

CSS 基础

李绍文

CSS = Cascading Style Sheet

层叠样式表

用于控制网页样式并可以将样式和内容进行分离的一种标记性语言

如何使用 CSS 样式

行内式，行内样式，直接在html 的 style 属性内加样式

```
<p style="color:#FF00FF; font-size:25px; " >行内样式</p>
```

内嵌式，通过 style 标签嵌入在 HTML 文件中

```
<style>p {color: #ff00ff;}</style>
```

链接式，通过 link 标签引入外部 css 文件

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

导入式，通过 @import 导入外部 css 文件

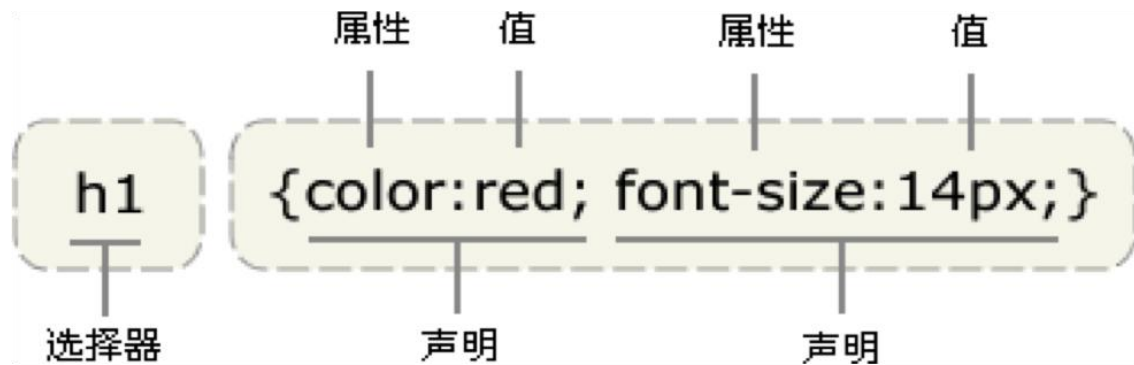
```
@import url(style.css)
```

通常情况下，推荐使用链接式

CSS 规则

由两个主要的部分构成：选择器，以及一条或多条声明

声明由属性和值两部分构成，CSS规则格式如下：



常见CSS 选择器

标签选择器

```
body {margin: 0;}
```

class 选择器

```
.notice {color: red;}
```

id 选择器

```
#logo {font-size: 48px;}
```

伪类选择器

```
.notice:hover {color: blue;}
```

伪元素选择器

```
.notice::before {content: 'hello';}
```

属性选择器

```
[data-user] {color: red;}
```

伪类选择器

常用来表示一些元素的状态

常见的伪类选择器有:



```
a:link、a:visited、a:hover、a:active  
:checked  
  
:first-child、:last-child、:nth-child()  
  
:nth-of-type( )  
:not( )
```

伪元素选择器

类似html标签元素，逻辑上存在，但是实际并不存在在文档中

常见的伪元素选择器有:



```
::first-letter
```

```
::before
```

```
::after
```

```
::selection
```

