## CSS样式

**引用方式（3）**

外链：<**link href=”css/bbb.css” type=”text/css” rel=”stylesheet”** />

内部：

<head>

<meta charset=”UTF-8” />

<title>标题</title>

**<style>**

**\* {**

**Padding: 0;**

**Margin: 0;**

**}**

**</style>**

</head>

内联：<textarea name=”ideas” **style=”resize:none;”**></textarea>

**CSS选择器**

**基础选择器（4）**

通配符\*

Id选择器#id

类选择器.class

元素选择器p

**复合选择器（6）**

交集选择器p.class/p#id/.class.Clearfkix

并集选择器p,.class,#id

后代选择器div p

子元素选择器div>p

相邻兄弟选择器P+p

属性选择器(7)

Input[type]

input[type=”text”]

Input[name~=””]含有词汇

Input[name|=””]以特定的完整的单词开头

Input[name^=””]以某值开头

Input[name$=””]以某值结束

Input[name\*=””]中间含有某值

**伪类选择器**

**伪类选择器——运用于a链接的伪类：**

：link

：visited

：hover

:active

**伪类选择器——运用于表单的（状态）伪类：**

:enabled

:disabled

:checked

**伪类选择器——否定伪类：**

:not([type=”button])

**伪类选择器——目标伪类：**

:target{Background：red；}表示与前面元素匹配的所有元素，输入地址和#属性值时才生效

**伪类选择器——结构为类（11）：**

1:First-child

2:last-child

3:nth-child() 括号内可填(数字/2n+1/odd/even)

4:nth-last-child()

5:nth-of-type()

6:nth-last-of-type()

7:first-of-type

8:last-of-type

9:only-child

10:only-of-type

11:empty

**三个原则：**

具体性原则

就近原则

重要性原则

**选择器权重：**

\*(0)<element(1)<.class(10)<id(100)<内联(1000)<Js<!important(10000)

**伪元素**

**::after**

**::before**

**CSS字体和文本**

**Font:**font-style font-variant font-weight font-size/line-height font-family;

Font-style: normal/italic/oblique/inherit;

Font-variant: （规定字体异体）normal /small-caps(显示小型大写字母的字体)/inherit;

Font-weight: normal(400)/bold(700)/bolder(900)/lighter(100);(100-900)/inherit;

Font-size: em/px/%/inherit；

line-height: 数字/em/px/百分比;

font-family：””,””, ;

**css文本**

Text-transform: uppercase/lowercase/capitalize;

Text-align: left/center/right/justify;

Text-decoration: none/overline/line-through/underline/baseline;

Text-indent: em/px/%;

Letter-spacing: ;

Word-spacing: ;

White-space: normal/pre/nowrap;

Vertical-align: top/middle/bottom/baseline;

Visibility/display/overflow/opacity/rgba()区别

**内容溢出部分以省略号显示**

White-space: nowrap;

Overflow: hidden;

Text-overflow: ellipsis;

**背景属性**

Background:background-color background-image background-repeat background-position;

1 Background-color:;

2 Background-image:url();

3 Background-repeat: repeat/no-repeat/inherit；

4 Background-position: px/百分比/center center；

5 Background-attachment：scroll/fixed;

6 Background-size: cover/contain/200px 200px/20% 30%;

**渐变色**

linear-gradient(color,color,color);可以加百分比/加角度/加方向to bottom right/重复渐变repeating-linear-gradient();

Racial-gradient(color,color,color);可以加圆形circle渐变/可以加百分比。

**盒子**

外边距之双坑

1.白色空间折叠：上下两个元素-上边的元素设置下外边距，下边的元素设置上外边距，则会取最大值。

2.父子元素，子元素的外边距效果出现在父元素

坑二之填坑方法（4种）

①给父元素加overflow:hidden;

②给父元素加一个透明边框

③给父元素加一个内边距或者直接给子元素定位

④给当前元素转换元素类型display: inline-block；

**元素类型**

哪些元素会生成BFC？

1.根元素

2.float属性不为none

3.position为absolute或fixed

4.display为inline-block, table-cell, table-caption, flex, inline-flex

5.overflow不为visible

·行内块元素：display: inline-block; (IE8以下无效)

