

- [CSS](#)
- [css 2.1](#)
 - [css总体结构](#)
- [选择器](#)
 - [选择器优先级](#)
 - [思考](#)
- [排版](#)
 - [盒子 和 盒模型](#)
 - [正常流排版](#)
 - [IFC](#)
 - [文字排版](#)
 - [float与clear](#)
 - [BFC](#)
 - [margin折叠](#)
 - [overflow:visible 与 BFC](#)
 - [flex 布局](#)

CSS

css 2.1

- 语法
 - <https://www.w3.org/TR/css-syntax-3/>
 - <https://www.w3.org/TR/CSS21/grammar.html%23q25.0>

css总体结构

- @import
- @charset
- rules
 - @media
 - @page
 - rule

At-rules

- @charset：<https://www.w3.org/TR/css-syntax-3/>
- @import：<https://www.w3.org/TR/css-cascade-4/>
- @media：<https://www.w3.org/TR/css3-conditional/>
- @page：<https://www.w3.org/TR/css-page-3/>
- @counter-style：<https://www.w3.org/TR/css-counter-styles-3/>
- @keyframes：<https://www.w3.org/TR/css-animations-1/>
- @fontface：<https://www.w3.org/TR/css-fonts-3/>
- @supports：<https://www.w3.org/TR/css3-conditional/>
- @namespace：<https://www.w3.org/TR/css-namespaces-3/>

选择器

选择器优先级

四元组方式：[]

• 简单选择器计数

1

2

#id div.a#id {
 //.....
}

[0, 2, 1, 1]

$$S = 0 * N^3 + 2 * N^2 + 1 * N^1 + 1$$

取N = 1000000

$$S = 2000001000001$$

```
// 写出下面选择器的优先级
div#a.b .c[id=x]  0 1 3 1
#a:not(#b)  0 2 0 0  (not伪类选择器不参与)
*.a  0 0 1 0  (*不参与)
div.a  0 0 1 1
```

A specificity is determined by plugging numbers into (a, b, c, d):

If the styles are applied via the style attribute, a=1; otherwise, a=0.

b is equal to the number of ID selectors present.

c is equal to the number of class selectors, attribute selectors, and pseudoclasses present.

d is equal to the number of type selectors and pseudoelements present.

翻译过来就是

优先级是由 A、B、C、D 的值来决定的，其中它们的值计算规则如下：

如果存在内联样式，那么 A = 1，否则 A = 0；

B 的值等于 ID 选择器 出现的次数；

C 的值等于 类选择器 和 属性选择器 和 伪类 出现的总次数；

D 的值等于 标签选择器 和 伪元素 出现的总次数 。

思考

为什么 first-letter 可以设置 display:block 之类的，而 first-line 不行呢？

- 如果设置float，float会脱离文档流，就不是第一行了，这样会无限循环。

first-line 为什么可以改字体？

- first-line的本质，待详细补充

排版

盒子 和 盒模型

为什么有盒模型？

- 排版所需

正常流排版

- 收集盒进行
- 计算盒在行中的排布
- 计算行的排布

IFC

从左到右的： IFC（inline formatting context）

文字排版

- 关键字：基线与撑开
 - vertical-align: 建议使用top, bottom, middle
- 基线：
 - 如果内容高度不大于line-height，然后根据line-height在来确定位置和基线。
 - 如果内容有了设置了 inline-block 的元素，且内容撑开了line-height，这时候，建议一定要给 inline-block 设置 vertical-align。然后基线和文字内容会根据最高子元素的 基线为准。

参考资料：

- freetype
- dom.getClientRects()

float与clear

- 对于需要实现文字绕排的需求，回归本来的用法，使用float
- float会导致部分区域的重排，

BFC

从上到下的： BFC（block formatting context）

<https://www.w3.org/TR/2011/REC-CSS2-20110607/visuren.html#block-formatting>

Floats, absolutely positioned elements, block containers (such as inline-blocks, table-cells, and table-captions) that are not block boxes, and block boxes with 'overflow' other than 'visible' (except when that value has been propagated to the viewport) establish new block formatting contexts for their contents.

In a block formatting context, boxes are laid out one after the other, vertically, beginning at the top of a containing block. The vertical distance between two sibling boxes is determined by the 'margin' properties. Vertical margins between adjacent block-level boxes in a block formatting context collapse.

In a block formatting context, each box's left outer edge touches the left edge of the containing block (for right-to-left formatting, right edges touch). This is true even in the presence of floats (although a box's line boxes may shrink due to the floats), unless the box establishes a new block formatting context (in which case the box itself may become narrower due to the floats).

For information about page breaks in paged media, please consult the section on allowed page breaks.

margin折叠

能容纳正常流的都是BFC，而且外面也是正常流，本身overflow是visible。就是BFC

对于一个BFC来说，margin其实是一个留白，只要保证周围有足够的留白就可。发生折叠其实就是留白重复利用了。

- 相互独立的不同BFC影响不同。
- 产生新的BFC:
 - overflow: visible
 - display: inline-block;

overflow:visible 与 BFC

flex 布局