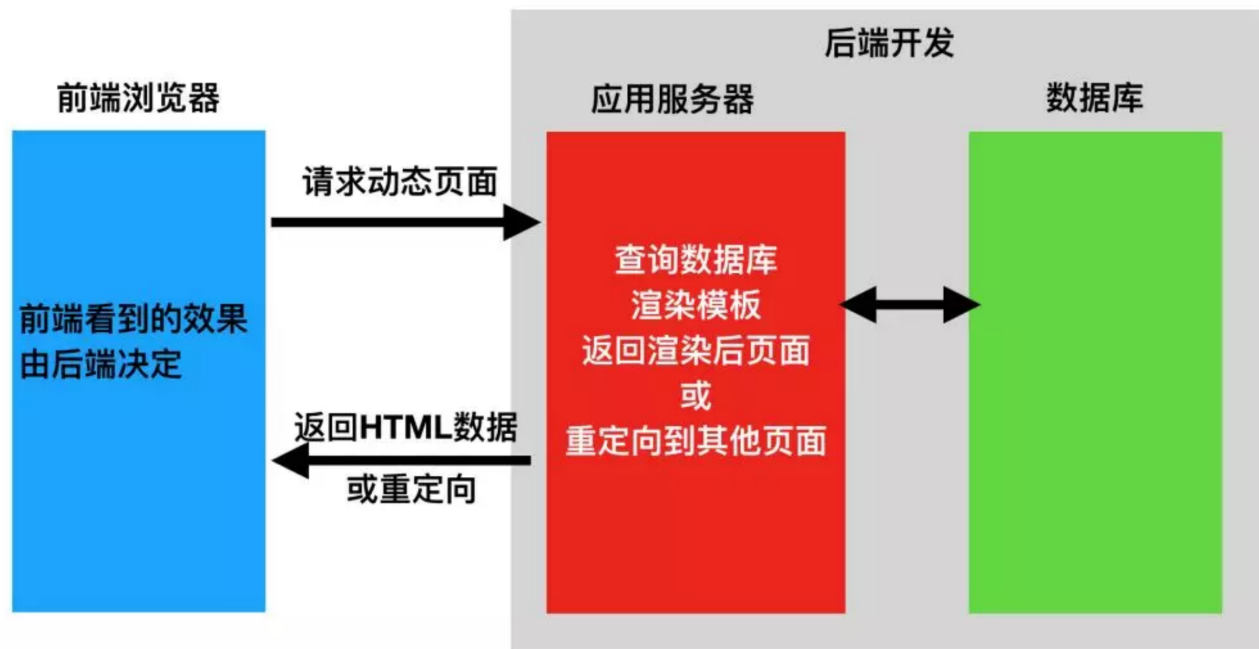
3 URL

URL（Uniform Resource Locator）统一资源定位符，就像每家每户都有一个门牌地址一样，每个网页也都有一个Internet地址。当你在浏览器的地址栏中输入一个URL或者单击一个超链接时，URL就确定了要浏览的地址。浏览器通过超文本传输协议(HTTP)，将Web服务器上站点的网页代码提取出来，并翻译成漂亮的网页。URL的格式如下：协议://ip:port/path 例如：<http://192.168.1.101:80/test/index.html>

4 前后端分离开发

以Java Web项目为例，在传统的开发模式中，前端代码（Html、css、js）写在JSP中，甚至JSP中嵌入Java代码。当用户访问网站时，页面数据也就是Html文档，由Servlet容器将jsp编译成Servlet，然后将jsp中的html，css，js代码输出到浏览器，这个过程需要经过很多步骤，才能响应用户的请求。这个过程非常繁琐，效率低下，直接造成了页面响应速度慢的效果。从项目维护的角度上，传统的开发模式，前端代码和后端代码耦合在一起，导致代码混乱不堪，极大的降低了项目的可维护性，增加了维护成本。从开发角度来看，研发人员在开发过程中，不仅要设计后端架构还要兼顾前端展示，导致开发效率低下，延长开发周期。扬长避短，为什么不让专业的人去做专业的事？

