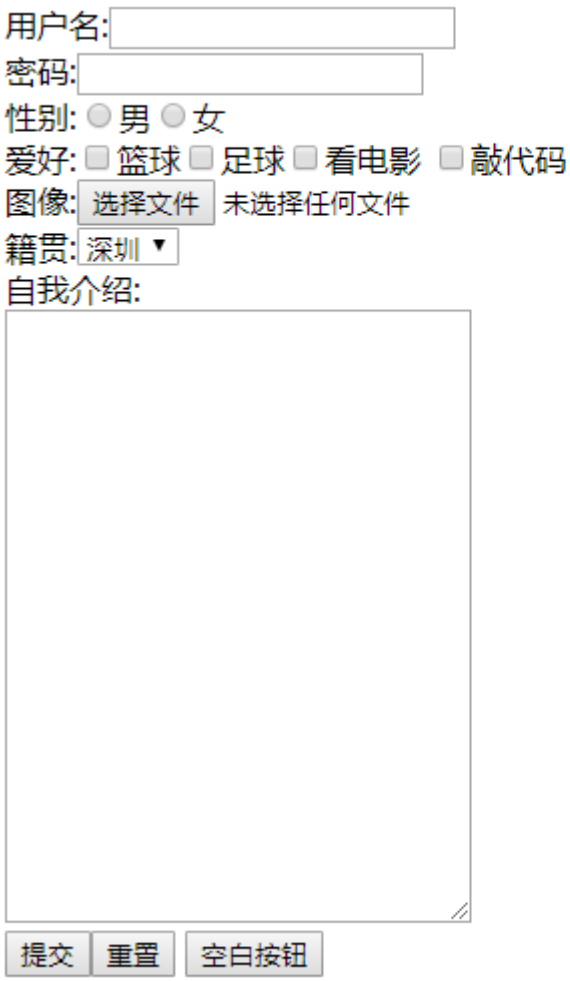
day21-HTML&CSS

学习目标

1. 能够使用表单form标签创建表单容器
2. 能够使用表单中常用的input标签创建输入项
3. 能够使用表单select标签定义下拉选择输入项
4. 能够使用表单textarea标签定义文本域
5. 能够使用CSS的基本选择器选择元素
6. 能够使用CSS的扩展选择器选择元素
7. 能够使用常见的CSS属性
8. 能够说出盒子模型的属性

案例一注册页面案例

一,案例需求



二，技术分析

1.表单标签【重点】

- 通过form来定义

```
<form action="http://www.baidu.com" method="post">
    ...
</form>
```

常用属性

action:提交路径,默认是当前页面,#

method:提交方式,常用的是get和post. 默认就是get

get和post区别

```
1. get方式提交的数据(请求参数)在地址栏可见(拼接在请求的路径后面), post方式不可见
2. get方式相对不安全, post方式相当安全一些
3. get方式对提交的数据(请求参数)的大小有限制的, post方式没有限制的
```

2.form的常见子标签

input:输入域, 通过type属性来指定类型

select :选择列表

textarea：文本域

2.1input：输入类型

```
<input type="xxx"/>
```

type属性,类型是由属性（type）定义的

- text:文本输入框
- password:密码域
- submit：提交按钮
- reset：重置按钮
- button:空白按钮
- radio:单选框
- checkbox:复选框
- hidden:隐藏字段
- file:文件

2.2select :选择菜单

```
<select name="">
    <option value="">显示的内容</option>
</select>
```

- option：选择菜单的选项

注意：

- name在select里面指定
- value在option里面指定
- option定义在select里面

2.3textarea:文本域

```
<textarea rows="20" cols="30" name="introduce"></textarea>
```

属性：

- cols列
- rows:行

3.通用属性

1.name

- 作为单选和多选框的分组
- 作为key上传到后台程序，后台程序通过key得到相对应的value。如果要想把数据提交到后台程序，一定要指定name属性

2.value

- 给按钮起名字
- 设置提交到服务器的值 name=value

4.设置默认值

- text,password:通过value属性

```
用户名: <input type="text" name="username" value="zs"/>
```

- radio checkbox:通过checked属性

```
性别: <input type="radio" name="sex" value="1"/>男
      <input type="radio" name="sex" value="0" checked="checked" />女
```

