

一、WEB概述

2018年10月29日 16:59

1. B/S: Browser-Server

- 浏览器服务器模型 WEBQQ 网页游戏

优点: 不需要下载客户端程序, 使用浏览器可以直接访问. 程序的升级操作是在服务器端进行的.

浏览器只需要刷新页面就可以看到升级后的效果

缺点: 浏览器具有一定的局限性, 页面的展示能力仍然是很差. 所有的页面数据都需要从服务器实时的获取, 所以对网速的依赖很高

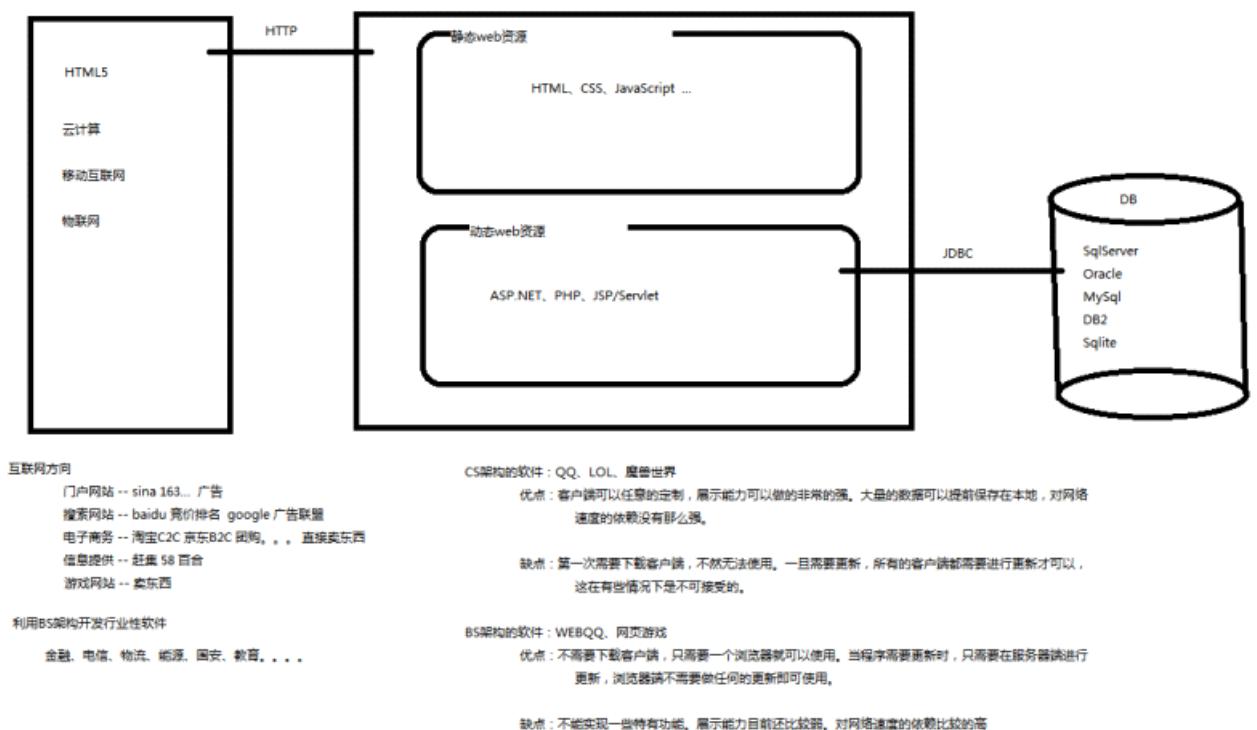
2. C/S: Client-Server

- 客户端服务器模型 QQ LOL

优点: 客户端可以任意的设计, 页面的展示能力就可以很强. 由于大量的资源都已经保存在了客户端, 和服务器交互的仅仅是一些变化的数据, 所以对网速的依赖很低

缺点: 第一次使用时需要下载客户端程序, 一旦程序需要升级操作, 所有的客户端程序都需要升级. 在有些场景中是不能被接受的.

