

Day02_CSS

今日目标

1. 了解div+css
2. 了解css
3. css的书写规范
4. css的引入方式
5. css的选择器
 1. 基本选择器
 2. 伪类选择器
6. 样式
 1. 字体样式
 2. 文本样式
 3. 边框样式
 4. 背景样式
 5. 盒子模型
 6. 布局
 7. 定位
 8. 标签类型转换

1.了解div+css

div: 区域块, 没有宽度和高度的, 也没有边框, 要想该区域块可以使用, 必须依靠**css**来添加, 在指定的页面位置设置相关的样式和结构, 做页面布局

2.css简介

css:层叠样式表,给页面进行样式渲染。除了加样式之外可以对页面的标签进行像素级的布局微调

3.css的书写规范

```
选择器{  
    样式名:值;  
    样式名:值;  
}
```

从以上的结构可以看出，**css**就是包含 选择器+样式

4.css的引入方式

1. 行内样式

在标签内部添加**style**属性，在该属性的值中添加样式名与值
格式如下：

```
<div style="样式名:值;样式名:值;"></div>  
<div style="border: 1px solid red;width:  
200px;height: 200px;"></div>
```

缺点：该样式引入只能在当前标签使用，作用范围比较小

2. 内联样式

1. 在页面添加<style></style>标签(在<head></head>标签内部)

2. 在该标签内部按照**css**的书写规范添加样式

3. 外联样式

1. 创建一个**.css**文件，在该文件中以**css**的编写规范添加样式

2. 在html页面通过<link/>标签来引用该css文件

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

href: 引入css文件的路径

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <style type="text/css">
      #d1{
        border: 3px solid orange;
        height: 250px;
        width: 250px;
      }
    </style>
  </head>
  <body>
    <div style="border: 1px solid red;width: 200px;height: 200px;">1</div>
    <div id="d1">2</div>
  </body>
</html>

```

内联样式

```

<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <link rel="stylesheet" type="text/css" href="css/demo.css"/>
    <style type="text/css">
      /* div{
        border: 3px solid orange;
        height: 250px;
        width: 250px;
      } */
    </style>
  </head>
  <body>
    <div style="border: 1px solid red;width: 200px;height: 200px;">1</div>
    <div>2</div>
    <div>3</div>
  </body>
</html>

```

外联样式

当三种引入方式同时对同一个标签添加相同样式那么就会引发样式冲突问题，如何解决？

引入方式优先级：

遵循就近原则

行内样式 > 内联样式 | 外联样式

5.css选择器

选择器：定位到要添加样式的标签

1.基本选择器

id选择器

以#开头，以标签中id的属性值作为选择器的名称