传统的开发模式下的系统数据交互图：



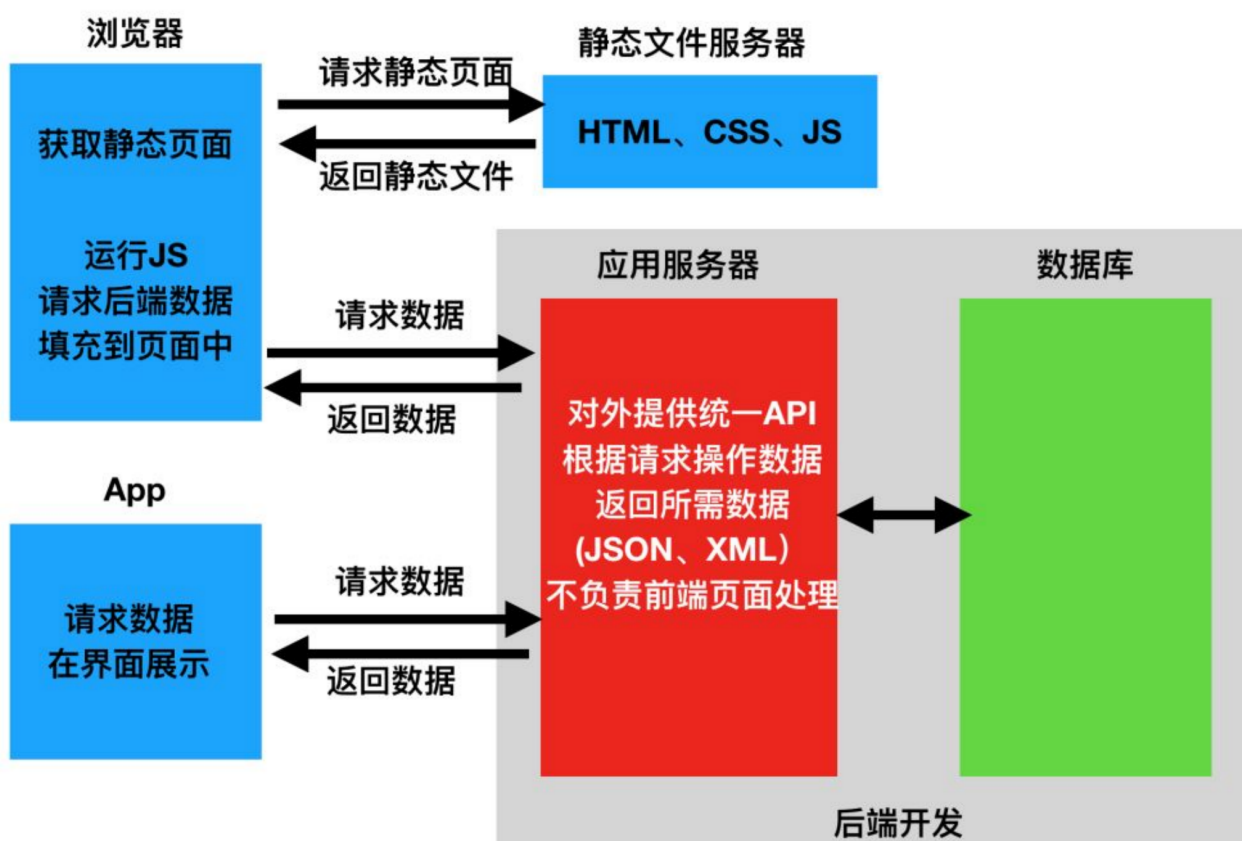
传统开发模式的劣势和不足

- 1. 开发出的软件响应速度慢，质量差，用户体验差。
- 2. 前后端严重耦合，代码混乱，可维护性差。
- 3. 研发人员前后端兼顾，开发效率低下，研发周期变长。

为了解决传统开发模式中的这些病痛，前后端分离框架应用而生。

什么是前后端分离？

在前后端分离的开发模式中，后端仅返回前端所需的数据，前端负责渲染HTML页面，后端不再控制前端的效果，用户看到什么样的效果，从后端请求的数据如何加载到前端中，都由前端自己决定，后端仅仅需要提供一套逻辑对外提供数据即可，并且前端与后端的耦合度相对较低，在这种模式中，我们通常将后端开发的每个视图都成为一个接口，或者API，前端通过访问接口来对数据进行增删改查。总结一句话，后台负责提供数据，前端负责数据展示，职责分离，分工明确。对应的数据交互如下图：