- select :在option上通过selected属性

```
籍贯: <select name="address">
        <option value="sz">深圳</option>
        <option value="bj">北京</option>
        <option value="sh" selected="selected">上海</option>
    </select><br/>
```

- textarea:直接在标签体中写

```
自我介绍: <textarea rows="5" cols="20" name="introduce">哈哈哈</textarea><br/>
```

三，思路分析

四，代码实现

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>表单标签</title>
</head>
<body>
    <!--一, 表单标签: 作用提交用户输入的数据的(收集用户输入的信息)
            action属性: 提交的路径(写远程的,也可以写本地的页面)

            method属性: 指定的提交的方式  get(get方式提交,默认) post(post方式提交)
                get和post方式区别:
                        1. get方式提交的数据(请求参数)在地址栏可见(拼接在请求的路径后面), post方式不可见
                        2. get方式相对不安全, post方式相当安全一些
                        3. get方式对提交的数据(请求参数)的大小有限制的, post方式没有限制的


            name属性作用: 1. 作为单选,复选的分组
                          2. 作为key把数据提交(如果你想把数据提交, name一定要设置)
            value属性作用: 1,给按钮取名字
                          2.作为值提交到服务器


    -->

    <form action="#" method="get">
        <!--隐藏字段: 页面不想看到, 但是又需要把数据提交到服务器的时候-->
        <input type="hidden" name="key" value="aaa"/>

        用户名: <input type="text" name="username" value="zs"/> <br/>
        密码: <input type="password" name="password"/><br/>
        性别: <input type="radio" name="sex" value="1"/>男
              <input type="radio" name="sex" value="0" checked="checked" />女
              <br/>
        爱好: <input type="checkbox" name="hobby" value="basketball"/> 篮球
              <input type="checkbox" name="hobby" value="football" checked="checked"/> 足球
              <input type="checkbox" name="hobby" value="film" checked="checked"/> 看电影
              <input type="checkbox" name="hobby" value="coding"/> 敲代码
              <br/>
        图像: <input type="file" name="icon"/><br/>
        籍贯: <select name="address">
                  <option value="sz">深圳</option>
                  <option value="bj">北京</option>
                  <option value="sh" selected="selected">上海</option>
              </select><br/>
        自我介绍: <textarea rows="5" cols="20" name="introduce">哈哈哈</textarea><br/>
        <input type="submit"/> <input type="reset" /> <input type="button" value="空白按钮"/>
    </form>
</body>
</html>
```

五,扩展

1. HTML5中新增的type类型

```
<input type="xxx"/>
```

注：不同的浏览器支持上有差异，有些浏览器依然不支持

值	描述
color	定义拾色器
date	定义日期字段（带有 calendar 控件）
email	定义用于 e-mail 地址的文本字段，如果输入不正确的邮箱地址有验证的功能
number	定义带有 spinner 控件的数字字段
range	定义带有 slider 控件的数字字段
search	定义用于搜索的文本字段，当输入内容的时候，搜索框后边会自动出现一个小×，点击这个小×，可以清除输入的内容。
tel	定义用于电话号码的文本字段，在手机上操作会出现输入数字的键盘。
url	定义用于 URL 的文本字段，在手机上操作会出现输入网址的键盘。

2.placeholder:用户提示

案例二:使用Div+CSS完成谷歌搜索页面

一,需求分析

二,技术分析

1.div和span

1.1什么是div和spn

div是html里面的一个标签

. 没有特定的含义,作为容器. 一般用于配合css完成网页的基本布局,

span也是一个标签,没有特定含义,一般作为文本容器

1.2div和span的区别

div是块级元素会独占一行,span是行内元素不会独占一行

div中可以嵌套其它所有的标签, span标签中只能嵌套文本/图片/超链接

2.CSS概述和体验

2.1什么是CSS

- 层叠样式表
- 层叠: 样式的层层叠加(eg: 刷墙)
- 样式表: 样式的集合, 说白了就是属性的集合

学习HTML的核心是标签, 学习CSS的核心是属性, 选择器

2.2 css能做什么

- 美化页面,修饰页面
- html的标签作用展示页面,定义页面的, CSS的作用修饰页面
- eg: 把HTML当做毛坯房(骨架), 把CSS当做装修(样式)

2.3为什么要学习CSS

- 我们在上次课中已经遇到了一些样式的问题, font标签的字体不能比1还小不能比7还大, 超链接标签的下划线去不掉....
- 通过标签来修改样式的缺点:

    1.需要记忆哪些标签有哪些属性, 如果该标签没有这个属性, 那么设置了也没有效果

    2.当需求变更时我们需要修改大量的代码才能满足现有的需求
- 所以在企业开发中修改样式都是交给CSS来做,通过CSS来修改样式的好处:

    1.不用记忆哪些属性属于哪个标签

    2.当需求变更时我们不需要修改大量的代码就可以满足需求

2.4CSS语法

