

## 实验 04 《CSS 与 DIV 页面布局》

### 实验目的：

熟悉 CSS 盒子模型的大小、内外边距、边框及浮动、定位等概念，掌握 DIV 标记的 CSS 属性，能通过 DIV、P、IMG 等标记及 CSS 属性进行基本的页面布局和修饰

### 实验课时：

课内：2 课时；课外：2 课时

### 实验内容：

#### 操作准备

1. 以下操作步骤中所涉及的 198009010001、文立斌、wlb 均应分别替换成你的完整学号、姓名、姓名的拼音缩写
2. 在 D: 或 E: 分区创建一个以你的“完整学号+姓名”命名的文件夹（名称应类似：198009010001 文立斌），我们把这个文件夹简称为“你的文件夹”
3. 将“素材”文件夹复制到“你的文件夹”内，并将其名称改为类似“imageswlb”

#### 任务 A、认识盒子模型

本任务中，我们通过制作以下效果图所示网页，来尝试理解和掌握盒子模型的基本属性：



具体步骤如下：

1. 创建一个名称类似“文立斌 A.html”的文档，将网页的标题设置为类似“文立斌 A”，

然后将该文档保存在“**你的文件夹**”内

2. 在主体部分用 p 标签创建两个段落：

- ✓ 第 1 个段落包含以下文字：

网页可看成由一个个盒子（板块）构成。要随心所欲地控制页面中每个盒子的样式，需掌握盒子模型的相关属性：背景颜色、背景图片、边框、外边距、内边距、宽度、高度、位置及定位方式、浮动方式等。

- ✓ 第 2 个段落中放置 6 张图片：图像文件可从“校花 1.jpg~校花 8.jpg”及“校草 1.jpg~校草 8.jpg”中任选其六

现在浏览一下页面，如下图所示（90%大小效果），由于没有任何布局和修饰，页面只是由上而下在两个段落中分列出文字和图片，段落上下左右也是采用 p、img 标记默认的边距。当然，浏览窗口的大小会使得您看到的页面效果有所不同：



3. 现在，在网页头部定义 2 个选择器，选择器的内容及要求如下：

- ✓ **通配符选择器**：清除 HTML 为所有页面元素提供的默认内、外边距

- ✓ **主体标记选择器**：将背景颜色设置为浅红色或其他，了解主体部分所占据的空间

浏览页面，如下图所示（90%大小效果），两个段落四周及其中间已没有默认的内、外边距，粉红色背景部分即页面主体（body）全部空间：



4. 按如下属性设置要求，在头部定义名称类似 **w1bp** 的一个类选择器，并在主体部分上面的、第一个文字段落中引用该选择器，以设置该段落的样式：

属性名	属性设置要求
背景颜色	黄色或其他醒目颜色
高度	66 像素
外边距	10 像素
内边距	5 像素
边框	红色、实线、5 像素宽
边框圆角半径	20 像素
背景裁剪方式 (background-clip)	content-box (内容盒子)
溢出处理方式 (overflow)	自动

再次浏览页面，对照如下页面效果图（90%大小效果），深刻理解以上选择器中所设置的属性对页面中第一个段落样式的影响：



- ✓ 文字段落上下左右都加了 10 像素的外边距
- ✓ 内边距、边框宽度都是 5 像素，20 像素的边框圆角半径也在效果图中得到了体现
- ✓ 已经将“溢出处理方式 (overflow)”设置为“自动”对不？那么为什么没有看到效果？这是因为浏览窗口够宽，刚好能显示下所有文字，没有导致溢出，如此而已！要看到溢出效果（自动添加纵向滚动条），您可以将浏览窗口缩窄一些，或在段落中加多一些文字
- ✓ 值得注意的是“背景裁剪方式 (background-clip)”这个属性，我们将盒子的背景色设置成了黄色，但段落中仅仅只是文字内容所在部分填充了黄色，这是因为 background-clip 属性的取值已设置为 content-box 及其如下效用：

