

# DAY02 CSS基础和表单基础元素

2023年3月22日 9:20

## 表单（收集用户信息）

### 表单域

**form**

action 数据提交的地址

method 数据提交的方式

get 明文显示在地址栏（默认）

post

### 表单控件

**input**

type 表单控件的类型

**文本输入框** type="text"

value 初始值

placeholder 没有初始值的时候的提示文字

maxlength 最大输入长度

name 表标识

**密码输入框** type="password"

value 初始值

placeholder 没有初始值的时候的提示文字

maxlength 最大输入长度

name 表标识

**提交按钮** type="submit"

**重置按钮** type="reset"

**普通按钮** type="button"

- value 按钮的文字

- 单标签

- 不独占一行

## get和post的区别

1. get用来获取数据 获取列表 获取详情  
post 用来发送数据 发送消息 发送表单数据
2. get发送的数据会明文展示在url, 相对不安全  
post发送的数据会放在请求头中
3. get发送的数据有大小限制, post没有

## CSS语法

选择器 声明

选择器: 要添加样式的对象

声明/样式: 属性 属性值

```
选择器{  
    属性1:属性值1;  
    属性2:属性值2;  
    ...  
}
```

## 注意事项

- 1.先写html, 再写css
- 2.所有的符号是英文
- 3.声明的分号不能忘记
- 4.只有最后一条声明的分号可以省略

## 样式表

### 内部样式表

在html文件中head标签里面的最下面用style标签中来书写样式

### 外部样式表

- 1.需要在外部新建一个css文件 在里面写css代码
- 2.和html文件建立联系
  - 01.在html文件中head标签里面的最下面用link标签引入  
`<link rel="stylesheet" href="./wai.css">`  
rel="stylesheet" 关联外部样式表  
href 路径 必须写
  - 02.在html文件中head标签里面的最下面用style中写  
`@import url(./style.css);`  
url 路径 必须写

### link和@import的区别

- link是html提供的方式, @import是css提供的方式
- link引入的文件是一边下载一边加载, @import引入的文件是下载后

再加载

-link引入的文件可以被js操作, @import不可以

## 内联样式表/行内样式表

### 行内样式表

在标签中用html属性style 属性值是css代码

优先级最高

### 样式表权重

行内样式表权重最大

内部和外部看书写顺序, 谁在下, 谁优先

### 元素选择器 (标记/类型/标签)

语法: 标签名{} div{}

作用: 选择相同的标签

### class选择器

html: <div class="class名字">div1</div>

css: .class名字{}

### 名字的规则

1. 可以英文 连字符- 下划线\_ 英文字母开头
2. 不可以数字开头
3. 不可以出现中文 特殊符号
4. 避免出现关键词
5. 尽量见名知意
6. 多个单词连接用- \_ 驼峰 例如: userName

### id选择器

html: <div id="id名字">div</div>

css: #id名字{}

id名字一个页面只能出现一次, 是一个唯一的不重复的值

## 通配符 \*

选中页面上所有的元素

常用：

去除内外边距

```
*{  
  padding:0;  
  margin:0  
}
```

## 群组选择器

选择器1, 选择器2, 选择器3...{}

代码：

```
.box,  
#tom,  
h2,  
span {  
  background-color: blue;  
  color: yellow;  
  font-size: 30px;  
}
```

## 后代选择器（包含选择器）

选择器1 选择器2 选择器3...{}

- 1.后面的选择器必须是前面的选择器的后代
- 2.样式会作用在最后一个选择器身上

代码：

```
ul li a {  
  font-size: 50px;  
}
```

后代和群组一起使用的时候，以逗号分割，逗号两边各为一组

代码：

```
.father .son1,  
.father .son2 {  
  background-color: red;
```

```
font-size: 30px;
}
```

## 子代选择器

选择器1>选择器2>选择器3...{ }

- 1.后面的选择器必须是前面的选择器的子代
- 2.样式最终作用到最后一个选择器身上

代码:

```
p>span {
    background-color: yellow;
}
```

## 伪类选择器

**a:**

- :link 超链接未访问
- :visited 超链接访问过
- :hover 鼠标经过
- :active 鼠标按下

**普通元素:**

- :hover 鼠标经过
- :active 鼠标按下

:focus 元素获取焦点

当伪类选择器同时写给超链接的时候，有先后顺序  
爱恨原则 love-hate