总结

经典的JSP+Servlet+JavaBean的MVC时代，到SSM（Spring + SpringMVC + Mybatis）和SSH（Spring + Struts2 + Hibernate）的Java 框架时代，再到前端框架（VueJS、AngularJS、ReactJS）为主的MV*时代，然后是NodeJS引领的全栈时代，技术和架构一直都在进步。虽然“基于NodeJS的全栈式开发”模式很让人兴奋，但是把基于Node的全栈开发变成一个稳定，让大家都能接受的东西还有很多路要走。创新之路不会止步，无论是前后端分离模式还是其他模式，都是为了更方便得解决需求，但它们都只是一个“中转站”。前端项目与后端项目是两个项目，放在两个不同的服务器，需要独立部署，两个不同的工程，两个不同的代码库，不同的开发人员。前端只需要关注页面的样式与动态数据的解析及渲染，而后端专注于具体业务逻辑。

5 什么是大前端？

简单来说，大前端就是所有前端的统称，比如Web、Android、iOS、Watch等，最接近用户的那一层也就是UI层，然后将其统一起来，就是大前端。

由于node的出现，前端工程师不需要依赖于后端程序而直接运行，从而前后端分离起来。所以当开发一个新产品的时候服务只需要写一次，但是面向用户的产品可能有很多，例如网站、Android客户端、iOS客户端和微信小程序等。由于各个平台使用的技术栈都不一样，代码无法复用，非常浪费人力、物力。那么有没有什么技术能够解决这一痛点呢？大前端应运而生，其实大前端的主要核心就是跨平台技术，有了跨平台技术，各个平台的差异性就抹平了，开发者只需要一套技术栈就可以开发出适用于多个平台的客户端。

大前端的学习流程：HTML5/CSS3->云服务器->JavaScript(核心)->jQuery->HTML5 API->ES6(核心)->Vue（React）->原生App/混合式开发App/小程序->NodeJs->MySQL

前端开发环境的搭建

安装nodejs，Vue等 安装编译工具VSCode，webstrom等

web前端基础

HTML CSS JavaScript(ES5,ES6)

web前端框架

jQuery Vue React Angular 第三方组件库(elementUI, iView)

原生移动端(可选)

iOS Android

混合移动端开发

小程序，微信小程序，支付宝小程序等 ReactNative Flutter、uni-app等

Node后端开发

CommonJS RequireJS Node.js

6 软件开发流程

第一阶段：初始需求阶段

将自己的Idea或客户的需求进行合理的需求分析与市场调研，分析项目可行性。

第二阶段：原型设计阶段

将需求分析转化成未来系统符合用户期望的原型设计，原型设计完成后，审核并确认具体设计，准备进入开发编码阶段。

第三阶段：个性化开发阶段

软件开发设计：对整个软件系统进行设计，如UI界面设计、系统框架设计、数据库设计等，为系统开发一个健壮的结构并调整设计使其与实现环境相匹配。

程序开发编码：在开发构建阶段，由程序员根据详细设计及计划，将所有应用程序功能开发并集成为产品。

软件测试阶段：测试要验证对象间的交互作用，验证软件中所有组件的正确集成，检验所有的需求已被正确的实现，识别并确认缺陷在软件部署之前被提出并处理。

第四阶段：交付维护阶段

产品软件部署：部署的目的是成功的生成版本并将软件分发给最终用户。

正式验收交付：要确定软件、环境、用户是否可以开始系统的运作，交付阶段的重点是确保软件对最终用户是可用的。

后期项目维护：软件产品发布后，根据需求变化或硬件环境的变化对应用程序进行修改。

7 企业级环境搭建

nodejs

下载地址：<https://nodejs.org/en/>、或者中文网站：<https://nodejs.org/zh-cn/download> 根据安装包提示下一步安装即可，安装成功后在终端中输入 `node -v` 查看版本号

cnpm

node服务国内淘宝镜像源: `npm install -g cnpm --registry=https://registry.npmmirror.com`

VSCode

下载地址：<https://code.visualstudio.com/>

```
1  根据安装包提示下一步安装即可
2  安装所需插件：
3      Chinese (Simplified) Language Pack for Visual Studio Code - 汉化插件
4      Live Server 服务器预览插件
5      open in browser - 允许您在默认浏览器或应用程序中打开当前文件
6      file-icons VSCode文件的图标
7
8      HTML CSS Support - html, css快捷操作
9      HTML Preview - 提供预览HTML文档的能力
10     HTML Snippets - 完整的HTML标签，包括HTML5片段
11     Vetur - VS Code的Vue工具
12     View In Browser - 在系统的默认浏览器中查看html文件
13     Vue 3 Snippets - 一个Vue.js 3和Vue.js 2的代码段扩展
14     Vue VSCode Snippets - 代码片段，它将增强您的Vue工作流
15     .....可自行在网上搜索其他好用的vscode插件
```