属性值	描述
<b>content-box</b>	背景被裁剪到仅包含内容部分（仅文字内容部分填充背景）
<b>padding-box</b>	背景被裁剪到包含内边距的部分（内边距、内容均填充背景）
<b>border-box</b>	背景被裁剪到边框部分（边框、内边距、内容均填充背景）

当然，您似乎应该尝试一下使用另外两个属性值的效果如何，对不？

5. 按如下属性设置要求，继续定义名称类似 **w1bm** 的一个类选择器，并在主体部分下面图像所在段落中引用该选择器，以设置该段落的样式：



属性名	属性设置要求
背景颜色	白色
外边距	20 像素
内边距	5 像素
边框样式	虚线
边框宽度	上下：10 像素，左右：5 像素
边框颜色	上：绿色，下：柠檬色（lime） 左： rgb(255,0,102)，右： 蓝色
文本对齐方式	居中
定位方式	相对定位

浏览页面，对照如下页面浏览效果图（100%大小效果），深刻理解以上选择器中所设置的属性对页面元素样式的影响：



- ✓ 首先，这会儿浏览窗口缩窄了一点，您看到上一个段落中的文字产生溢出，并自动加上了纵向滚动条对不对？
- ✓ 由于没有设置“背景裁剪方式（background-clip）”属性，图像所在段落的背景按默认值 border-box（边框盒子）被裁剪到包含边框部分在内，即边框、内边距、内容部分均填充了白色背景（当然，虚线本身颜色部分肯定不会被填充到）
- ✓ 图像段落上下左右都加了 20 像素的外边距，注意：由于相邻两个盒子之间的间隔

只是取其中最大的外边距值，而上一个段落的外边距是 10 像素，下一个段落的外边距是 20 像素，因此上下段落之间也就被设置为了 20 像素的间隔

- ✓ 上下、左右边框的宽度被分别设置成了 10 像素、5 像素，颜色也各各不同
- ✓ 文本居中对齐的设置导致该段落中的图像被横向居中对齐（看来这个属性不应该叫 `text-align`，而应该叫 `content-align`，您说呢？），但纵向的话都是底部对齐，要使得图像纵向顶部对齐，需要使用 `vertical-align` 属性
- ✓ 我们将定位方式设置为了“相对”，但没有看到效果对不对？这是因为我们没有设置其 `left`、`right`、`top` 和 `bottom` 属性，**在此如此设置的目的是在后！**

6. 定义名称类似 `wlby` 的类选择器，将边框设置为单像素、宽度和高度分别设置为 100 像素、120 像素，再定义名称类似 `wlby` 的类选择器，将边框设置为单像素、宽度和高度分别设置为 50 像素、60 像素，然后参照以下截图在图像标记中引用这两个选择器：



现在，我们将使用标签的 `style` 属性直接定义 CSS 样式，通过这些样式的定义，来了解 CSS 图层主要的定位方式：

7. 将左起第 3 幅图像定义为：顶部对齐，相对定位方式，左边距 20 像素，顶边距 20 像素；如下图所示，您应注意到，这幅图像由默认的底部对齐变为了顶部对齐，但由于采用了相对定位方式，而且设置了左边距、顶边距各 20 像素，因此图像在顶部对齐的基础上，又向右、向下都漂移了 20 个像素——





8. 将左起第 4 幅图像定义为：绝对定位方式，左边距 5 像素，顶边距 5 像素；如下图所示，该图像跑到其所在段落的左上角去，之所以如此，是因为其由 p 标签定义的父级元素的定位方式为相对（参见第 5 步红字标注），因此在这个父元素内进行绝对定位（距 p 元素左边、顶边的距离都是 5 像素），而且，由于绝对定位方式不保留原文档流位置，因此右边的第 5 幅图像向左边移动，取代了第 4 幅图像原来的位置



9. 将原来左起第 5 幅图像定义为：绝对定位方式，右边距 5 像素，底边距 5 像素；如下图所示，该图像跑到其所在段落的右下角，道理当然跟第 8 步中描述的一样