```
{
        属性:属性值 属性值;
        属性:属性值 属性值
}
```

注意

- 属性和属性值用:连接

- 如果有多个属性值用空格隔开
- 如果有多个属性,属性和属性之间用;隔开 最后一个;不写

3.html和css常见的结合方式

3.1通过标签的style属性来结合【了解】

```
<!--通过style属性-->
<p style="属性名称:属性值;..."></p>
```

3.2通过style标签来结合【掌握】

```
<head>
        <style type="text/css">
        标签名称{
            属性名称: 属性对应的值;

        }
        </style>
</head>

注意点:
        1.style标签必须写在head标签的开始标签和结束标签之间(也就是必须和title标签是兄弟关系)
        2.style标签中的type属性其实可以不用写, 默认就是type="text/css"
        3.设置样式时必须按照固定的格式来设置. key: value; 其中:不能省略, 分号大多数情况下也不能省略
```

3.3通过link标签结合【掌握】

1. 创建一个css文件(后缀是css)
2. 通过link标签引入

```
<head>
        <link rel="stylesheet" href="../../css/myCss.css" />
</head>

link标签属性:
        - href:css文件路径
```

3.4 三种结合方式优先级

就近原则(相对于代码,也就是要修饰的标签)

4.选择器

4.1什么是选择器

css修饰页面,作用某个标签.CSS选择器就是控制html标签,说白了就是找到标签的

4.2基本选择器【掌握】

选择器分类	作用	语法	细节
标签选择器	通过标签名选择同名的所有的标签	标签名 { }	
类选择器	通过 class 属性的值选择元素	.类名 { }	前提：先给标签进行分类 使用 class 属性分类 类名：不能以数字开头
ID 选择器	通过属性 ID 选择元素	#ID { }	前提：先给标签指定 ID 属性 唯一：建议 ID 在同个网页中唯一，不要重复
通用选择器	**选中整个网页中所有的元素**	* { }	

- 通用选择器 < 标签选择器 < 类选择器 < ID选择器

4.3扩展选择器

由基本选择器组合而成，可以有更加灵活的选择方式

扩展选择器	格式	作用	语法符号
层级选择器	父选择器 子孙选择器{ }	选择某个元素下面的所有子孙元素	**空格**
属性选择器	标签名[属性名] 标签名[属性名="属性值"]	包含属性名即可 属性名=属性值才可以 通过属性名和属性值选中符合条件的元素	**[ ] 中括号**
并集选择器	**选择器 1,选择器 2**	**同时使用 2 个选择器的和**	**，逗号**

4.4 伪类选择器【了解】

伪类选择器	格式	作用	语法符号
伪类选择器	**链接：** **a:link 正常状态** **a:visited 访问过的** **a:hover 鼠标悬停** **a:active 正在激活**	**元素在操作的过程中，不同的状态呈现不同的样式**	**：冒号**

5.CSS常用的属性

5.1背景属性

功能	属性名	属性取值
背景色	background-color	颜色常量，如：red 使用十六进制，如：#123 使用 rgb(红,绿,蓝)
背景图片	background-image	url(图片文件)
平铺方式	background-repeat	repeat 默认。背景图像将在垂直方向和水平方向重复。 repeat-x 背景图像将在水平方向重复。 repeat-y 背景图像将在垂直方向重复。 no-repeat 背景图像将仅显示一次。

- 代码

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>

    <style>
        body{
            /*背景图片*/
            background-image: url("../../img/a.gif");
            /*设置平铺*/
            background-repeat: repeat;
        }

        div{
            width: 200px;
            height: 200px;
            /*背景颜色
            background-color: red;
            */
        }

    </style>
</head>
<body>
    <div></div>
</body>
</html>
```

