

Day09

java课程-李彦伯

Day09

CSS

CSS样式的种类

内部样式

行内样式

外部样式

选择器

常用选择器

不太常用的选择器(写在这方便今后查找)

文本相关样式

font-style

font-weight

font-size

font-family

连写形式

text-decoration

text-align

text-indent

color

背景相关样式

background-color

background-image

background-repeat

background-position

background-attachment

连写形式

边框

盒子模型

padding

margin

盒子模型

标准流

display

浮动流

float

clear

定位流

相对定位

绝对定位

固定定位

zIndex

CSS

CSS(Cascading Style Sheets)是层叠样式表表的简写,用于美化我们所写的HTML页面

- 格式

```
<style type="text/css">
选择器{
属性1:属性值;
属性2:属性值;
属性3:属性值1 属性值2 属性值3;
...
}
</style>
```

选择器严格区分大小写,属性和属性值不区分大小写
属性与属性之间使用分号隔开最后一个可以省略
如果一个属性有多个属性值,需要使用空格隔开
type属性可以省略

CSS样式的种类

内部样式

在head标签中使用 `<style>` 标签

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style type="text/css">
h1{
    color:red;
    font-style: italic;
}
</style>
</head>
<body>
<h1>好好学习,天天向上</h1>
</body>
</html>
```

行内样式

直接在标签中写**style**属性进行赋值,**style**属性的”“就相当于内部样式的{},把需要的属性按照上述要求写进去即可

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1 style="color: red; font-style: italic;">好好学习,天天向上</h1>
</body>
</html>
```

外部样式

单独写一个文件命名为xxx.css,将css代码写入文件,在 `<head>` 中进行引用

```
h1{
    color:red;
    font-style: italic;
}
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<!--type可以省略-->
<link rel="stylesheet" type="text/css" href="css/index.css">
</head>
<body>
<h1>好好学习,天天向上</h1>
</body>
</html>
```

注意如果了一个标签有多个css样式按照就近原则进行覆盖

选择器

我们要想设置某些标签的显示样式,就必须让css找到对应的标签,我们可以通过选择器来找到对应的标签

常用选择器

- 标签选择器 `标签类型{}`,直接写标签的名字就行

- id选择器 `#id名称{}`, id不能重复, 需要给标签添加一个id属性
- 类选择器 `.class名称{}` class可以重复, 需要给标签添加class属性
- 并集选择器, `选择器1, 选择器2{}`
- 属性选择器 `标签[属性="具体属性值"]{}` 例如
`input[type="text"]{}`

不太常用的选择器(写在这方便今后查找)

- 后代选择器 `选择器1 选择器2{}` 会找到选择器1下的选择器2的所有后代
- 子元素选择器 `选择器1>选择器2{}`, 选择器1下的所有所有子标签符合选择器2的条件
- 交集选择器 `选择器1选择器2{}`
- 相邻兄弟选择器, `选择器1+选择器2{}`
- 通用兄弟选择器, `选择器1~选择器2{}`
- 选中同级别的第一个标签, `标签:first-child{}`
- 选中同级别中同类型的的第一个标签, `标签:first-of-type`
- 选中同级别的最后一个标签, `标签:last-child{}`
- 选中同级别中同类型的的最后个标签, `标签:last-of-type`
- 选中同级别第几个标签, `标签:nth-child(3){}`
- 选中同级别中同类型的的第几个标签, `标签:nth-of-type(3)`

文本相关样式

用于设置文本类的相关样式

font-style

设置文字样式,常用取值为italic和normal

```
font-style: italic;
```

font-weight

设置文字的粗细,常用取值 lighter,bold和bolder还可以进行数字取值100,900

```
font-weight: bold;
```

font-size

设置文字大小,取值是像素

```
font-size: 100px;
```

font-family

设置字体"宋体"或者"微软雅黑"

```
font-family: "宋体";
```

连写形式

```
font: italic bold 100px "楷体" ;
```

text-decoration

文本装饰属性,常用取值underline(下划线)line-through(删除线)overline(上划线)none(什么都没有)
可以用none去掉a标签的下划线

text-align

对齐方式: left right center

text-indent

缩进方式 2em代表两个文字宽度

color

设置颜色"red"或者rgb(255,0,0)或者rgba(255,0,0,1)或者#十六进制

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style>
    h1{
        text-align: center;
        color: rgb(255,0,0);
        font-style: italic;
        font-weight: bold;
        font-size: 100px;
        font-family: "宋体";
        font: italic bold 100px "楷体" ;
    }
    h6 {
        text-align: center;
        font-style: italic;
    }
    p {