10. 将原来左起第 6 幅图像定义为：固定定位方式，左、顶边距分别是 85、20 像素，透明度为 0.5；现在，如以下最终浏览效果所示，固定定位方式总是相对页面浏览窗口来进行定位的，无论页面如何滚动，该图像始终保持距页面左边 85 像素、页面顶端 20 像素！



11. 至此，希望您基本懂得了盒子模型的主要属性。当然，记得在页面最后添加作者段落！
12. 保存网页，在浏览器中浏览、测试，撰写实验报告

### 任务B、认识布局（1）

从布局结构看，任何网页都可以看成由一个个盒子（box）构成。本任务将结合 div 和 css 属性处理盒子的相邻关系，进行简单的三行两列式页面布局，完成后的页面类似：



具体要求及步骤如下：

1. 创建一个名称类似“**文立斌 B.html**”的文档，将网页的标题设置为类似“**文立斌 B**”，然后将该文档保存在“**你的文件夹**”内
2. 页面布局结构设计

我们分析一下，要实现以上页面的布局设计，关键你得懂得盒子模型的特点：默认情况下，盒子是遵循从上到下、从左到右的规则布局的。

- ✓ 根据以上规则及设计要求，这个页面首先是上、中、下三个大的盒子构成，即呈现出这样的布局结构：

```
<div>页眉</div>
```

```
<div>主体</div>
```

```
<div>页脚</div>
```

- ✓ 然而，关键的是，中间的大盒子中还套着左、右两个小盒子，这要怎么处理？是否可以考虑，简单地将中间的盒子一分为二，然后通过浮动（float）属性的设置将左右两个盒子并排显示？也就是说采用底下这样的布局结构？暂不考虑浮动：

```
<div>页眉</div>
```

```
<div>主体左侧</div><div>主体右侧</div>
```

```
<div>页脚</div>
```

这样的布局不是说不可以，但要实现更精准的控制，更好的做法是，将主体部分左

右两个盒子放在另一个盒子里，形成这样的页面布局结果：

```
<div>页眉</div>
```

```
<div>
```

```
    <div>主体左侧</div><div>主体右侧</div>
```

```
</div>
```

```
<div>页脚</div>
```

- ✓ 更进一步的要求，我们希望这些盒子做一个整体上的控制，比如像上图那样，这些盒子总的宽度为页面宽度的 80% 且居中显示在页面上，为了达到类似这样的目的，应该考虑在这些盒子的外面再加多一个盒子，即最终页面的布局结构为：

```
<div>
```

```
    <div>页眉</div>
```

```
    <div>
```

```
        <div>主体左侧</div><div>主体右侧</div>
```

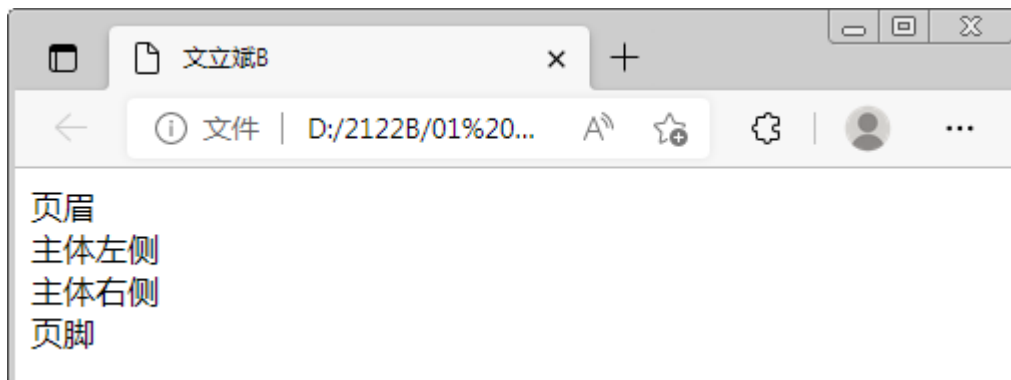
```
    </div>
```

```
    <div>页脚</div>
```

```
</div>
```