特点：

1行内摆放，相邻元素依次排列

2可以设置宽高和内外边距

3在不设置宽高时，宽高由内容撑开

·行内元素：display: inline;

特点：

1.行内摆放，相邻元素依次排列

2.不可以设置宽高和上下内外边距

3.在不设置宽高时，宽高由内容撑开

4.不可以包含块元素

·块元素：display: block;

特点：

1独占一行

2可以设置宽高和内外边距

3不设置宽度时，自动撑满父级元素

设计浮动的初衷是为了实现文字环绕的特效，后来用于布局

元素浮动后的八大特征：

1浮动可以使块元素在一行显示

2浮动使行内元素具有块级元素的特征

3浮动元素不设置宽高时，宽度由内容撑开

4浮动元素向指定的方向移动，知道他的边框碰到包含框或者其他浮动元素的边框位置

5浮动元素脱离文档流，浮动后的无法撑开父元素

6向同一方向浮动的元素形成流式布局

7浮动会影响后续布局

8浮动元素会脱离文档流，提升层级

**清除浮动的方式方法（5种）**

1给父级加高度

2给父级加overflow:hidden;（触发BFC）

给父级加overflow:auto;

（必须定义width或者zoom:1;，同时不能定义height，使用时，浏览器会自动检查浮动区域的高度）

3给浮动元素后方加一个空的块标签，设置clear:;清除浮动

4万能清浮动（使用伪元素::after和zoom）

5给父级加display:inline-block;（触发BFC）

父级div也一起浮动（所有代码一起浮动，就变成了一个整体，没有优点，会产生新的福鼎问题，不推荐使用，只作了解）

给父级加display：table;（将div属性变成表格，没有优点，会产生新的未知问题，不推荐使用，只作了解）

\*块元素会填补浮动元素的空缺

\*行内级元素会围绕浮动元素显示（行内块也会围绕浮动元素显示）

\*尽管父元素高度塌陷，但是父元素包含了子元素，所以父元素宽度还是会影响子元素的显示

流体特性和BFC下的两栏自适应布局（此处针对的还是传统布局，关于现代布局，eg：弹性盒子模型布局Flexbox Layout，格栅布局Grid Layout不在探讨范围内）

博客网址：https://www.zhangxinxu.com/wordpress/2015/02/css-deep-understand-flow-bfc-column-two-auto-layout/

流体特性

块状水平元素，如div元素（下同），在默认情况下（非浮动、绝对定位等），水平方向会自动填满外部的容器；如果有margin-left/margin-right, padding-left/padding-right, border-left-width/border-right-width等，实际内容区域会响应变窄。

与纯流体特性布局的优势

BFC自适应布局优势我总结了下面2点：

1.自适应内容由于封闭，更健壮，容错性强。比方说，内部clear:both不会与兄 弟float产生矛盾。而纯流体布局，clear:both会让后面内容无法和float元素在 一个水平上，产生布局问题。

2.自适应内容自动填满浮动以为区域，无需关心浮动元素宽度，可以整站大规模 应用。而纯流体布局，需要大小不确定的margin/padding等值撑开合适间距，无 法CSS组件化。

自适应布局：

.float-left {

float: left; margin-right: 20px;

}

.bfc-content {

overflow: hidden; background-color: #beceeb;

}

BFC

BFC定义：

BFC(Block formatting context)直译为"块级格式化上下文"。它是一个独立的渲染区域，只有Block-level box参与， 它规定了内部的Block-level Box如何布局，并且与这个区域外部毫不相干。

BOX-Type（盒子类型）:

· block-level box: display属性为block ； list-item ； table的元素（参与formatting context）

· inline-level box: display 属性为inline inline-block inline-table的元素，会生成inline-level box 并参与inline formatting context.

