# css中的概念

关键字

inherit：泛关键字（所有属性都可以用）

solid，transparent

变量（特殊关键字）

currentColor（当前继承的文字颜色）

对于字体图标颜色变换很有用（悬浮变色图标）

关系选择器

后代选择器：空格

子元素选择器： >

兄弟选择器 ~

相邻后面的兄弟选择器： +

长度单位

相对长度单位

em

ex

rem

ch

绝对长度单位

功能符

值以函数的形式指定（括号括起来的那种）

盒模型

分为块级盒子和内联盒子

# 伪类：

同一类型中的第一，最后一个

:first-of-type ， :last-of-type

# 文档流

# 单位

px

em ：相对于当前对象内文本字体大小

rem ：相对于根元素的字体大小

# css层叠规则

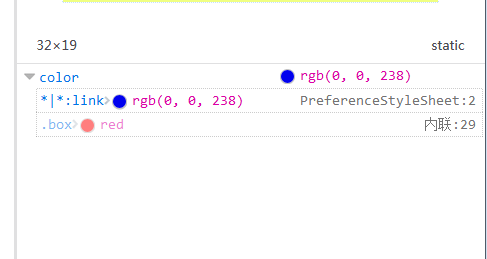
## 来源优先级：

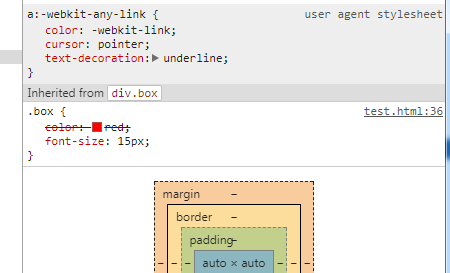
用户代理 < 用户一般样式 < 作者一般样式 < 作者重要样式 < 用户重要样式

注意：

继承样式优先级最低

如下列a继承的color优先级比用户代理优先级低





## 权重：

a,b,c,d

元素内嵌样式： a

id选择器相加数论： b

class选择器，属性选择器， 伪类选择器相加： c

标签选择器，伪元素相加： d

注意：

通用选择符 \* ， 结合符的特异度为 0

# css解析

1. 空格不敏感（没有分号，会将后面的样式作为一个整体）
2. 遇到无法识别的属性或者值时，会忽略这个属性（复合属性中有一个值无法识别，整个属性作废）
3. 换行符会被视为空格。
4. 关键字不能用引号（会被忽略）
5. 多组选择器中，其中一个选择器如果是错的，其他的也不会被执行
6. @import只能放在所有样式规则之前。
7. 属性值计算

在客户端解析文档并生成文档树之后，会将文档中每一个元素的属性分配一个值，这个值的最终值可能会进过指定值，计算值，使用值，实际值 4个步骤。

1. 指定值
2. 计算值
3. 使用值

计算值会被尽可能的被处理，（如em，rem，百分比），但是一些情况下，需要在渲染时才能确认，如百分比宽度，必须要父元素不确定宽度时，这时，必须要包含快宽度被确定，才能计算。

1. 实际值

# 继承

继承的值时父元素的计算值

inherit关键字

无论是否继承属性，都可以通过inherit关键字继承。

# 替换元素

定义：

指的是：浏览器根据元素的标签和属性来决定元素的具体内容 。

如 img ，根据 src属性

如 input ，根据 type属性

而非替换元素，则是直接显示出来。

# 定位机制

## 普通流

## 浮动

浮动的框可以左右移动，知道他的外边缘遇到包含框或则另一个浮动框的边缘

不会影响块级框，只会影响到内联框的排列。

构建一个BFC(块级格式化上下文: 闭合浮动 防止marign重叠 多栏布局的一种)

float:除了none外的值

overflow:除了visible外的值

display: table-cell , table-caption, inline-block( table:这个与前面的是不一样的，table会产生匿名框，匿名框中的table-cell会创建BFC )

position: absolute fixed

fieldset 元素

### 闭合浮动一些好的的解决方法

一、零宽度空格

.clearfix:after {content:"200B"; display:block; height:0; clear:both; }

.clearfix { \*zoom:1; }

二、避免内部元素的margin-top与上一个盒子的margin-top叠加（如果只写after）

cf:before,.cf:after {

content:"";

display:table;

}

## 绝对定位

（fixed是absolute的一个子类）

# 一些css技巧

## 镂空图标

仅仅适用于背景色固定的情况下

图标镂空，周边底色为背景色。

可以调节镂空图标背景来切换图标颜色（用currentColor更适合）

## 负margin

margin-left/margin-top : 使得元素位移

margin-bottom/margin-right : 不会位移

margin-right: 如果没有固定宽度，则会增加元素宽度

margin-botton: 不会增加高度，会减少供CSS读取的高度（本身显示的高度不会变，但是下面的元素会向上浮动，遮蔽该元素）

## 居中布局

水平居中

行内元素: text-align :center

块级元素： margin: 0 auto;

flex布局

垂直居中

行内元素

单行

上下padding相等 line-height与height相等

多行

上下padding相等 display:table-cell vertical-align:middle

flex/伪元素

块级元素

已知高度

绝对定位

top:50%;margin-top:-高度的一般

未知高度

绝对定位

top:50%; transorm:translateY(-50%)

flex

**水平垂直居中**

绝对定位：

定高定宽

1. top:50%;left:50%; margin-left: - marign-right:-

2. top:0;right:0;bottom:0;left:0;top:0;margin:auto;

未知高宽

1 . top:50%;left:50%;transform:translate(-50%,-50%)

1. 父盒子设置:display:table-cell; text-align:center;vertical-align:middle;

Div 设置: display:inline-block;vertical-align:middle;

flex

## 通过pointer-events控制鼠标事件

元素在什么情况下可以成为鼠标事件事件的target。（点透）

target:none;

缺陷：

这会导致 如cursor： not-allowed这种属性失效。

注意：

子元素仍然会触发事件，仍然会通过冒泡触发该父元素的事件

用处：

1. 在页面滚动，或其他操作时，避免用户意外触发跳转，可以在body上加人这个属性，可以避免。
2. 对遮罩层设置none，可以点击到遮罩层后面的内容。

## 选择子元素中唯一的同一元素类型（而不是class）

p:last-of-type:first-of-type

而 ：only-child选中时唯一的元素

# 经典布局

## 圣杯布局与双飞翼布局（两边固定，中间自适应）

优点：相较于一般的浮动布局，页面有限加载中间部分（重要内容）

<!-- 圣杯的 HTML 结构start -->

<div class="container">

<!-- 中间的 div 必须写在最前面 -->

<div class="middle">中间弹性区</div>

<div class="left">左边栏</div>

<div class="right">右边栏</div>

</div>

<!-- 圣杯的 HTML 结构end -->

.container{ width: 1200px;margin:0 auto;}

.middle,.left,.right{border:1px solid black;height: 500px;}

.middle{width:100%;float: left;padding-left: 150px;padding-right:200px;box-sizing: border-box;;}

.left{margin-left: -100%;border-color: green;width:150px;float: left;box-sizing: border-box;}

.right{margin-left:-200px;width: 200px;border-color: red;float: left;box-sizing: border-box; }

<!-- 双飞翼的 HTML 结构start -->

<div class="container">

<!-- 中间的 div 必须写在最前面 -->

<div class="middle\_t">

<div class="middle-inner">中间弹性区</div>

</div>

<div class="left">左边栏</div>

<div class="right">右边栏</div>

</div>

<!-- 双飞翼的 HTML 结构end -->