注意：这个页面结构中，蓝色、红色标示的 div 并不包含实质内容，而只是作为一个容器，对其中的其他 div 元素起控制作用

现在，请按以上结构进行页面布局，为后面的操作方便起见，请将这些 div 的 id 依次设置为类似：**wlbcontainer**、**wlbheader**、**wlbmain**、**wlbleft**、**wlbright**、**wlbfooter**。尝试浏览，效果真不咋地——



显然，还得通过选择器来进一步控制盒子的样式，以形成符合要求的页面结构

### 3. 通过选择器定义样式

定义选择器前，我们先了解一个**非常重要的属性**——**box-sizing**，这个属性的值用于设置盒子大小的计算方式，在处理盒子大小时起着决定性的作用：

- ✓ **content-box (内容盒子)**：这个属性值意味着在给盒子指定的宽度和高度外，再去添加内边距、绘制边框，也就是内边距、边框宽度不计算在 width、height 指定的宽度、高度范围内
- ✓ **border-box (边框盒子)**：在给盒子指定的宽度和高度范围内绘制盒子，包括内边距、边框宽度及盒子的实际内容。换句话说，给盒子指定的宽度、高度减去边框宽度和内边距，才是盒子内容的实际宽度、高度
- ✓ **inherit (继承)**：用于告诉浏览器，box-sizing 属性的值从父元素继承而来，即取决于外层盒子的设置

现在，请参照前面的效果图，用 **style 标签** 创建一个通配符选择器并为页面中的六个 div 盒子各创建一个 id 选择器，通过这些选择器来定义各 div 的样式：



- ✓ **通配符选择器**：清除所有默认的内、外边距，设置适当的字号及文本对齐方式，当然，既然叫“通配符选择器”，这些设置自然将影响到页面内所有元素
  - ✓ **容器（wlbcontainer）选择器**：将宽度设置为 80%，外边距设置为适当值，以确保整个容器盒子上下边距适当且盒子整体水平居中对齐
  - ✓ **头部（wlbheader）选择器**：适当设置盒子的背景色、内边距、边框及盒子高度
  - ✓ **主体（wlbmain）选择器**：适当设置盒子的背景色、宽度（100%）、高度和盒子大小计算方式（box-sizing）（这是关键！）
  - ✓ **主体左边栏（wblleft）选择器**：适当设置盒子的内边距、边框、宽度（30%），高度、浮动方式（float）和盒子大小计算方式（box-sizing）（后两个是关键！）
  - ✓ **主体右边栏（wlbright）选择器**：适当设置盒子的内边距、边框、宽度（70%），高度、浮动方式和盒子大小计算方式（后两个当然还是关键！）
  - ✓ **脚注（wlbfooter）选择器**：适当设置盒子的背景色、内边距、边框、高度，并清除浮动方式，使之不受给前面的盒子设置的浮动方式的影响，回到标准文档流——不过，由于中间的左右两个盒子被放在了 wlbmain 内，而 wlbmain 并未设置任何浮动方式，因此这里其实可以不清除浮动
4. 现在，请参照截图，在页脚区添加作者信息，并使用 span 标签进行必要的修饰
5. 大功告成，保存您的网页，用浏览器浏览、测试网页，撰写实验报告