3. web概述, 如图所示:



二、HTML

2018年10月29日 17:00

1. HTML是什么

- HTML(Hyper Text Mark-up Language)超文本标记语言 最基础的网页语言 W3C
- HTML不是一门编程语言 而是一门标记语言
- HTML是用标记(标签/元素)来描述网页内容的
- HTML是文档的一种

2. HTML的结构

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
  <HEAD></HEAD>
```

```
  <BODY></BODY>
```

```
</HTML>
```

<!DOCTYPE HTML>用来指定当前页面所遵循的html的版本

- 头部分用来存放html页面的基本属性信息,优先被加载
- 体部分用来存放页面数据,是可见的页面内容

<title></title>指定网页的标题

<meta http-equiv="Content-type" content="text/html; charset=UTF-8" />其中的charset的值用来指定浏览器用什么编码解析当前页面

3. HTML语法

- html标签分为开始标签和结束标签,如果标签内没有修饰的内容,开始标签和结束标签可以合并为一个自闭标签

如:
 换行

<hr/> 在当前行画一条线

- 标签通常都可以具有属性,属性与属性值用"="连接,属性的值可以用双引号、单引号引起来或者不用引号,一般会用双引号引起来。

如果不生效查看页面代码使用的是否为英文双引号

- html中对页面中代码需要做注释: <!-- html的注释 -->
- html中多个连续的空白字符(制表符,空格,换行)默认会合并为一个空格来显示。

如果非要输入空格,可以用转义字符来替代

如果非要输入换行,可以用
 来代替

***转义字符

< <

> >

" "

' '

空格

4. font标签

用来指定文本的字体/大小/颜色

color:指定字体颜色

值可以指定为颜色名 如:red

或 十六进制的颜色值 如:#000000

或 rgb三原色值 如:rgb(255,255,255) ie适用

例: 字体标签示例

5. 标题标签

指定特定样式字体的一组标签

<h1>一级标签</h1>

<h2>二级标签</h2>

<h3>三级标签</h3>

<h4>四级标签</h4>

<h5>五级标签</h5>

<h6>六级标签</h6>

属性:

align:指定文本的排列

| | |
|---------|-----|
| left | 靠左 |
| center | 居中 |
| right | 靠右 |
| justify | 自适应 |

6. 列表标签

○ 无序列表

 定义一个无序列表

 定义列表中的项

属性:

type:定义项目符号的类型

disc square circle

○ 有序列表

7. img标签-- 图像标签

○ 必选属性:

src:图片的路径

alt:代替图像显示的文本

可选属性:

width:宽度 px %

height:高度 px %

border:边框的宽度 px

8. 超链接-- <a> 锚

○ 用于指向当前位置以外的资源

i. 用于创建指向另外一个文档的超链接

ii. 用于在当前页面的不同位置之间进行跳转, 利用id或name属性进行跳转。

一般在本页面中使用, 当网页内容过长, 定位标记会比拖动滚动条方便快捷。

注: 定位标记要和超链接结合使用才有效。

例:

`标记位置`

`<p>.....<!--很多空行以制造网页过长的效果-->`

`返回标记位置`

注: 使用定位标记时一定要在href值的开始加入#标记名。

○ 重要属性:

href: 所指向资源的URL

name: 指定锚的名字

target: 指定浏览器打开目标URL的方式:

| | |
|--------|---------------|
| _blank | 在新窗口中打开目标url |
| _self | 在当前窗口中打开目标url |

9. 表格标签 -- <table>

<table> 定义一个HTML表格

<tr> 定义表格中的行

<td> 定义表格中的单元格

<th> 定义表格中的表头

○ table的重要属性:

border 边框宽度

cellspacing 单元格之间的空白距离

cellpadding 边框与单元格内容之间的距离

bgcolor 背景颜色

bordercolor 边框颜色

width 宽度

align 对齐方式

○ tr重要属性:

align 对齐方式

bgcolor 背景颜色

○ th/td重要属性:

align 对齐方式

bgcolor 背景颜色

width 宽度

height 高度

colspan 可横跨的列数

rowspan 可竖跨的行数

<caption> 定义表格的标题

1. 表单

(1) 浏览器向服务器发送数据的方式, 有两种:

a. 利用超链接向服务器发送数据 -- 请求参数

在超链接的后面拼接上要发送的请求参数, 链接和请求参数之间用?分割, 参数名和参数值用=连接, 多个参数之间用&分割, 可以存在多个同名的参数

b. 利用表单向服务器发送数据

利用HTML中的<form>标签以及一些表单项标签, 用户可以输入数据, 通过提交表单发送数据给服务器

(2) form标签

o 必须存在的属性:

action: 指定表单发送的目标URL地址

o 可选的属性:

method: 指定以何种方式发送表单

o http协议指定了7种提交方式, 其中5种使用的极少, 多数只用GET提交和POST提交。

o 只有使用表单并且明确的指定提交方式为post时(也就是设置method="POST")才是POST提交, 其他提交都是GET提交。

o GET提交和POST提交的区别:

主要区别体现在数据传输方式的不相同。

1) GET提交: 请求参数会赋在地址栏后进行传输

这种方式发送的数据量有限, 最大不超过1kb(或4kb)

数据显示在地址栏, 安全性差

2) POST提交: 请求参数在底层流中传输

这种方式发送的数据量无限制

地址栏上看不到数据, 比较安全

2. 表单中的项

表单中可以有多个输入项, 输入项必须有name属性才可以被提交, 如果输入项没有name属性, 则表单在提交时会忽略它

(1) <input> 输入框

o 重要属性:

▪ type属性

| | |
|-----------------|---|
| 文本框 text | 输入的文本信息直接显示在框中 |
| 密码框 password | 输入的文本以圆点或者星号的形式显示 |
| 单选框 radio | 进行单项的选择 如性别选择 多个radio的name属性相同会被当作一组来使用 必须用value为选项指定提交的值 |
| 复选框 checkbox | 进行多项选择, 爱好的选择。多个checkbox具有相同的name属性时会被当作一组来使用 必须用value为选项指定提交的值 |
| 隐藏字段 hidden | 如果有一些信息, 不希望用户看见, 又希望表单能够提交, 就可以用隐藏字段隐含在表单中 |
| 提交按钮 | 实现表单提交操作的按钮 可以通过value属性指定按钮显示的文字 |

| | |
|---------------|---|
| submit | |
| 重置按钮 reset | 重置表单到初始状态 |
| 按钮 button | 普通按钮,没有任何功能 需要配合JavaScript为按钮指定具体的行为。可以用value属性指定按钮显示的文字。 |
| 文件上传项 file | 提供选择文件进行上传的功能。 |
| 图像 image | 利用一张图片替代提交按钮的功能,不常用 |

- name属性
表单中可以有多输入项,输入项必须有name属性才可以被提交,如果输入项没有name属性,则表单在提交时会忽略它. 另外name属性的值是可以重复的
- value属性 可以给input输入框设置一个初始值
- readonly属性
使当前输入项变为只读,不能修改,但是提交时仍会被提交
- disabled
使当前输入项不可用,不能修改值,也不会被提交
- size属性
指定当前输入框的宽度
- checked属性
指定单选框/复选框被选中

(2) <textarea> 文本域

- 属性:
 - rows 指定文本域的行数(高度)
 - cols 指定文本域的列数(宽度)
 - readonly 只读
 - disabled 禁用

(3) <select> <option>

select 提供下拉选择功能
option 下拉选框中的选项 可以用value属性指定提交的值,如果不指定,将会提交标签内的文本

- 重要属性:
 - name: 下拉列表的名称
 - disabled 禁用下拉选框
- 其他属性:
 - size: 设置下拉选项中可见选项的个数
 - multiple 是否支持多选

3. 注册表单练习

略

1. 网页组织的两种常用方式

- 表格套表格方式定义网页结构 -- 目前不再是主流，只在一些结构简单的页面中有所使用
- DIV+CSS方式定义网页结构 -- 目前主流的网页开发方式，可以非常灵活的定义网页

2. 容器标签

本身没有任何特殊的能力，最主要的功能是用来包含其他标签组成一个整体。

常见的容器标签：div span p

<div> 是一个块级元素。这意味着它的内容自动地开始一个新行。

是一个行内元素。多个行内元素不会要求独占一行

<p> 是一个块级元素。用来声明一个段落。会在当前段落前后多出额外的空行。

3. CSS 的概念

CSS：层叠样式表

实现了网页中数据和样式的分离，使网页结构更加明细，解决了样式重复定义的问题，提高了开发效率和后期代码的可维护性，另外还增强了网页的美化能力。

~css中的注释只有一种：/* */

4. css的四种引入方式

- 方式一：通过style属性指定元素的样式

