

css笔记

一、css的简介

1、什么是css

层叠样式表，css是对html进行样式修饰语言

层叠：就是层层覆盖叠加，如果不同的css样式对同一html标签进行修饰，样式有冲突的部分应用优先级高的，不冲突的部分共同作用

样式表：就是css属性样式的集合

2、css的作用

- (1) 修饰html的 使其html样式更加好看
- (2) 提高样式代码的复用性
- (3) html的内容与样式相分离 便于后期维护

3、css的引入方式和书写规范

(1) 内嵌样式

内嵌样式是把css的代码嵌入到html标签中

```
<div style="color:red;font-size: 100px;">你好啊 小朋友</div>
```

语法：

- (1) 使用style属性将样式嵌入到html标签中
- (2) 属性的写法：属性：属性值
- (3) 多个属性之间使用分号;隔开

不建议使用

(2) 内部样式

在head标签中使用style标签进行css的引入

```
<style type="text/css">div{color:red;font-size: 100px;}</style>
```

语法：

- (1) 使用style标签进行css的引入

```
<style type="text/css">
```

属性： type： 告知浏览器使用css解

析器去解析

- (2) 属性的写法：属性：属性值
- (3) 多个属性之间使用分号;隔开

(3) 外部样式

将css样式抽取成一个单独css文件 谁去使用谁就引用

```
<link rel="stylesheet" type="text/css"
```

```
href="demo1.css"/>
```

语法:

(1) 创建css文件 将css属性写在css文件中

(2) 在head中使用link标签进行引入

```
<link rel="stylesheet" type="text/css"
```

```
href="css文件地址"/>
```

rel:代表要引入的文件与html的关系

type: 告知浏览器使用css解析器去解析

href: css文件地址

(3) 属性的写法: 属性: 属性值

(4) 多个属性之间使用分号;隔开

(4) @import方式

```
<style type="text/css">
```

```
@import url("css地址");
```

```
</style>
```

link与@import方式的区别:

(1) link所有浏览器都支持 import部分低版本IE不支持

(2) import方式是等待html加载完毕之后在加载

(3) import方式不支持js的动态修改

二、css选择器

1、基本选择器

(1) 元素选择器

语法: html标签名 {css属性}

示例:

```
<span>hello css!!!</span>
```

```
<style type="text/css">
```

```
span{color:red;font-size:100px; }
```

```
</style>
```

(2) id选择器

id唯一性

语法: #id的值 {css属性}

示例:

```
<div id="div1">hello css1!!!</div>
<div id="div2">hello css2!!!</div>
<style type="text/css">
    #div1{background-color: red;}
    #div2{background-color: pink;}
</style>
```

(3) class选择器

语法: .class的值 {css属性}

示例:

```
<div class="style1">div1</div>
<div class="style1">div2</div>
<div class="style2">div3</div>
<style type="text/css">
    .style1{background-color: red}
    .style2{background-color: pink}
</style>
```

***选择器的优先级: id>class>元素

2、属性选择器

语法: 基本选择器[属性= '属性值'] {css属性}

示例:

```
<form action="">
    name:<input type="text" /><br/>
    pass:<input type="password" /><br/>
</form>
<style type="text/css">
    input[type='text'] {background-color: yellow}
    input[type='password'] {background-color: pink}
</style>
```

3、伪元素选择器

a标签的伪元素选择器

语法:

静止状态	a:link{css属性}
悬浮状态	a:hover{css属性}
触发状态	a:active{css属性}
完成状态	a:visited{css属性}

示例:

```
<a href="#">点击我吧</a>
<style type="text/css">
    a:link{color:blue}
    a:hover{color:red}
    a:active{color:yellow}
    a:visited{color:green}
</style>
```

4、层级选择器

语法: 父级选择器 子级选择器

示例:

```
<div id="d1">
    <div class="dd1">
        <span>span1-1</span>
    </div>
    <div class="dd2">
        <span>span1-2</span>
    </div>
</div>
<div id="d2">
    <div class="dd1">
        <span>span1-1</span>
    </div>
    <div class="dd2">
        <span>span1-2</span>
    </div>
</div>

<style type="text/css">
```

```
#d1 .dd2 span{color:red}
</style>
```

三、css属性

1、文字属性

font-size:大小
font-family:字体类型

2、文本属性

color:颜色
text-decoration:下划线
属性值: none underline

text-align:对齐方式
属性值: left center right

```
<div>hello css!!!</div>
```

```
<a href="#">click me!!!</a>
```

```
<style type="text/css">
```

```
div{color:red;text-decoration: underline;text-align: right }
```

```
a{text-decoration: none;}
```

```
</style>
```

3、背景属性

background-color:背景颜色

background-image:背景图片

属性值: url("图片地址");

background-repeat:平铺方式

属性值: 默认横向纵向平铺

repeat:横向纵向平铺

no-repeat:不平铺

repeat-y: 纵向

repeat-x: 横向

```
body{
```

```
background-color: black;
```

```
background-image: url("images/dog.gif");
background-repeat: repeat-y;
}
```

4、列表属性

list-style-type:列表项前的小标志

属性值: 太多了

list-style-image:列表项前的小图片

属性值: url("图片地址");

```
<ul>
    <li>黑马程序员</li>
    <li>黑马程序员</li>
    <li>黑马程序员</li>
    <li>黑马程序员</li>
</ul>
<style type="text/css">
    /* ul{list-style-type: decimal-leading-zero;} */
    ul{list-style-image: url("images/forward.gif");}
</style>
```