5.2文本样式

功能	属性名	属性取值
颜色	color	颜色常量，如：red 使用十六进制，如：#123 使用 rgb(红，绿，蓝)函数
文字修饰	text-decoration	underline 下划线 overline 上划线 line-through 删除线
文本缩进	text-indent	用于缩进文本，可以使用 em 单位，表示缩进 2 个字符，无论字符的大小。
文本对齐	text-align	left 把文本排列到左边。默认值：由浏览器决定。 right 把文本排列到右边。 center 把文本排列到中间。

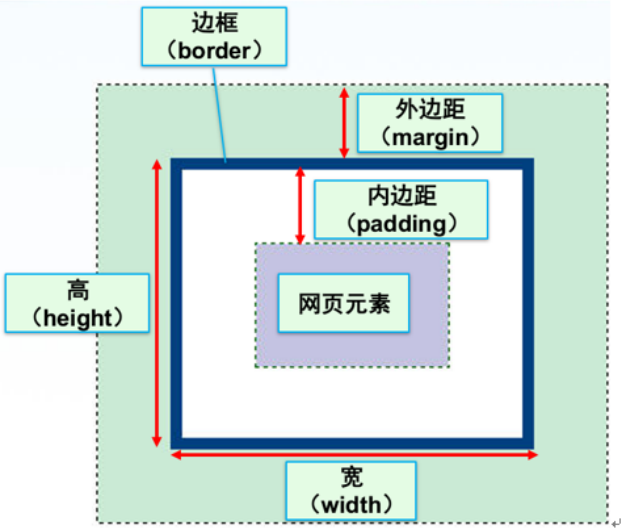
5.3字体属性

功能	属性名	作用
字体名	font-family	设置字体，本机必须要有这种字体
设置大小	font-size	单位：像素
设置样式	font-style	字体设置为斜体 italic 浏览器会显示一个斜体的字体样式。 normal 默认值。浏览器显示一个标准的字体样式。
设置粗细	font-weight	bolder 加粗

6.盒子模型

任何一个网页元素包含由这些属性组成：内容(content)、内边距(padding)、边框(border)、边界(margin)，这些属性我们可以用日常生活中的常见事物——盒子作一个比喻来理解，所以叫它盒子模型。整个网页由各种小盒子组成。

6.1盒子属性



属性	作用
width	宽度
height	高度
margin	外边距
padding	内边距
border	边框：粗细 线型 颜色

6.2 外边距属性 margin

- 标签和标签之间的距离就是外边距

```
一,设置外边距
1.连写
margin: 上 右 下 左;

2.非连写
margin-top: ;
margin-right: ;
margin-bottom: ;
margin-left: ;

3. 注意地方:
3.1 连写注意的地方
        - margin:10px; 上下左右都是10px
        - margin:10px,20px; 上下是10px,左右20px
        - margin:10px,20px,30px;上是10px,右是20px,下30px,左是20px
3.2 设置外边距特点
        外边距的那一部分是没有背景颜色的
```

6.3内边距属性

- 就是标签里面的内容距离边框距离

```
一,设置内边距
1.连写
padding: 上 右 下 左;

2.非连写
padding-top: ;
padding-right: ;
padding-bottom: ;
padding-left: ;

3. 注意地方:
3.1 连写注意的地方
        - padding:10px; 上下左右都是10px
        - padding:10px,20px; 上下是10px,左右20px
        - padding:10px,20px,30px;上是10px,右是20px,下30px,左是20px
3.2 设置内边距特点
        给标签设置内边距之后, 标签占有的宽度和高度会发生变化
        给标签设置内边距之后, 内边距也会有背景颜色
```

