第三天笔记

复习

CSS的引入方式

- 行内式 ：通过标签属性style引入css样式，直接将css代码写在开始标签内

```
<tagName style="属性名:属性值;属性名:属性值;"></tagName>
```

- 内嵌式 ： 将css代码写在`<style></style>`元素内，通常来说style元素要放在head元素内

```
<head>
<style>
/*css的基本语法*/
选择器{声明}
/*声明由属性名和属性值组成，一个选择器内可以有多条声明*/
</style>
</head>
```

- 外联式 ：html文件和css文件是独立的文件，通过link标签将css文件引入到html文件中

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

- 导入式 ：html文件和css文件是独立的文件，通过css语句@import “url”引入一个css文件，这条语句有特定的规则：
  - 这条语句需要写在样式表中
  - 这条语句必须放在所有样式之前

```
<style>
@import "url";
p{}
</style>
```

CSS的选择器

基本选择器

- 标签选择器：直接将标签名当做选择器来使用
  权重为1
  `tagName{}`
- 类选择器：将标签属性class的属性值当做选择器使用，在使用的时候需要在属性值前面加一个”.”
  权重为10
  一个元素的类名可以有多个，每一个类名之间用空格隔开 `<p class="act1 act2 act3"></p>`
  `.className{}`
- id选择器 ：将标签属性id的属性值当做选择器使用，在使用的时候需要在属性值前面加一个”#”
  权重为100
  id的属性值在当前文档下只能有一个
  `.idName{}`
- 通配符选择器 ：选择当前文档下所有的元素
  权重小于1
  `*{}`
- 标签属性选择器 ：是利用标签属性的属性名和属性值来进行元素的选择
  权重为10
  `[属性名]{}` `[属性名=属性值]{}`

其他选择器

- 后代选择器（派生选择器）：父辈选择器用来划定范围，子级选择器用来添加样式，后代选择器不用将每一层级的元素都写出来，只需写出关键节点即可
  后代选择器的个数尽量不超过**三个**
  权重是所有选择器权重之和

```
<style>
div li{color:red;}
div ul li{}
ul li{}
</style>
<div>
div
    <ul>
        <li>li</li>
    <ul>
</div>
<nav>
    <ol>
        <li></li>
    <ol>
</nav>
```

- 子级选择器：父级选择器>子级选择器（父子元素必须是相邻关系）

权重是所有选择器权重之和

```
<style>
div>ul>li{}
</style>
<div>
    <ul>
        <li></li>
    </ul>
</div>
```

- 相邻兄弟选择器：上一个选择器+下一个相邻选择器
  权重是所有选择器权重之和

```
.box+span{}
<div class="box"></div>
<span></span>
```

- 并集选择器（分组选择器）：可以使不同的元素使用同一段css样式，每一个选择器用逗号隔开
  权重：每一个逗号隔开的选择器都是独立的选择器，所以权重都是独立计算

```
<style>
.box li,.list>.link,#tab{}
   11         20       100
</style>
```

- 交集选择器：两种属性出现在同一个元素上的时候，可以使用这两种属性准确的找到要操作的元素
  将同一元素的不同属性结合到一起就是交集选择器，交集选择器链接什么符号都不需要，直接写选择器即可
  权重是所有选择器权重之和
  ps:标签选择器和其他选择器组合的时候，标签选择器要写在前面

```
<style>
dl{}
.list1{font-size:100px;}
.list2{color:red;}
#lists{}
ul.list2{color:green;}
.list1.list2{color:green;}
input[type=text]{}
</style>
<dl class="list1 list2" id="lists">
</dl>
<dl class="list1"></dl>
<ul class="list1 list2"></ul>
<input type="text" />
<input type="button" />
```