· run-in box: css3中的内容。

Formatting context（css2.1）

它是页面中的一块渲染区域，并且有一套渲染规则，它决定了其子元素将如何定位，以及和其他元素的关系和相互作用。

常简的 formatting context 有

Block formatting context (BFC)

Inline formatting context (IFC)

GFc

FFC(后两个是css3中的内容)

BFC布局规则

1.内部的Box会在垂直方向，一个接一个地放置。

2.Box垂直方向的距离由margin决定。属于同一个BFC的两个相邻Box的margin会发生重叠

3.每个元素的margin box的左边， 与包含块border box的左边相接触(对于从左往右的格式化，否则相反)。即使存在浮动也是如此。

4.BFC的区域不会与float box重叠。

5.BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。反之也如此。

6.·计算BFC的高度时，浮动元素也参与计算

哪些元素会生成BFC？

1.根元素

2.float属性不为none

3.position为absolute或fixed

4.display为inline-block, table-cell, table-caption, flex, inline-flex

5.overflow不为visible

盒子

外边距之双坑

1.白色空间折叠：上下两个元素-上边的元素设置下外边距，下边的元素设置上外边距，则会取最大值。

2.父子元素，子元素的外边距效果出现在父元素

坑二之填坑方法（4种）

①给父元素加overflow:hidden;

②给父元素加一个透明边框

③给父元素加一个内边距或者直接给子元素定位

④给当前元素转换元素类型display: inline-block；

利用边框画形状

1.画胶囊：radius设置为高度的一半

2.画椭圆：radius设置为100%或者50%

3.画三角形：

Div {

width: 50px;

height: 50px;

border: 50px solid #adccee;

Width: 0;

Height:0;

Border-top-color: transparent;

Border-left-color:transparent;

Border-right-color: transparent;

}

如何设置一个满屏的颜色

Body,div {

Width: 100%;

Height: 100%;

Background-color: violet;

}

定位

定位类型：

默认定位属性：static

相对定位属性：relative

绝对定位属性：absolute

固定定位属性： fixed

①默认定位属性：静态定位，无任何特殊作用

②相对定位属性：

1.相对定位元素不会脱离文档流

2.相对定位元素会提升元素层级

3.Left,top等值生效

4.相对定位元素不会改变元素自身的元素类型

5.相对定位元素还会保留原来所占的位置

6.相对定位的定位原点是元素本身原来所占的位置（原始位置）

③绝对定位属性：

1. 对定位元素会脱离文档流

2. 绝对定位元素会提升元素层级

3. Left，top等值生效

4. 绝对定位元素不会保留原来所占的位置

5. 绝对定位元素元会具有块级元素的一些特性

6. 绝对定位元素不设置宽高时，宽度有内容决定，可以设置宽高，可以设置内外边距

7. 绝对定位的定位原点是离自己最近的设置了定位属性的祖先元素

④固定定位属性：

1. 固定定位元素会脱离文档流

2. 固定定位元素会提升元素层级

3. Left，top等值生效

4. 固定定位元素不会保留原来所站的位置

5. 固定定位元素会具有块级元素的一些特性

6. 固定定位元素不设置宽高时，宽度有内容决定，可以设置宽高和内外边距

7. 固定定位的定位原点是可视窗口，固定定位元素位置不随页面滚顶而滚动

百分比

1.让元素垂直居中于父盒子：

2. 绝对定位，设置子元素top:50%;left:50%;

3. 然后再设置margin-top和margin-left

4. margin值为小盒子宽高的一半

浮动

元素类型

·行内块元素：display: inline-block; (IE8以下无效)

特点：

1.行内摆放，相邻元素依次排列

2.可以设置宽高，可以设置内外边距

3.在不设置宽高时，宽高由内容撑开

·行内元素：display: inline;

特点：

1.行内摆放，相邻元素依次排列

2.不可以设置宽高，不可以设置上下内外边距

3.在不设置宽高时，宽高由内容撑开

4.不可以包含块元素

·块元素：display: block;

特点：

1.独占一行

2.可以设置宽高

3.可以设置内外边距

4.不设置宽度时，自动撑满父级元素

浮动

设计浮动的初衷是为了实现文字环绕的特效，后来用于布局

元素浮动后的八大特征：

1.浮动可以使块元素在一行显示

2.浮动使行内元素具有块级元素的特征

3.浮动元素不设置宽高时，宽度由内容撑开

4.浮动元素向指定的方向移动，知道他的边框碰到包含框或者其他浮动元素的边框位置

5.浮动元素脱离文档流，浮动后的无法撑开父元素

6.向同一方向浮动的元素形成流式布局

7.浮动会影响后续布局

8.浮动元素会脱离文档流，提升层级