HTML

超文本标记语言（英语：HyperText Markup Language，简称：HTML）是一种用于创建网页的标准标记语言。

学习网站：<https://www.w3school.com.cn/html/index.asp>

HTML5和HTML4的关系：

1. HTML5是应用超文本标记语言（HTML）的第五次修改，代表html语言第5版本。
2. HTML4是应用超文本标记语言（HTML）的第四次修改，代表html语言第4版本。
3. HTML4是为了适应 `PC`（Personal Computer）时代产生的，HTML5是为了适应移动互联网时代产生的。

4. HTML5和HTML4都是w3c（World Wide Web）推荐的标准语言。
5. HTML5在HTML4的基础上新增了一些特性。比如，让标签更具有语义化，新增了一些表单控件，新增了Web API等。

文档头信息

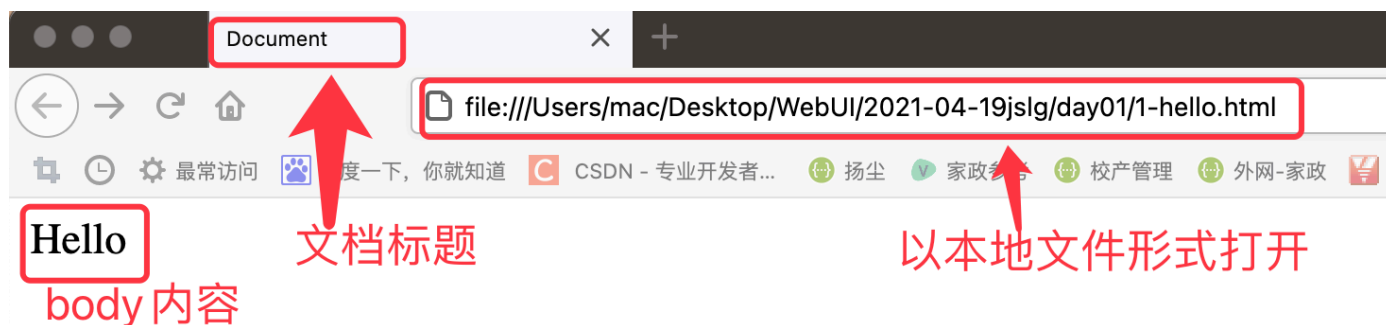
使用 **VSCode** 工具可以快速生成文档头信息，在 **VSCode** 中新建 **1-hello.html**，在该文件内容中输入英文`!`，回车即可快速生成完整的文档头信息。以 `.html`或者`.htm` 结尾的文件中使用 `!`或者`html:5` 来快速产生文档头信息。

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Document</title>
9  </head>
10
11 <body>
12
13 </body>
14
15 </html>
```

说明如下

