**元素分类**

**在讲解CSS布局之前，我们需要提前知道一些知识，在CSS中，html中的标签元素大体被分为三种不同的类型：块状元素、内联元素(又叫行内元素)和内联块状元素。**

**常用的块状元素有：**

**<div>、<p>、<h1>...<h6>、<ol>、<ul>、<dl>、<table>、<address>、<blockquote> 、<form>**

**常用的内联元素有：**

**<a>、<span>、<br>、<i>、<em>、<strong>、<label>、<q>、<var>、<cite>、<code>**

**常用的内联块状元素有：**

**<img>、<input>**

**块级元素特点：**

1、每个块级元素都从新的一行开始，并且其后的元素也另起一行。（真霸道，一个块级元素独占一行）

2、元素的高度、宽度、行高以及顶和底边距都可设置。

3、元素宽度在不设置的情况下，是它本身父容器的100%（和父元素的宽度一致），除非设定一个宽度。

**内联元素特点：**

1、和其他元素都在一行上；

2、元素的高度、宽度及顶部和底部边距**不可**设置；

3、元素的宽度就是它包含的文字或图片的宽度，不可改变。

小伙伴们你们观查一下右侧代码段，有没有发现一个问题，内联元素之间有一个间距问题，这个问题在本小节的 wiki 中有介绍，感兴趣的小伙伴可以去查看。

inline-block 元素特点：

1、和其他元素都在一行上；

2、元素的高度、宽度、行高以及顶和底边距都可设置。

text-decoration:line-through;

text-indent:2em;

line-height:1.5em;

font-family:"Microsoft Yahei";

**中文字间隔、字母间隔设置：**

如果想在网页排版中设置**文字间隔**或者**字母间隔**就可以使用    **letter-spacing**来实现。

如果我想设置英文单词之间的间距呢？可以使用 **word-spacing**来实现。

想为**块状元素**中的文本、图片设置居中样式吗？可以使用text-align样式代码，如下代码可实现文本居中显示。

盒模型宽度和高度和我们平常所说的物体的宽度和高度理解是不一样的，css内定义的宽（width）和高（height），指的是填充以里的内容范围。

因此一个元素实际宽度（盒子的宽度）=左边界+左边框+左填充+内容宽度+右填充+右边框+右边界。

在网页中，元素有三种布局模型：  
1、流动模型（Flow）  
2、浮动模型 (Float)  
3、层模型（Layer）

这三个布局模型究竟是

**流动模型（一）**

先来说一说**流动模型**，流动（Flow）是默认的网页布局模式。也就是说网页在默认状态下的 HTML 网页元素都是根据流动模型来分布网页内容的。

流动布局模型具有2个比较典型的特征：

第一点，**块状元素**都会在所处的**包含元素内**自上而下按顺序垂直延伸分布，因为在默认状态下，块状元素的宽度都为**100%**。实际上，块状元素都会以行的形式占据位置。如右侧代码编辑器中三个块状元素标签(div，h1，p)宽度显示为100%。

第二点，在流动模型下，**内联元素**都会在所处的包含元素内从左到右水平分布显示。（内联元素可不像块状元素这么霸道独占一行）

右侧代码编辑器中内联元素标签a、span、em、strong都是内联元素。

## 浮动模型

块状元素这么霸道都是独占一行，如果现在我们想让两个块状元素并排显示，怎么办呢？不要着急，设置元素浮动就可以实现这一愿望。

任何元素在默认情况下是不能浮动的，但可以用 CSS 定义为浮动，如 div、p、table、img 等元素都可以被定义为浮动。如下代码可以实现两个 div 元素一行显示。

## 什么是层模型？

什么是层布局模型？层布局模型就像是图像软件PhotoShop中非常流行的图层编辑功能一样，每个图层能够精确定位操作，但在网页设计领域，由于网页大小的活动性，层布局没能受到热捧。但是在网页上局部使用层布局还是有其方便之处的。下面我们来学习一下html中的层布局。

如何让html元素在网页中精确定位，就像图像软件PhotoShop中的图层一样可以对每个图层能够精确定位操作。CSS定义了一组定位（positioning）属性来支持层布局模型。

层模型有三种形式：

1、**绝对定位**(position: absolute)

2、**相对定位**(position: relative)

3、**固定定位**(position: fixed)

## 层模型--绝对定位

如果想为元素设置层模型中的绝对定位，需要设置**position:absolute**(表示绝对定位)，这条语句的作用将元素从文档流中拖出来，然后使用left、right、top、bottom属性相对于其最接近的一个具有定位属性的父包含块进行绝对定位。如果不存在这样的包含块，则相对于body元素，即相对于**浏览器窗口**。

## 层模型--相对定位

如果想为元素设置层模型中的相对定位，需要设置position:relative（表示相对定位），它通过left、right、top、bottom属性确定元素在**正常文档流中**的偏移位置。相对定位完成的过程是首先按static(float)方式生成一个元素(并且元素像层一样浮动了起来)，然后相对于**以前的位置移动，**移动的方向和幅度由left、right、top、bottom属性确定，偏移前的位置保留不动。

## 层模型--固定定位

fixed：表示固定定位，与absolute定位类型类似，但它的相对移动的坐标是视图（**屏幕内的网页窗口**）本身。由于视图本身是固定的，它不会随浏览器窗口的滚动条滚动而变化，除非你在屏幕中移动浏览器窗口的屏幕位置，或改变浏览器窗口的显示大小，因此固定定位的元素会始终位于浏览器窗口内视图的某个位置，不会受文档流动影响，这与background-attachment:fixed;属性功能相同。以下代码可以实现相对于**浏览器视图**向右移动100px，向下移动50px。并且拖动滚动条时位置固定不变。

实例：

<https://www.w3cschool.cn/html/>

<https://www.w3cschool.cn/html/codedemo-484056575.html>

<html>

<head>

<title>Layout Example</title>

<style>

#header {

background-color: #cccccc;

padding: 10px;

height: 120px;

}

#navigation {

background-color: #efefef;

padding: 10px;

height: 40px;

}

#footer {

background-color: #cccccc;

padding: 10px;

height: 40px;

clear: both;

border-top: 20px solid #ffffff;

}

/\* 3 columns \*/

.column1of3, .column2of3, .column3of3 {

float: left;

width: 28%;

margin-left: 2%;

background-color: #cccccc;

padding: 1%;

margin-top: 20px;

height: 175px;

}

</style>

</head>

<body>

<div id="header">header</div>

<div id="navigation">navigation</div>

<div class="column1of3">column 1 of 3</div>

<div class="column2of3">column 2 of 3</div>

<div class="column3of3">column 3 of 3</div>

<div id="footer">footer</div>

</body>

</html>