```
<div id="d1"></div>    #d1{}
```

类(class)选择器

以.开头，以标签中class属性值作为选择器的名称

```
<div class="c1"></div>    .c1{}
```

标签选择器

以标签名作为选择器的名称

```
<div></div>    div{}
```

通配选择器

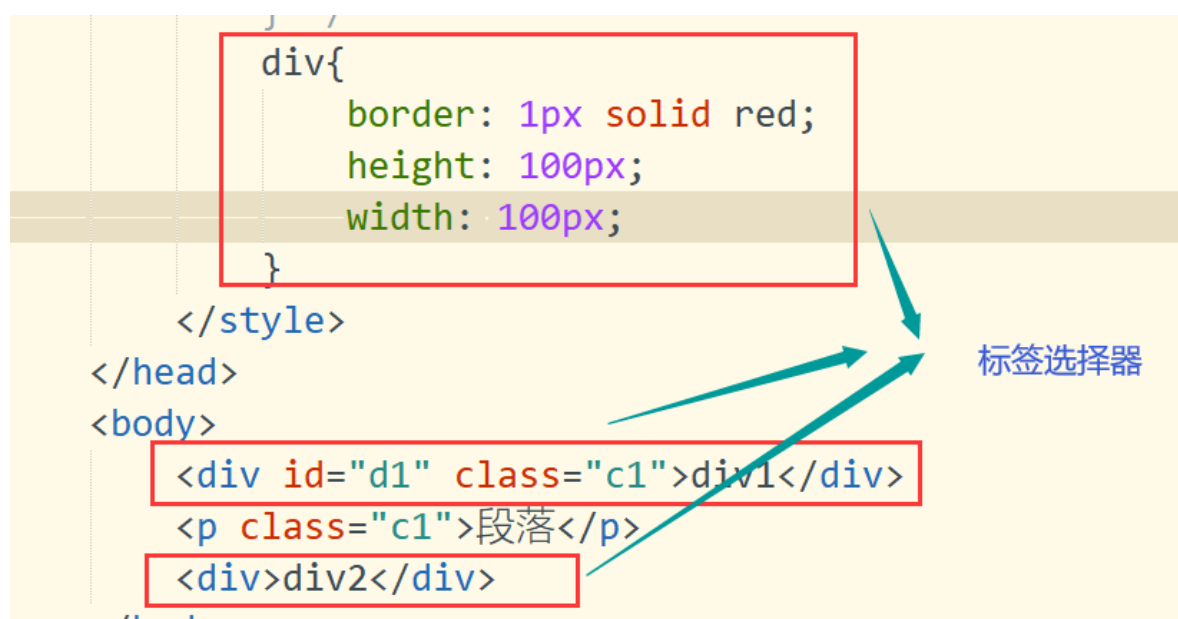
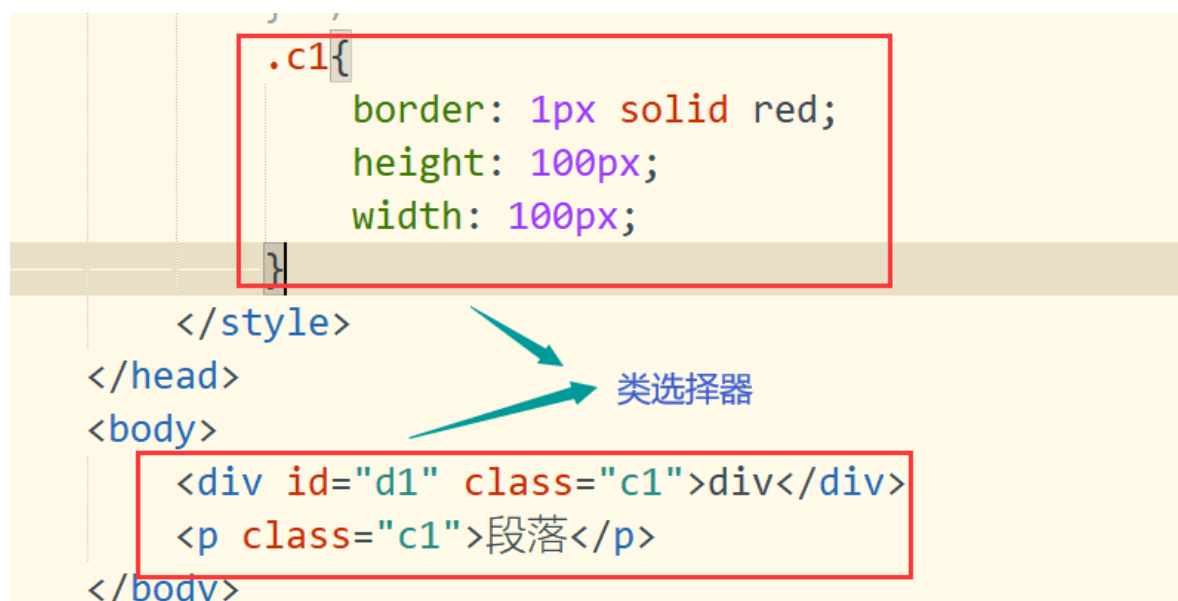
* 匹配所有标签，往往用来做去除浏览器自带样式

混合选择器

由多个选择器组成，每个选择器之间以逗号隔开

```
<head>
  <meta charset="utf-8">
  <title></title>
  <style type="text/css">
    #d1{
      border: 1px solid red;
      height: 100px;
      width: 100px;
    }
  </style>
</head>
<body>
  <div id="d1"></div>
```

id选择器



2. 伪类选择器

主要应用在超链接中

:link 未访问的超链接

:hover 鼠标悬停超链接上

:visited 已经访问过的超链接

:active 激活状态(点击不要放开状态)的超链接

如果以上四个状态都存在，那么按照以上顺序来使用

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title></title>
```

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

```
      a:link{ /*匹配没有访问过的超链接*/
```

```
        color: #FF0000;
```

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

```
    }
```

```
      a:hover{ /*鼠标放在超链接上*/
```

```
        color: blueviolet;
```

```
        font-size: 40px;
```

```
    }
```

```
      a:visited{ /*匹配访问过的超链接*/
```

```
        color: #008000;
```

```
        font-size: 15px;
```

```
    }
```

```
      a:active{ /*激活状态的超链接*/
```

```
        color: orange;
```

```
        background-color: burlywood;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <a href="#">超链接</a>
```

```
</body>
```

```
</html>
```