```
1  <!-- <!DOCTYPE> 不是 HTML 标签。它为浏览器提供一项信息（声明），即 HTML 是用HTML5版本编写的。 -->
2  <!DOCTYPE html>
3  <!-- 语言是English -->
4  <html lang="en">
5  <!-- 文档的头部，不在网页的视口区显示 -->
6  <head>
7      <!-- 元信息，字符编码UTF-8 -->
8      <meta charset="UTF-8">
9      <!-- 元信息，设置浏览器优先使用什么模式来渲染页面，以下代码告诉IE浏览器，IE8/9及以后的版本都会以最高版本IE来渲染页面，是针对IE浏览器的设置 -->
10     <meta http-equiv="X-UA-Compatible" content="IE=edge">
11     <!-- 元信息，设置可使用的宽度为设备的宽度，初始缩放比例为1 -->
12     <meta name="viewport" content="width=device-width, initial-scale=1.0">
13     <!-- 文档的标题，在选项卡上显示 -->
14     <title>Document</title>
15 </head>
16 <!-- 文档的体部，在网页的视口区显示 -->
17 <body>
18     <!-- 该内容会显示在网页中 -->
19     Hello
20 </body>
21 </html>
```

以上代码运行到浏览器中效果如下图



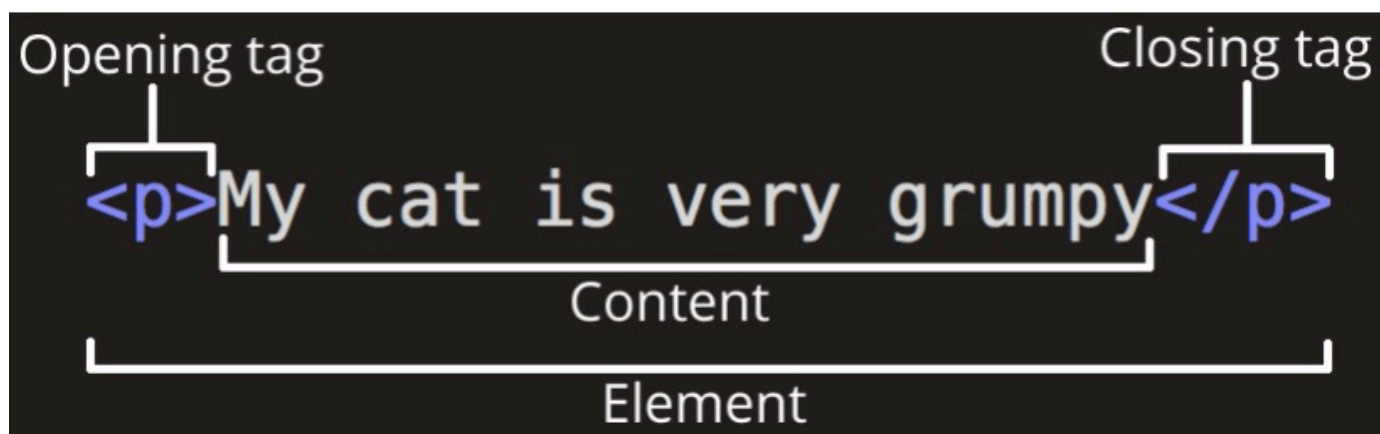
另外，HTML 4.01的DOCTYPE如下，了解即可，之后使用HTML5的DOCTYPE声明，因为其简便很多。

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
```

注释

```
1 <!-- 注释的内容 -->
```

标签结构



常用的标签

div 和 span

```
1 <body>
2   <div>我是div标签内容</div>
3   <div>我是div标签内容</div>
4   <span>我是span标签内容</span>
5   <span>我是span标签内容</span>
6 </body>
```

页面展示效果如下

我是div标签内容
我是div标签内容

两个div

我是span标签内容 我是span标签内容

两个span

div是块级元素，独占一行

span是行内元素，与其他行内元素共享一行

表格标签

```
1  <body>
2  <table>
3    <caption>信息表</caption>
4    <thead>
5      <tr>
6        <th>列的标题</th>
7        <th>列的标题</th>
8        <th>列的标题</th>
9        <th>列的标题</th>
10     </tr>
11   </thead>
12   <tbody>
13     <tr>
14       <td>11</td>
15       <td>12</td>
16       <td>13</td>
17       <td>14</td>
18     </tr>
19     <tr>
20       <td>21</td>
21       <td>22</td>
22       <td>23</td>
23       <td>24</td>
24     </tr>
25     <tr>
26       <td>31</td>
27       <td>32</td>
28       <td>33</td>
29       <td>34</td>
30     </tr>
31   </tbody>
32 </table>
33 </body>
```

效果

信息表

列的标题 列的标题 列的标题 列的标题

11	12	13	14
21	22	23	24
31	32	33	34

如果想要表格有边框和宽度，则需要一些css设置，代码如下

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Document</title>
9      <style>
10         /* 设置表格标签样式 */
11         table {
12             /* 表格宽度 */
13             width: 400px;
14             /* 表格内文字水平对齐方式 */
15             text-align: center;
16         }
17
18         /* 设置table标签、td标签、th标签的样式 */
19         table,
20         table td,
21         table th {
22             /* 设置边框为1px的宽度、实线、颜色为黑灰色 */
23             border: 1px solid #444;
24             /* 表格内边框合并 */
25             border-collapse: collapse;
26         }
27     </style>
28 </head>
29
30 <body>
31     <table>
32         <caption>信息表</caption>
33         <thead>
```