选择器优先级

选择器的优先级从高到低依次为：

使用了 !important 声明的规则

内嵌在 HTML 元素的 style 属性里面的声明

使用了 ID 选择器的规则

使用了类选择器、属性选择器、伪元素和伪类选择器的规则

使用了元素选择器的规则

选择器优先级

一个元素同时应用多个 class, 后定义的优先



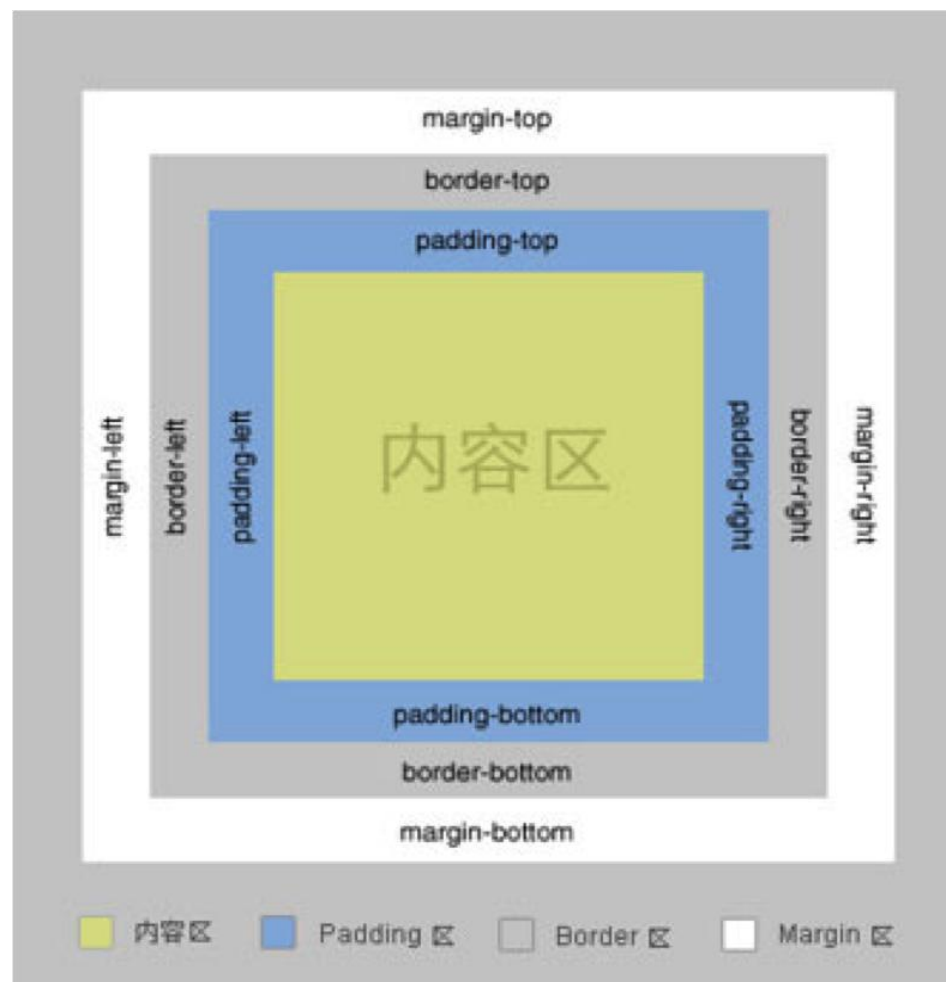
```
<style>
  .red {color: red;}
  .blue {color: blue;}
</style>

<div class="red blue">text</div>
<div class="blue red">text</div>
```