- 伪类：给某一个元素添加状态
  基本语法：选择器:伪类{}
  权重：10<权重<11

```
div:hover{}
.box:hover{}
```

- 伪元素：通过css向html结构中，输出假元素

```
<style>
/*选择器和伪元素之间是父子关系*/
选择器:after,选择器:before{
display:block;
content:"可以有内容也可以为空";
}
p:after,p:before{}
</style>
<p>
<!--我是before-->
我是内容
<!--这里是after-->
</p>
```

盒子模型

组成网页的每一个元素，在页面上的表现形式都是一个矩形（所占的位置是一个矩形），这一个个的矩形都叫做盒子（因为每一个矩形中都会装有内容）

盒子模型是用来描述每一个元素之间的位置关系，和盒子本身的大小属性

元素的css属性分为两种：

- 内置属性：元素天生自带的css属性

- 自定义属性：人为添加给元素的css属性，自定义属性的属性权重大于内置属性

盒子模型的组成

盒子模型由外边距，边框，内边距和内容部分组成，其中外边距，边框，内边距都是四个方向，内容只有宽高每一个元素都有自己的内置样式，在编辑页面之前需要将这些内置样式同一重置，保证页面样式的正常展示

- 外边距：盒子与盒子之间的距离，用css属性margin表示，margin有四个方向，分别是
  - margin-top：上外边距
  - margin-right：右外边距
  - margin-bottom：下外边距
  - margin-left：左外边距
    margin值可以是数字、百分比、em、rem，并且margin值支持负值
    *margin*的缩写：
  - margin:10px 20px 30px 40px; 上右下左
  - margin:10px 20px 30px;上 左右 下
  - margin:10px 20px; 上下 左右
  - margin:10px;四个方向数值一致

外边距的兼容问题：

- 当上面的盒子有下外边距，下面的盒子有上外边距的

时候，两个值只有最大值起作用

- 子级盒子的上边距会传递给父级盒子（指的是有包含关系的所有盒子），当父级盒子没有内边距或边框的时候，就会出现这个问题

    - 解决方式
        - 给父级元素添加overflow:hidden;(溢出隐藏)属性
        - 给父级元素添加内边距或边框属性
        - 将子元素的margin-top值变为父级元素的padding-top值 【最优选项】

- 边框：在内边距和外边距之间，包裹盒子的四个防线的线条

    - 边框有四个方向 上 右 下 左
    - 边框可以设置颜色
    - 边框可以设置不同的样式

属性	**描述**
border-top-width	上边框的宽度
border-top-color	上边框的颜色
border-top-style	上边框的样式
border-top:1px solid red	上边框宽度样式颜色

border-width:10px 20px 30px 40px;	上右下左四个方向的边框宽度
border-color:red yellow green pink;	上右下左四个方向的边框颜色
border-width:10px 20px 30px;	上 左右 下的边框宽度
border-width:10px 20px;	上下 左右的边框宽度
border-width:10px;	四个方向的边框宽度
border:10px solid red;	四个方向的宽度 颜色 样式都一致

`transparent`透明

> 利用边框制作三角形

```
1.  div{
2.     border-width:30px;
3.     border-color: red transparent tra
   nsparent;
4.     border-style:solid;
5.     width: 0;
6.     height: 0;
7.         }
```

- 内边距：边框距离盒子内容之间的距离
    - padding有四个方向，分别是
    - padding-top：上内边距
    - padding-right：右内边距
    - padding-bottom：下内边距
    - padding-left：左内边距
      padding值可以是数字、百分比、em、rem
      *padding*的缩写：
    - padding:10px 20px 30px 40px; 上右下左
    - padding:10px 20px 30px;上 左右 下
    - padding:10px 20px; 上下 左右
    - padding:10px;四个方向数值一致
- 内容（content）分别有两个属性来代表
    - 宽度 width 数字 百分比 rem em
    - 高度 height 数字 百分比 rem em
- 如何计算盒子的大小
    - 盒子的宽度=内容的宽度+左右内边距+左右边框宽度
    - boxWidth=width+(padding-left)+(padding-right)+(border-left-width)+(border-right-width)
    - 盒子的高度=内容的高度+上下内边距+上下边框宽度
    - boxHeight=height+(padding-top)+(padding-

bottom)+(border-top-width)+(border-bottom-width)

```
<style>
.box{width:100px;padding:10px 20px 30px;border-width:20px 50px;border-style:solid;margin-bottom:10px;}
<style>
boxWidth=100+20+20+50+50
boxHeight=0+10+30+20+20
```