```
34     <tr>
35         <th>列的标题</th>
36         <th>列的标题</th>
37         <th>列的标题</th>
38         <th>列的标题</th>
39     </tr>
40 </thead>
41 <tbody>
42     <tr>
43         <td>11</td>
44         <td>12</td>
45         <td>13</td>
46         <td>14</td>
47     </tr>
48     <tr>
49         <td>21</td>
50         <td>22</td>
51         <td>23</td>
52         <td>24</td>
53     </tr>
54     <tr>
55         <td>31</td>
56         <td>32</td>
57         <td>33</td>
58         <td>34</td>
59     </tr>
60 </tbody>
61 </table>
62 </body>
63
64 </html>
```

效果如下

信息表

列的标题	列的标题	列的标题	列的标题
11	12	13	14
21	22	23	24
31	32	33	34

表单form

表单可以用来采集用户信息和用户需求的，用户可以在表单控件里输入或选择内容。

```
1  <body>
2  <form action=" ">
3  <div>
```

```
4      用户名: <input type="text" name="username">
5  </div>
6  <div>
7      密码: <input type="password" name="password">
8  </div>
9  <div>
10     性别: <label for="male"><input id="male" type="radio" name="gender" value="male">
男</label>
11     <label for="female"><input id="female" type="radio" name="gender" value="female">
女</label>
12 </div>
13 <div>
14     爱好:
15     <label for="basketball">
16         <input id="basketball" type="checkbox" name="hobby" value="basketball">
17         篮球
18     </label>
19     <label for="swimming">
20         <input id="swimming" type="checkbox" name="hobby" value="swimming">
21         游泳
22     </label>
23     <label for="dancing">
24         <input id="dancing" type="checkbox" name="hobby" value="dancing">
25         跳舞
26     </label>
27 </div>
28 <div>
29     出生日期: <input type="date" name="birth">
30 </div>
31 <div>
32     城市:
33     <select name="city">
34         <option value="">请选择</option>
35         <option value="beijing">北京</option>
36         <option value="shanghai">上海</option>
37         <option value="guangzhou">广州</option>
38     </select>
39 </div>
40 <div>
41     个人介绍:
42     <textarea name="info" cols="30" rows="3"></textarea>
43 </div>
44 <div>
45     <button>提交按钮</button>
46 </div>
47 </form>
48 </body>
```

效果如图

用户名:

密码:

性别: ☐ 男 ☐ 女

爱好: ☐ 篮球 ☐ 游泳 ☐ 跳舞

出生日期:

城市:

个人介绍:

提交按钮

图像 **img**

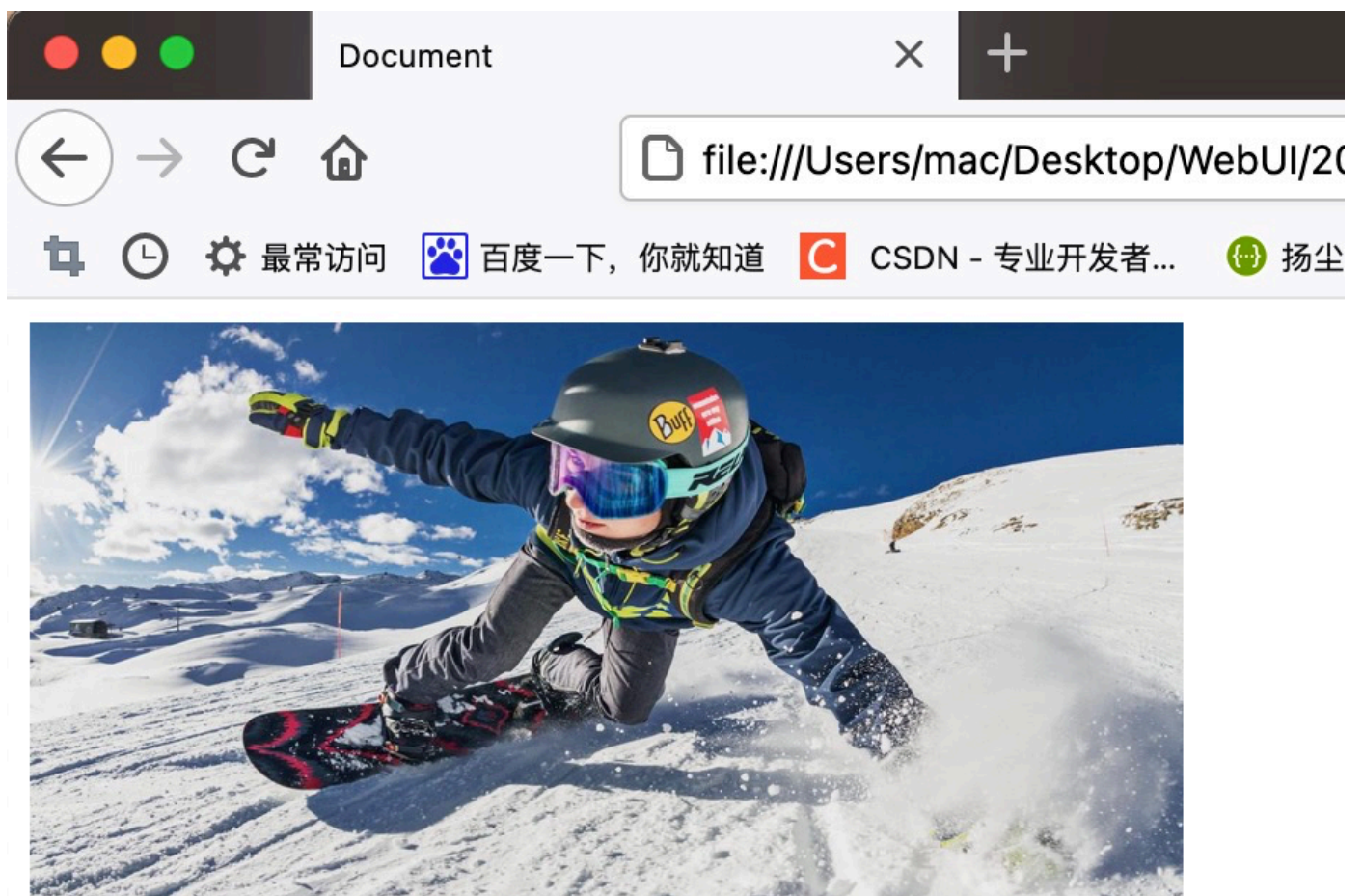
```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta http-equiv="X-UA-Compatible" content="IE=edge">
7    <meta name="viewport" content="width=device-width, initial-scale=1.0">
8    <title>Document</title>
9    <style>
10     /* 设置img标签样式 */
11     img {
12       /* 设置宽 */
13       width: 400px;
14       /* 设置高 */
15       height: 200px;
16     }
17   </style>
18 </head>
19
20 <body>
```