### 任务 C、认识布局（2）

本任务要求制作类似以下截图（85%效果）布局的页面，完成后的页面类似：



具体要求及步骤如下：

1. 创建一个名称类似“文立斌 C. html”的文档，将网页的标题设置为类似“文立斌 C”，然后将该文档保存在“你的文件夹”内

## 2. 基本要求

- ✓ 页面内容占浏览器窗口 95% 宽度，上下外边距 20 像素，左右外边距为 0，浏览器宽度变化时，页面内容自动居中对齐
- ✓ 以上截图中，不同颜色的区域都是一个单独的 div 盒子，这些 div 按照其中所给的类似 `wlb???`  形式定义 id 及相应的 id 选择器，各个盒子的背景色尽可能与其他同学做的区别开来
- ✓ 为方便布局（div 尺寸调整），请参照任务 B 的做法，将其中一些 div 组织在另外的 div 内（例如，将 `wlblogo`、`wlbnav` 被组织在 `wlbheader` 内，当然，截图中看不到这些幕后 div）
- ✓ 所有文字全都水平居中对齐，遗憾，暂时做不到垂直居中对齐
- ✓ `wlblogo` 底下的“Welcome to my website”为滚动字幕（marquee 标记）
- ✓ `wlbnav` 右边是 5 个导航栏（无序列表，列表中每个项目都是一个超链接）
- ✓ `wlbleft1~wlbleft4`：每个盒子的高度为外层盒子高度的 25%
- ✓ `wlbcenter1~wlbcenter2`：每个盒子的高度为外层盒子高度的 50%，其中图片自选
- ✓ `wlbright1~wlbright3`：三个盒子的高度分别为外层盒子高度的 33%、34%、33%
- ✓ 页面主体左中右 3 列自定义一定占比

## 3. 完成后保存您的网页，用浏览器浏览、测试网页，撰写实验报告

## 任务 D、CSS 与 DIV 布局应用

本任务将在以上三行两列式页面布局实验的基础上，制作下图所示页面（85% 效果）。显然，这是一个三行三列式页面布局结构的网页，页面效果接近实际应用场景。



大体要求如下：

1. 创建一个名称类似“**文立斌 D.html**”的文档，将网页的标题设置为类似“**文立斌 D**”，然后将该文档保存在“**你的文件夹**”内
2. 参照页面效果图所呈现的三行三列结构（即纵向呈现出上中下结构，而中间一行是左中右结构），有了前面一个实验任务的铺垫，这里不提供具体的操作步骤及属性设置描述，而只是给出盒子中要显示的文字内容，以便您复制使用：

唐诗欣赏

静夜思 窗前明月光 疑是地上霜 举头望明月 低头思故乡

李 白（公元 701 年——762 年 12 月）：字太白，号青莲居士，又号“谪仙人”，唐代伟大的浪漫主义诗人，被后人誉为“诗仙”，与杜甫并称为“李杜”，为了与另两位诗人李商隐与杜牧即“小李杜”区别，杜甫与李白又合称“大李杜”。《旧唐书》记载李白为山东人；《新唐书》记载，李白为兴圣皇帝李暹九世孙，与李唐诸王同宗。其人爽朗大方，爱饮酒作诗，喜交友。

唐诗宋词是中国文学史上的两颗璀璨明珠，唐代被称为诗的时代，而宋代则被称为词的时代。词源于民间，始于唐，兴于五代，盛于两宋。在宋代，随着城市的发展和市民阶层的兴起，物质生活的丰富，人们对文化生活的追求也更加强烈。

3. 这里有一个问题，正如我们看到的，这个页面的内容不仅是浏览窗口 100% 的宽度，而且高度也是 100%，解决这个问题的途径，或者说很多问题的途径，就是“度娘”！我就在度娘上搜索“div 设置百分比高度无效”，而后解决这个问题的
4. 创建一个名称类似“**文立斌 C.css**”的文档，在该文档内编写所有 CSS 代码，并将该文档保存在“**你的文件夹**”内，使用导入方式将这些 CSS 代码导入.html 文档内。注意：参照任务二，各种选择器的名称中都必须包含类似 **w1b** 这样的字符，以将您的代码与其他同学所写的代码区别开来
5. 页面浏览效果必须与前面的效果图高度相似，脚注区必须添加作者信息并用 span 标记突出显示这些信息
6. 保存网页，浏览并测试网页，撰写实验报告

## 善后处理：

1. 右键单击“**你的文件夹**”，从快捷菜单中选择执行“添加到‘198009010001 文立斌.rar’”以将整个文件夹中的所有文档压缩成一个文件
2. 提交压缩包文件“198009010001 文立斌.rar”