6.4边框属性

属性	边框样式	取值
border-style	边框线型	**dotted** 定义点状边框。在大多数浏览器中呈现为实线。 **dashed** 定义虚线。在大多数浏览器中呈现为实线。 **solid** 定义实线。 **double** 定义双线。双线的宽度等于 border-width 的值。
border-width	边框宽度	**length** 允许您自定义边框的宽度。
border-color	边框颜色	**常量** 规定颜色值为颜色名称的边框颜色（比如 red）。 **十六进制** 十六进制值的边框颜色（比如 #ff0000）。 **函数** 为 rgb 代码的边框颜色（比如 rgb(255,0,0)）。
border-radius	边框圆角	指定圆角的半径
border	简写	线型 宽度 颜色

7.浮动属性

7.1 概述

当某一个元素设置浮动之后, 那么这个元素就会脱离文档流. 感觉就像在上面一层覆盖着,有点像PS里面图层. 之前用来做图文混排,后面通常和div一起做页面的布局了

7.2设置浮动

- 语法

```
float: 取值 left(左浮动)
                    right(右浮动)
```

- 特点:在浮动流中是不区分块级元素/行内元素/行内块级元素的, 都可以设置宽度和高度
- 浮动规则

```
1.相同方向上的浮动元素, 先浮动的元素会显示在前面, 后浮动的元素会显示在后面
2.不同方向上的浮动元素, 左浮动会找左浮动, 右浮动会找右浮动
3.浮动元素浮动之后的位置, 由浮动元素浮动之前在标准流中的位置来确定
```

7.3清除浮动

- 语法

```
clear: 取值  none: 默认取值,不清除
                        left: 不要找前面的左浮动元素
                        right: 不要找前面的右浮动元素
                        both: 不要找前面的左浮动元素和右浮动元素
```

- 清除浮动的方式

```
方式一: 在当前的元素里面添加clear属性清除
方式二: 添加一个新的盒子添加clear属性清除
```

三,思路分析

四,代码实现

- 案例二下的index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>谷歌搜索页面</title>

    <style>
        div{
            text-align: center;
        }

        input[type='text']{
            width: 400px;
            height: 30px;
        }
        input[type='submit']{
            width: 100px;
            height: 30px;
        }

        .content{
            margin-top: 60px;
        }

        .wiki{
            margin-top: 20px;
        }

        .map{
            margin-top: 60px;
        }

        .footer{
            margin-top: 100px;
        }

        .info{
            margin-top: 20px;
        }

        .right{
            float: right;
        }

        .header{
            clear: both;
        }

        button{
            border: none;
            width: 60px;
            height: 28px;
            background-color:#5891F4 ;
```

```
            color: white;
            border-radius: 4px;

        }

    </style>
</head>
<body>

    <!--右边的部分-->
    <div class="right">
        <a href="#">Gmail</a>
        <a href="#">图片</a>
        <button>登录</button>
    </div>

    <!--头部-->
    <div class="header">
        <!--logo-->
        <div class="logo">
            <img src="../img/logo.png">
        </div>
        <!--链接-->
        <div class="link">
            <a href="#">全 部</a>
            <a href="#">首 页</a>
            <a href="#">首 页</a>
            <a href="#">首 页</a>
            <a href="#">首 页</a>
            <a href="#">首 页</a>
            <a href="#">首 页</a>
            <a href="#">首 页</a>
        </div>

    </div>

    <!--中间内容部分-->
    <div class="content">
        <!--搜索-->
        <div class="search">
            <input type="text" name="key"/>
            <input type="submit" value="谷歌一下"/>
        </div>
        <!--wiki-->
        <div class="wiki">
            <!--维基百科 维基文库 自由的图书馆 | 更多>>-->
            <a href="#">维基百科</a>
            <a href="#">维基文库</a>
            <a href="#">自由的图书馆</a>
            |
            <a href="#">更多>></a>

        </div>
```

```
        <!--谷歌地图-->
        <div class="map">
            <img src="../img/ic.jpg"/>
            <a href="#">谷歌地图带你吃喝玩乐，全心全意为人民服务</a>
        </div>

    </div>

    <!--底部部分-->
    <div class="footer">
        <!--谷歌设置主页-->
        <div class="home">
            <a href="#">把谷歌设为主页</a>
        </div>

        <div class="info">
            <a href="#">广告 |</a>
            <a href="#">广告 |</a>
            <a href="#">广告 |</a>
            <a href="#">广告 |</a>
            <br/>
            ©2018 Google 提供

        </div>

    </div>


</body>
</html>
```