[超链接](#)

未访问的超链接

[超链接](#)

悬停的超链接

[超链接](#)

访问过的超链接

[超链接](#)

激活状态的超链接

6. 样式

1. 字体样式

1. `font-size` 字体大小 默认16px

2. `font-family` 字体系列 比如宋体，新宋体，楷体等

3. `font-style` 字体风格 正常字体(`normal`), 斜体(`italic`)

4. `font-weight` 字体粗细 取值 100-900的整百数
`bold`(700的值)

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title></title>
```

```
<style type="text/css">
    span{
        font-size: 25px;
        font-family: "楷体";
        font-style: italic;
        font-weight: bold;
    }
</style>
</head>
<body>
    <span>
        大家都好帅
    </span>
</body>
</html>
```

大家都好帅

2.文本样式

1. **text-indent** 文本缩进 单位px
 2. **text-align** 文本对齐方式,该文本必须在一个块标签中才可以生效。
 3. **text-decoration** 文本装饰 **none**(没有任何装饰)|**underline**(下划线)|**overline**(上划线)|**line-through**(删除线)
 4. **line-height** 行高,一般来做垂直方向的居中处理,将行高和外部的标签高度保持一致
- 块标签和行内标签:
- 块标签: 默认占满一行,并且自动换行,具有宽度和高度样式
div,p,h1, h2... h6,ol,ul,li...

行内标签：在一行中使用就占用多少，不会自动换行，除非该行已经写满，没有高度和宽度样式

a, span, font, textarea, select...

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title></title>
```

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

```
      p{
```

```
        text-indent: 80px;
```

```
      }
```

```
      div{
```

```
        width: 50%;
```

```
        height: 300px;
```

```
        border: 1px solid red;
```

```
        text-align: center; /*文本水平居中
```

```
*/
```

```
        text-decoration: line-through;
```

```
        line-height: 300px; /*垂直方向的居
```

```
中*/
```

```
      }
```

```
      a{
```

```
        text-decoration: none;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <p>前不久，江苏小吴网购了一支名牌口红作为女友生日礼物，发现是假货后报案</p>
```

```
    <div>一支名牌口红作为女友生日礼物</div>
```

```
    <a href="">超链接</a>
```

```
  </body>
```

```
</html>
```

3.边框样式

1. **border-width** 边框宽度
2. **border-color** 边框颜色
3. **border-style** 边框的样式
4. **border-radius** 边框的弧度

最多有四个值表示每个角的弧度

顺序是 上左 上右 下右 下左

如果是2个值 （上左,下右）（上右,下左）

如果是一个值：四个角的值都是这个值

如果每一个边角的弧度是该标签高度的50%,那么不是内切圆就,内切椭圆

top,bottom,left,right

border-(top|bottom|left|right)-width

border-(top|bottom|left|right)-color

border-(top|bottom|left|right)-style

简化写法:

border: 宽度 样式 颜色

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title></title>
```

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