两个 text 显示什么颜色？

盒模型

浏览器在布局时把页面元素当成一个个矩形来处理，CSS盒模型的示例图如下



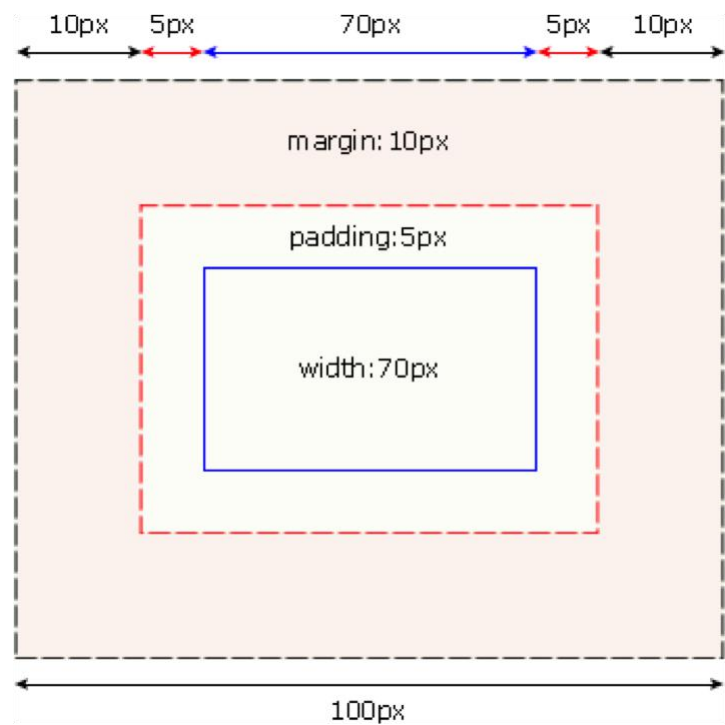
盒模型

CSS样式定义的宽（width）和高（height），指的是内容区的宽和高
盒子的实际尺寸包括内容区、内边距、边框、外边距的尺寸

盒模型



```
#box {  
  width: 70px;  
  margin: 10px;  
  padding: 5px;  
}
```



盒模型

box-sizing 属性
可以改变盒模型

布局

布局相关的属性

position

display

float

top/right/left/bottom

z-index

布局的机制

四种常见的机制

正常文档流

static position
relative position

浮动

float

定位

absolute

position

fixed position

flex 布局

display: flex

正常文档流

block：块级元素

- 1.从上到下一个接一个地排列
- 2.外边距叠加特性

inline：行内元素

- 1.水平方向、顺序布局
- 2.不能设置高度和宽度，宽度由实际内容决定
- 3.垂直方向的padding/margin无效

正常文档流

inline-block: 内部表现为block格式，外部表现为inline格式

- 1.水平方向、顺序布局