```
<p style="background-color:#FF0000; color:#FFFFFF">
  p标签段落内容。
</p>
```

- 方式二：通过<style>定义样式，可以在html的<head>标签中定义<style>标签，在其中为当前页面设定样式。

```
<head>
  <style type="text/css">
    p { color:#FF0000;}
  </style>
</head>
```

- 方式三：引入外部样式文件。可以在html的<head>标签中定义<link>标签，引入外部的css文件来修饰当前页面。

```
<link rel="stylesheet" href="css.css" >
```

- 方式四：可以通过@import url(xxx.css) 在css的内部引入一个css文件中定义的css样式片段。可以实现css代码的引入从而实现css代码的复用。

```
<style type="text/css">
  @import url(div.css);
  div { color:#FF0000;}
</style>
```

如果有多种方式为同一个元素设定了样式，则样式起作用的原则是：由上到下，由外到内，优先级由低到高。(编辑样式的代码哪个离标签近哪个有效)

5. css基本选择器

(1) 标签名选择器：

通过html标签的名字来选择标签的选择器

标签名{

```
div {
```

```

    color:red;
}
*{
    Color:red;
}

```

(2) 类选择器:

html的所有标签都具有一个通用的属性叫做class，通过它可以为标签指定类名，通过类选择器可以选择指定类名的元素。

.类名{

```

.aaa{
    color:red;
}
<div class="aaa">hahahaha</div>

```

(3) id选择器:

html的所有标签都具有一个通用的属性叫做id，通过它可以为标签指定id，id必须在整个html中唯一。通过id选择器可以选择出指定id的元素。

#id{

```

#bbb{
    color:red;
}
<div id="bbb">hehehe</div>

```

6. css扩展选择器

(1) 后代选择器

选择父元素中的**后代**元素.

父元素选择器 后代元素选择器{

```

p { color:#00FF00;}
p b { color:#FF0000;}
<p>P标签<b>刘德华</b><a href="#"><b>段落</b>样式</a></p>
<p>P标签段落</p>

```

(2) 子元素选择器

选择父元素中的子元素.

父元素选择器 > 子元素的选择器{

```

h1 > strong {color:red;}

```

这个规则会把第一个 h1 下面的两个 strong 元素变为红色，但是第二个 h1 中的 strong 不受影响：

```

<h1>
This is <strong>very</strong> <strong>very</strong> important.
</h1>
<h1>
This is <em>really <strong>very</strong></em> important.
</h1>

```

(3) 分组选择器

将多个选择器的选择结果进行一个或 的操作.

选择器1,选择器2,...{

```

p,div { color:#FF0000;}
<p>P标签显示段落。</p>
<div>DIV标签显示段落</div>
注：多个不同选择器要用逗号分隔开。

```

(4) 属性选择器

选择具有指定属性，或指定属性的值等与指定值的选择器.

选择器[属性名]{}

选择器[属性名='属性值']{}

如果希望把包含属性 (title) 的所有元素变为红色, 可以写作:

```
*[title] {color:red;}
```

与上面类似, 可以只对有 href 属性的锚 (a 元素) 应用样式:

```
a[href] {color:red;}
```

为了将同时有 href 和 title 属性的 HTML 超链接的文本设置为红色, 可以这样写:

```
a[href][title] {color:red;}
```

假设只希望选择 moons 属性值为 1 的那些 planet 元素:

```
planet[moons="1"] {color: red;}
```

上面的代码会把以下标记中的第二个元素变成红色, 但第一个和第三个元素不受影响:

```
<planet>Venus</planet>  
<planet moons="1">Earth</planet>  
<planet moons="2">Mars</planet>
```

(5) 相邻兄弟选择器

选择选择器选择到的元素的相邻的兄弟元素

选择器+兄弟元素名{}

```
h1 + p {margin-top:50px;}
```

(6) 伪元素选择器

其实就在html中预先定义好的一些选择器。称为伪元素。是因为CSS的术语。

:link 未点击的状态

:visited 被点击过的状态

注: 在不设置visited状态时active生效, 否则会出现visited覆盖active效果

:hover 鼠标移动到元素之上但是仍然未点击的状态

:active 被鼠标点击着的状态

7. 拓展: css中的常用属性简介