```
      div{
```

```
        width: 300px;
```

```
        height: 300px;
```

```
        border-width: 4px;
```

```
        border-color: red;
```

```
        border-style: solid;
```

```
        /*右边框为虚线*/
```

```
        border-right-style: dotted;
        border-bottom-color: blue;
        border-left-width: 1px;
        border-radius: ;
    }
    p{
        border: 3px solid red;
        width: 300px;
        height: 500px;
        border-radius: 50%;
    }
    #d1{
        border: 1px solid red;
        width: 80px;
        height: 30px;
        border-radius: 10px;
        background-color: orange;
        text-align: center;
        line-height: 30px;
    }
</style>
</head>
<body>
    <div>1</div>
    <p></p>
    <div id="d1">登录</div>
</body>
</html>
```

4.背景样式

1. background-color: 背景颜色

2. **background-image:url()** 背景图片, **url** 中路径不需要加引号

3. **background-position**: 背景定位, 偏移, 图片以什么位置开始显示

left, top, right, center, bottom, 由这些值中的两个值来确定背景图片的显示位置

如果只有一个值的时候, 另外一个值默认取 **center**

4. **background-repeat**: 背景平铺, 默认平铺 (**no-repeat | repeat-x | repeat-y**)

5. **background-size(css3)** 可以是具体值, 也可以百分比 宽度 高度, 如果设置一个值, 另外的高度值会进行等比例缩放

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<style type="text/css">

body{

background-color: #FFA500;

background-image:

url(img/u=2801979887,881698885&fm=26&gp=0.jpg);

/* background-repeat: repeat-y;

***/**

/* background-position: top; */

background-size: 100% 800px;

}

</style>

</head>

<body>

</body>

</html>

5. 盒子模型

将页面中所有的标签都看成一个盒子,目的是为了更好实现页面的布局,通过快捷键**f12**

1. 内边距 **padding** 内容和边框的距离 , 有4个值, 4个值分别是 上右下左

2. 外间距(补白) **margin** 边框与边框的距离 有4个值, 4个值分别是 上右下左

3. 内容

padding-(left|right|top|bottom) 设置单个方向上的内边距

margin-(left|right|top|bottom) 设置单个方向上的外边距

如何计算一个完整盒子的宽度和高度?

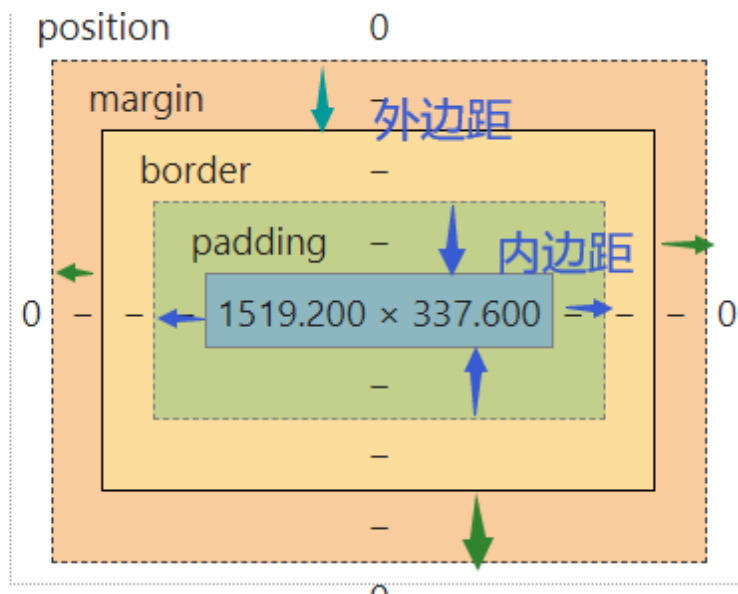
我们在**css**中设置的标签的宽度和高度其实是设置的是盒子内容的宽度和高度

宽度=左边框+左内边距+内容宽度+右内边距+右边框;

高度=上边框+上内边距+内容的高度+下内边距+下边框;

##注意:当纵向产生冲突的时候取最大值

##横向产生冲突时取两者之后



6.布局

浮动:漂浮流动,脱离标准文档流,会形成一个新的流(上层流)实现左右的偏移

##注意:产生浮动的标签一定不会覆盖文字

文档流:在html页面中正常标签顺序显示的页面

float: left|right

clear:消除浮动对标签产生的影响 left|right|both

7.定位

position: relative|absolute|fixed

relative:相对定位

相对原来的位置发生偏移。定位后不会脱离标准文档流

absolute:绝对定位,如果父标签中有定位

(relative|absolute|fixed),那么相对父标签偏移,如果父标签中没有定位,那么相对body进行偏移,会脱离标准文档流

fixed:固定定位,将一个标签固定在某一个地方,不会随着滚动条的移动而移动,并且也会脱离标准文档流

left|right|top|bottom这些值实现定位之后的偏移

```

<meta charset="utf-8">
<title></title>
<style type="text/css">
    div{
        border: 1px solid red;
        height: 200px;
        width: 30%;
    }
    #d1{
        position: relative;
        left: 50px;
    }
    #d2{
        position: absolute;
    }
    #d22{
        height: 50px;
        width: 50px;
        background-color: red;
        position: absolute;
        right: 0;
        bottom: 0;
    }
</style>
</head>
<body>
    <div id="d1">1</div>
    <div id="d2">2
        <div id="d22">2.1</div>
    </div>
    <div id="d3">3</div>
</body>
</html>

```

相对定位

绝对定位

```

<html>
<head>
    <meta charset="utf-8">
    <title></title>
    <style type="text/css">
        body{
            background-image: url(img/u=2494280858,2319139394&fm=26&gp=0.jpg);
        }
        div{
            border: 1px solid red;
            height: 400px;
            width: 300px;
            position: fixed;
            right: 0;
            top: 100px;
        }
    </style>
</head>
<body>
    <br><br><br><br><br>
    <div></div>
    <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
    <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
    <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
    
</body>
</html>

```

固定定位