- 2.可以设置高度和宽度
- 3.垂直方向的padding/margin有效

正常文档流

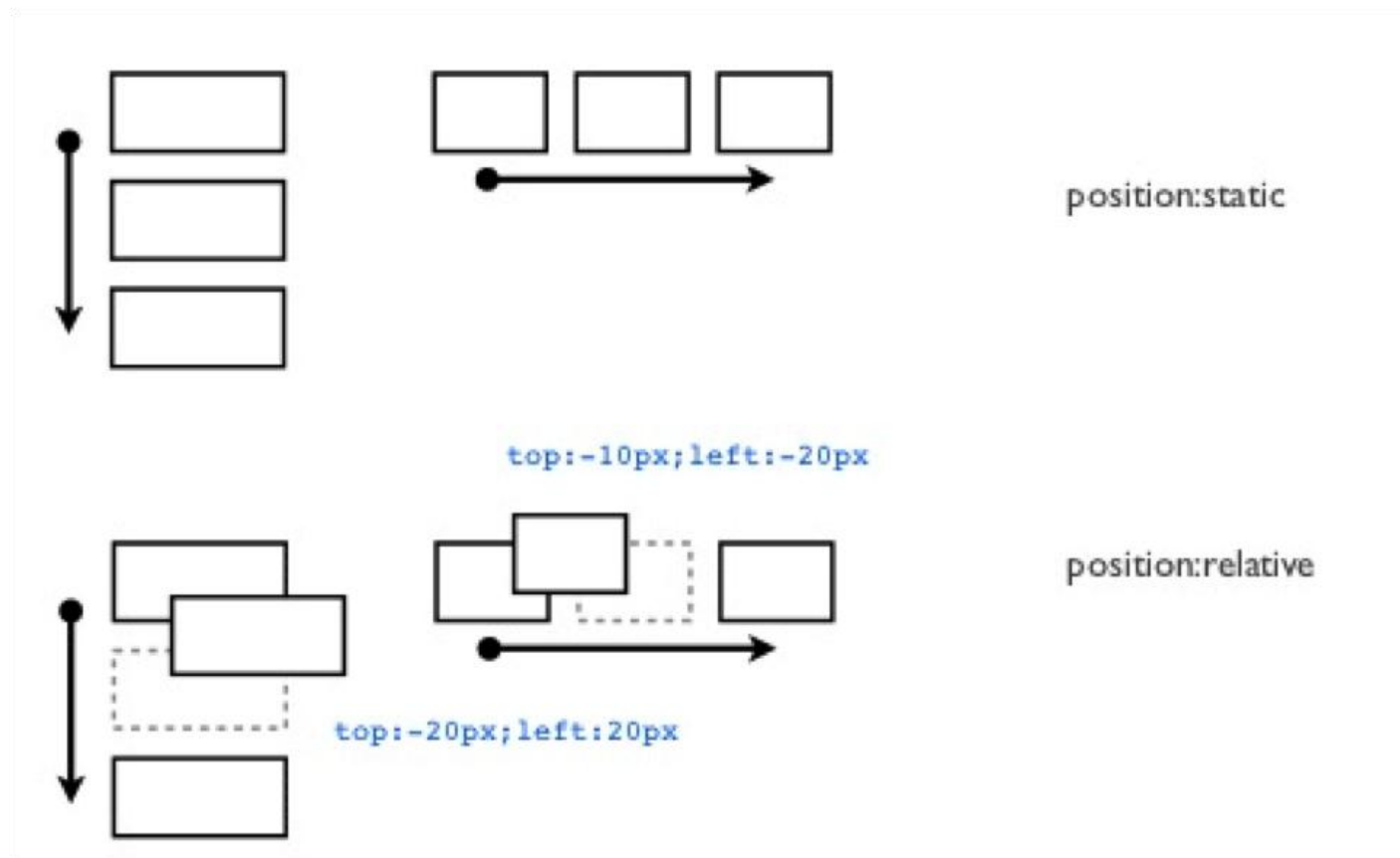
position:static

left/right/top/bottom/z-index属性无效

position:relative

本身仍占据文档流中的位置，只是相对于自身作位置移动，
通过top、bottom、right、left指定位置

正常文档流



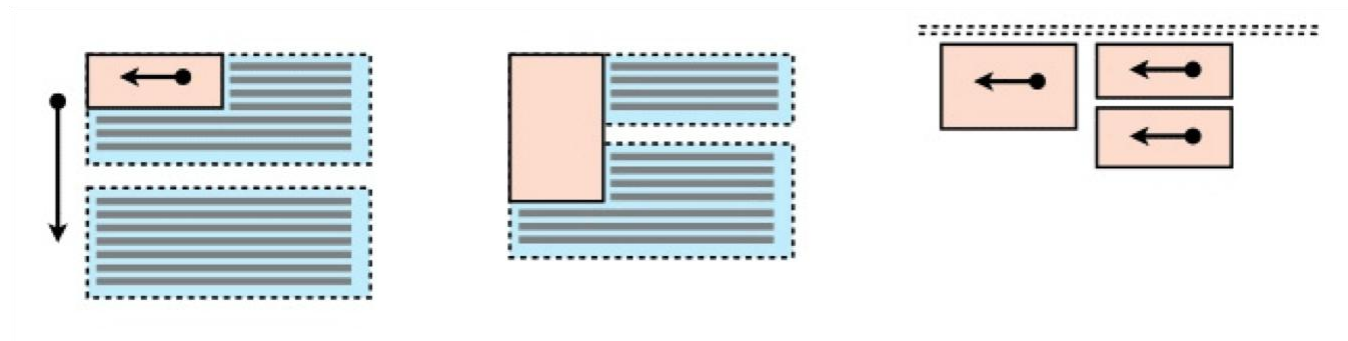
浮动

`float: left / right`

1. 脱离当前文档流，定位到父容器的边缘，或另一个float box的边缘
2. 不定宽情况下，实际宽度由内容决定，父容器水平方向空间不足时，向下移动直到可以放下
3. 会和块级重叠，用clear清除浮动避免重叠
4. left/right/top/bottom/z-index属性无效
5. float不会超出所在BFC的范围，也不会和其它BFC重叠

浮动

float: left



定位

`position: absolute`

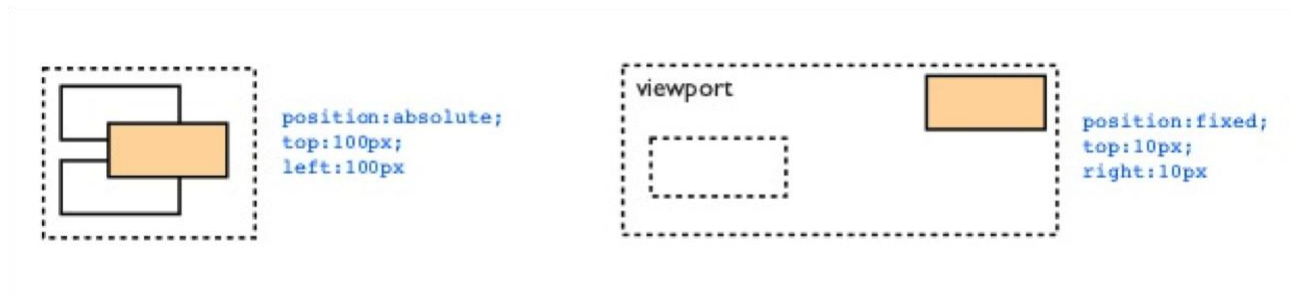
1. 脱离当前文档流，浮在其它元素上
2. 相对于其包含块进行定位。包含块指：最近的 `position` 为 `absolute` / `relative` / `fixed` 的祖先元素
3. `left/right/top/bottom/z-index`有效