        color: black;
        text-align: center;
    }

</style>
</head>
<body>
```

```
<h1>登鹳雀楼</h1>
<h6>王之涣</h6>
<p style="text-indent: 2em">白日依山尽，
</p>
<p>黄河入海流.</p>
<p>欲穷千里目,</p>
<p style="text-decoration: line-through
h;">更上一层楼.</p>
</body>
</html>
```

背景相关样式

设置标签的背景颜色

background-color

设置颜色"red"或者rgb(255,0,0)或者rgba(255,0,0,1)或者#十六进制

background-image

设置背景图片, background-image:
url("img/1.jpg");,会自动平铺

background-repeat

设置平铺方式,有四个值,repeat(默认),no-repeat(不平铺),repeat-x(水平平铺),repeat-y(垂直平铺)

background-position

设置背景定位方式格式为: 水平方向数值 垂直方向数值 , 水平各有left center right ,垂直有top center bottom,也可以是具体的像素值比如 100px 100px

background-attachment

设置背景的关联方式常用有两个scroll(会随着滚动条的滚动而滚动),fixed(不会随着滚动而滚动)

连写形式

background:颜色 图片 平铺方式 关联方式 定位方式, 任意写都可以

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style>

    *{
        margin: 0px;
        padding: 0px;
    }
    div{
        width:100%;
        height:1000px;
/*      background-color: green;
        background-image: url("img/1.JPG
G");
        background-repeat: no-repeat;
        background-position: right bottom;
        background-attachment: fixed; */
        background: green url("img/1.JPG
G") no-repeat right bottom fixed;
    }

</style>
</head>
<body>
```

```
<div></div>
```

```
</body>
```

```
</html>
```

边框

border 属性,连写格式 宽度 样式 颜色


```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style>
/*只要是标签就能设置边框*/
.box{
    width: 100px;
    height: 100px;
    background-color: red;
    border: black 10px solid ;
}
</style>
</head>
<body>
<div class="box"></div>
</body>
</html>
```

盒子模型

padding

内容到边框的距离叫做内边距,内边距属性按照的顺序是上又下左顺序进行设置
也可以分开进行设置

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>

<style>
div{
    width:200px;
    height:80px;
    border: 1px black solid;
    background-color: red;
}

.box1{
    padding-top: 20px;
}
.box2{
    padding-right: 40px;
}
.box3{
    padding-bottom: 80px;
}
.box4{
    padding-left: 160px;
}
.box5{
    padding: 20px 40px 80px 160px;
}
```

```
/*设置内边距后元素的自身的宽高会发生改变
    设置完内边距后颜色也会被设置上
*/

</style>
</head>
<body>
    <div class="box1">内边距测试内边距测试内
    边距测试内边距测试内边距测试内边距测试内边距测
    试</div>
    <hr>
    <div class="box2">内边距测试内边距测试内
    边距测试内边距内边距测试内边距测试测试内边距测
    试</div>
    <hr>
    <div class="box3">内边距测试内边距测试内
    边距测试内边距测试内边距测试内边距测试</div>
    <hr>
    <div class="box4">内边距测试内边距测试内
    边距测试内边内边距测试内边距测试距测试</div>
    <hr>
    <div class="box5">内边距测试内边距测试内
    边距测试内边距测试内边距测试内边距测试内边距测试
    内边距测试</div>
    <hr>
</body>
</html>
```

1. 设置内边距后元素的自身的宽高会发生改变
2. 设置完内边距后颜色也会被设置上

margin

外边距,元素边框与边框之间的距离,依次设置的上右下左

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style>
*{
    padding: 0px;
    margin: 0px;
}
span {
    border: 1px solid black;
    width: 50px;
    height: 50px;
}
div {
    width:100%;
    height: 40px;
    border: 1px solid black;
}
/*外边距那部分没有背景颜色*/
.box1{
    background-color:red;
    margin: 20px 40px 60px 80px;
}
</style>
</head>
<body>
<span class="box1">我是span1</span><span
```

```
class="box2">我是span2</span><span class="box3">我是span3</span><div class="box4"></div>
</body>
</html>
```

margin设置为 0 auto 就代表距离上方0像素,距离左右两边水平居中

盒子模型

1. 元素的宽度:左边边框+左边内边距+内容宽度+右边内边距+右边边框
2. 元素的高度:上边边框+上边内边距+内容高度+下边的内边距+下边边框
3. 元素空间的宽度:左边的外边距+元素的宽度+右边的外边距
4. 元素空间的高度:上边的外边距+元素的高度+下边的外边距

- 作业

两个div,一个宽高500px,一个200px,让小的在大的正中央

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<style>
div{
    margin: 0 auto;
}
.big{
    width:500px;
    height:500px;
    background: red;
    border: solid 1px black;
    /*无论padding如何改变宽高都不会进行改变*/
    box-sizing:border-box;
    padding-top: 150px;
}
.small{
    width:200px;
    height:200px;
    background: green;
    /* margin-top: 150px; */
}
</style>
<body>
<div class="big">
<div class="small"></div>
```