5、尺寸属性

width:宽度

height:高度

```
<div id="d1">div1</div>
<div id="d2">div2</div>
<style type="text/css">
    #d1{background-color: red;width: 200px;height:
200px;}
    #d2{background-color: pink;width: 200px;height:
200px;}
</style>
```

6、显示属性

display:

属性值: none:隐藏

block:块级显示

inline: 行级显示

```
<form action="">
    name:<input id="name" type="text" /><span
id="span">对不起 输入不符合要求</span>
    <br>
    pass:<input id="pass" type="password" />
    <br>
    <input id="btn" type="button" value="button" />
</form>
<style type="text/css">
    span{color:red;display: none}
</style>
<script type="text/javascript">
    document.getElementById("btn").onclick =
function() {
document.getElementById("span").style.display = "inline";
    };
</script>
```

7、浮动属性

float:

属性值: left right

clear:清除浮动 left right both

缺点: (1)影响相邻元素不能正常显示

(2)影响父元素不能正常显示

四、css盒子模型

border:

border-width:边框的宽度

border-color:边框的颜色

border-style:边框的线型

border-top:上边框
border-bottom:下边框
border-left:左边框
border-right:右边框

padding:

代表边框内壁与内部元素之间的距离
padding:10px;代表上下左右都是10px
padding:1px 2px 3px 4px;上右下左
padding:1px 2px;上下/左右
padding:1px 2px 3px;
padding-top:单独设置

margin:

代表边框外壁与其他元素之间的距离
margin:10px;代表上下左右都是10px
margin:1px 2px 3px 4px;上右下左
margin:1px 2px;上下/左右
margin:1px 2px 3px;
margin-top:单独设置

javascript笔记

一、js的简介

1、js是什么

js是可以嵌入到html中，是 基于对象 和 事件驱动 的 脚本语言
特点：

- (1)交互性
- (2)安全性：js不能访问本地磁盘
- (3)跨平台：浏览器中都具备js解析器

2、js能做什么

(1) js能动态的修改(增删)html和css的代码

(2) 能动态的校验数据

3、js历史及组成

ECMAScript BOM(浏览器对象模型) DOM(文档对象模型)

4、js被引入的方式

(1) 内嵌脚本

```
<input type="button" value="button"
onclick="alert('xxx')" />
```

(2) 内部脚本

```
<script type="text/javascript">
    alert("xxx");
</script>
```

(3) 外部脚本

首先先创建一个js文件

其次在html中引入

```
<script type="text/javascript" src="demo1.js">
</script>
```

js代码放在哪?

放在哪都行 但是在不影响html功能的前提下 越晚加载越好

二、js基本语法

1、变量

(1)

```
var x = 5;
x = 'javascript';
var y = "hello";
var b = true;
```

(2)

```
x = 5;
```

2、原始数据类型

(1) number: 数字类型

(2) string: 字符串类型

- (3) boolean: 布尔类型
- (4) null: 空类型
- (5) undefined: 未定义

注意: number、boolean、string 是伪对象

类型转换:

number \ boolean 转成 string

toString();

string \ boolean 转成 number

parseInt()

parseFloat()

boolean 不能转

string 可以将数字字符串转换成 number 如

果 “123a3sd5” 转成 123

强制转换

Boolean() 强转成布尔

数字强转成布尔 非零就是 true 零就是

false

字符串强转成布尔 非 “” (空字符串) 就是

true 空字符串 “” 就是 false

Number() 强转成数字

布尔转数字 true 转成 1 false 转成 0

字符串转数字 不能强转

3、引用数据类型

java: Object obj = new Object();

js: var obj = new Object();

var num = new Number();

4、运算符

(1) 赋值运算符

var x = 5;

(2) 算数运算符

+ - * / %

+: 遇到字符串变成连接

-: 先把字符串转成数字然后进行运算

*: 先把字符串转成数字然后进行运算

/: 先把字符串转成数字然后进行运算

(3) 逻辑运算符

&& ||

(4) 比较运算符

< > >= <= != ==

===: 全等: 类型与值都要相等

(5) 三元运算符

3<2?"大于":"小于"

(6) void运算符

xxxxxx

(7) 类型运算符

typeof: 判断数据类型 返回我的数据类型

instanceof: 判断数据类型 是否是某种类型

```
var obj = new Object();
```

```
alert(typeof obj);//object
```

```
alert(obj instanceof Object);//true
```

5、逻辑语句

(1) if-else

//条件:

//数字非0 字符串非空===true

```
if(9){
    alert("true—");
} else {
    alert("false—");
}
```

(2) switch

```
var x = "java";
```

```
switch(x){
    case "css":
        alert("css");
}
```

```
        break;
    case "js":
        alert("js");
        break;
    case "java":
        alert("java");
        break;
    default:
        alert("def");
}
```

(3) for

```
    for(var i = 0;i<5;i++){
        alert(i);
    }
```

(4) for in

```
var arr = [1, 3, 5, 7, "js"];
for(index in arr) { //index代表角标
    //alert(index);
    alert(arr[index]);
}
```

三、js的内置对象

四、js的函数

五、js事件

六、js的bom

七、js的dom