`position: fixed`

一种特殊的绝对定位，以浏览器窗口定位

定位

position: absolute vs fixed



flex 弹性布局

display: flex

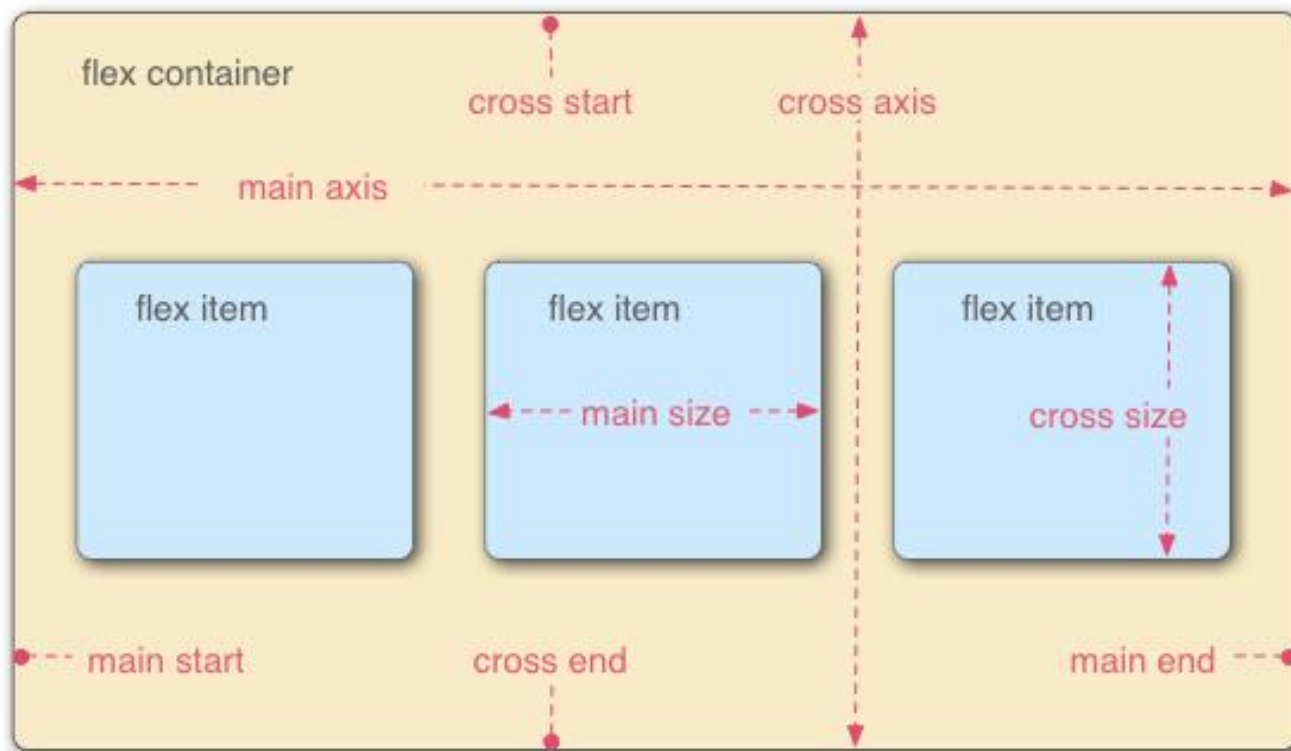
flex 弹性布局

flex container

flex 布局容器，通过 `display: flex` 开启

flex item

容器内部的元素



flex 弹性布局

flex 可以很方便地解决垂直居中 :)

额外的

Block Formatting Contexts

(块级格式化上下文)

它属于普通流布局

BFC是指浏览器中创建了一个独立的渲染区域，并且拥有一套渲染规则，他决定了其子元素如何定位，
以及与其他元素的相互关系和作用。

可触发 BFC 特性:

- 1、float的值不是none
- 2、position的值不是static或者relative
- 3、display的值是inline-block、table-cell、flex、table-caption或者inline-flex
- 4、overflow的值不是visible

BFC 的利用

- 1、创建不同的 BFC 避免 margin 重叠
- 2、BFC 可以包含浮动的元素（清除浮动）
- 3、BFC 可以阻止元素被浮动元素覆盖（创建两栏布局）

BFC

```
<style>
  .content{
    background: yellow;
    overflow: hidden;    // 触发BFC
  }
  .sidebar{
    background: green;
    float: left;
    width: 100px;
  }
</style>

<div class="wrap">
  <div class="sidebar">sidebar</div>
  <div class="content">content</div>
</div>
```


CSS 变量



```
<style>
:root {
  --main-bg-color: brown;
}

.one {
  background-color: var(--main-bg-color);
}
</style>

<div class="one">
</div>
```

BEM 命名

BEM 来自于 yandex, 是一个简单又非常有用的命名约定

```
<style>
/*
    .block 代表了更高级别的抽象或组件。
    .block__element 代表.block的后代，用于形成一个完整的.block的整体。
    .block--modifier代表.block的不同状态或不同版本。
*/

.block {}
.block__element {}
.block--modifier {}
</style>
```

CSS 处理器

preprocessor 预处理，新的语法，生成 css

saas / less

**postprocessor 后处理，对正常的 CSS 的代码进行增强处理，如兼容性的处理
(autoprefixer)**

postcss

额外要了解的

媒体查询

px / rem / vw / vh

css 动画

参考

<https://www.bemcss.com/>

<https://zh.learnlayout.com/>

<https://jonitrythall.com/flexbox-cheatsheet>

谢谢