BFC 块级盒子渲染模式

> 块级盒子应该如何在页面中排布
>
> - 盒子从父级盒子的左上开始
> - 垂直排布
> - 如果遇到上下margin值的时候，会发生margin值的折叠
> - 所有BFC渲染的盒子都是基本流（文档流）内的盒子
>
> block formatting context

IFC 行内盒子渲染模式

> 行内盒子应该如何在页面中排布

inline formatting context

- 盒子从父级盒子的左上开始
- 横向排布，达到父级宽度的最大值，自动折行
- padding-top，padding-bottom，margin-top，margin-bottom，可以识别但是不起作用

块级元素的特点

- 独占一行，垂直排列
- 可以设置盒子模型的所有属性(width、height、padding、border、margin)
- 在不设置盒子宽高的时候，宽度是其父级盒子内容的宽度，高度是其本身内容的高度
- 块级元素可以嵌套其他元素
  - ul 、ol里面第一层级只能是li
  - dl第一层级只能是dt和dd
  - p不能嵌套块级元素

行内元素的特点

- 在一行显示，从左到右，达到父级元素的最大宽度的时候，自动折行

- 不能设置宽度和高度，padding和margin的上下值
- 宽度和高度是由本身内容的宽高决定的
- 行内元素进行嵌套的时候，不要嵌套块级元素
- 行内元素之间如果有空格或者回车，在页面上显示的时候就会出现盒子与盒子之间的空隙

overflow

属性值	描述
hidden	隐藏[超出盒子的部分不显示]
auto	自动识别[哪个方向内容溢出，那么哪个方向就会出现滚动条]
scroll	出现滚动条[无论是否溢出，都会出现滚动条]
visible	默认状态
inherit	从父元素的属性上继承属性值

display 以什么样的方式显示元素

属性值	描述

block	块级
inline	行内
inline-block	行内块级元素
none	没有
list-item	列表项目[列表队列]
table	表格
table-cell	表格项

```
ul{display: table;}
li{display: table-cell;}
```

inline-block 行内块级元素

- 从父级盒子的左上角开始排列，从左到右排列，遇到父级元素最大宽度的时候，自动折行
- 可以直接设置盒子模型的所有属性
- 元素之间如果有空格或者回车，在页面上显示的时候就会出现盒子与盒子之间的空隙
- 同级元素之间默认垂直方向对齐方式为基线对齐

基线对齐：同级元素之间的垂直对齐方式

vertical-align: baseline;

属性值	描述
baseline	基线对齐
top	顶部对齐[兄弟元之间最高的那个元素的顶部对齐]
bottom	底部对齐[兄弟元之间最高的那个元素的底部对齐]
middle	中部对齐[兄弟元之间最高的那个元素的中部对齐]

> 如果想要改变基线对齐方式，那么需要给每一个改变的元素都添加上这个属性

水平对齐

- 文字的水平对齐
    - 左对齐 text-align:left;
    - 居中对齐 text-align:center;
    - 右对齐 text-align:right;

    > 这个属性需要给有宽度的元素上

- 盒子的水平对齐

- 行内或行内块级元素的对齐方式
  - 左对齐 text-align:left;
  - 居中对齐 text-align:center;
  - 右对齐 text-align:right;

> 这个属性需要给有宽度的元素上

- 块级元素的对齐方式
  - 给这个块级元素宽度，并且margin:0 auto; 左右值为auto即可，那么这个块级元素就会在其父级元素内水平居中