```

21     
22 </body>
23
24 </html>

```

在页面中效果如图



超链接a标签

a 标签的 `target` 属性代表目标链接打开的方式, `_blank` 代表在新页面打开

```

1 <a href="http://www.baidu.com" target="_blank">这是一个链接</a>

```

音视频

音频播放

```

1 <!-- autoplay自动播放 controls控制条 loop循环播放 -->
2 <audio autoplay controls loop src="./images/稻香.mp3"></audio>

```

播放视频

```
1 <!-- 视频 -->
2 <video autoplay controls loop src="./images/1.mp4"></video>
```

iframe标签

```
1 <iframe src="https://echarts.apache.org/zh/index.html" frameborder="0"></iframe>
```

可以使用iframe标签在当前网页内嵌入其他页面

参考 <https://www.tianqi.com/plugin/>，展示天气

```
1 <iframe width="300" height="30" frameborder="0" scrolling="no" hspace="0"
  src="https://i.tianqi.com/?c=code&a=getcode&id=34&py=chongqing&icon=1"></iframe>
2 <iframe width="280" height="300" frameborder="0" scrolling="no" hspace="0"
  src="https://i.tianqi.com/?c=code&a=getcode&id=55&icon=1"></iframe>
```

网页选项卡图标

```
1 <link rel="icon" sizes="any" mask href="https://www.baidu.com/favicon.ico">
```