```
</div>  
</body>  
</html>
```

标准流

1. HTML标签被分为两种类型,一种是块级标签(独占一行,可以设置宽高),一种是行内标签(不独占一行,不能设置宽高)
2. 我们之前学的所有的标签,都是按照标准流进行排列的,块级元素从上到下,行内元素从左到右

display

display属性:inline(行内)block(块级)inline-block(行内块级,能够设置宽高),none(隐藏,不占空间)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style>
```

```
.box1{
    width: 100px;
    height: 100px;
    background: red;
    border: 3px solid black;
}
```

```
span{
    display:inline-block;
    width: 100px;
    height: 100px;
    background: green;
    border: 3px solid black;

}
```

```
.box2{
    display: none;
}
```

```
.box3{

    background: yellow;
}
```

```
</style>
</head>
<body>
</body>
<div class="box1"></div>
<span class="box2"></span><span class="box3"></span>
</html>
```

浮动流

float

浮动流,设置属性**float**能让元素向左或向右进行浮动
元素会脱离标准流,那么标准流的元素会相应顶上来
其次脱离标准流,会在相应的标准流的行号上进行浮动
如果向左或向右进行浮动后,左右两边又有其他元素,
就会紧贴那个元素

浮动流不分行内和块级元素都可以设置宽高

clear

设置**clear**属性,能够使元素在浮动的过程中不去贴靠其他元素,只能影响自己,不能影响其他元素

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style type="text/css">

div{
    width: 100px;
    height: 100px;
}

.box1{
    float:left;
    background: red;
}

.box2{
    float:left;
    background: blue;
    clear: both;
}

</style>
</head>
<body>
<div class="box1"></div>
<div class="box2"></div>
</body>
</html>
```

定位流

相对定位

不会脱离标准流,相对于在标准流的位置进行偏移,所以下面元素不会顶上来

position属性为relative,再结合 top right bottom left 四个属性进行位置的确定

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style type="text/css">
div{
    width: 100px;
    height: 100px;
}
.box1{
    background: red;
}
.box3{
    background: green;
}
.box2{
    position:relative;
    top:20px;
    left:100px;
    background: blue;
}
</style>
</head>
<body>
<div class="box1"></div>
<div class="box2"></div>
<div class="box3"></div>
</body>
```

```
</html>
```

绝对定位

会脱离标准流,所以下面元素不会顶上来,相对于在标准流的位置进行偏移

设置position为absolute,通过top right bottom left定位

如果祖先元素没有定位流(相对定位,绝对定位,固定定位),绝对定位相对于body定位,如果祖先元素是定位流,绝对定位相对于祖先元素(就近原则)

绝对定位不分行内和块级元素都可以设置宽高

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style type="text/css">

div{
    width: 100px;
    height: 100px;
}
.box1{
    background: red;
}
.box3{
    background: green;
}
.box2{

    background: blue;
}
.box4{
    position: absolute;
    top: 0px;
    left: 0px;
    width: 100px;
    height: 100px;
    background: yellow;
}
```



```
</style>
</head>
<body>

<div class="box1"></div>
<div class="box2"></div>
<div class="box3"></div>
<span class="box4"></span>

</body>
</html>
```

注意一般定位的时候注意准则为“子绝父相”

固定定位

固定定位脱离标准流

设置position为fixed,通过top right bottom left定位

不区分行内和块级

和绝对定位相同,唯一不同就是不会随着滚动条滚动而滚动

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style>

div {
    width: 100px;
    height: 100px;

}
.div1 {
    background-color: red;
}
.div2 {
    background-color: green;
    position: fixed;
}
.div3 {
    background-color: blue;
}
.div4 {
    background-color: yellow;
    width: 100px;
    height: 20000px;
}
</style>
</head>
```

```
<body>

<div class="div1"></div>
<div class="div2"></div>
<div class="div3"></div>
<div class="div4"></div>
</body>
</html>
```

zIndex

z-index 改变定位流中的前后顺序

- 定位流中有一下规律

默认情况下定位流会覆盖标准流

默认情况下后来的会覆盖之前的

定位流中设置z-index,谁的大谁就在前

如果父元素设置z-index,子元素的z-index就失效,谁的父元素大,谁就会显示在上方(从父原则)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>

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

```
div {
    width: 100px;
    height: 100px;

}

.div1 {
    background-color: red;
    position: relative;
}

.div2 {
    background-color: green;
    position: fixed;
    z-index: 1;
}

.div3 {
    background-color: blue;
    position: absolute;

}

.div4 {
```

```
        background-color: yellow;
        width: 100px;
        height: 20000px;
    }
</style>
</head>
<body>
<div class="div1"></div>
<div class="div2"></div>
<div class="div3"></div>
<div class="div4"></div>
</body>
</html>
```