# 1 常用属性

## 1.1 display的属性值及其作用



## 1.2 link和@import的区别

两者都是外部引用CSS的方式，它们的区别如下

**本质区别：**

1.Link是html的标签之一，而import只是css提供的一种方法

2.Link除了加载css之外还可以加载其他文件，@import只能加载css

3.Link引入的css与html解析并行进行，而@import引入的css在html解析后执行

4.Link无兼容性问题，@import低版本浏览器不支持

## 1.3 transition、animation、transform

1.transition是过渡属性，以动画效果改变css，需要事件来触发

效果：将鼠标悬停在一个 div 元素上，逐步改变表格的宽度从 100px 到 300px：

div

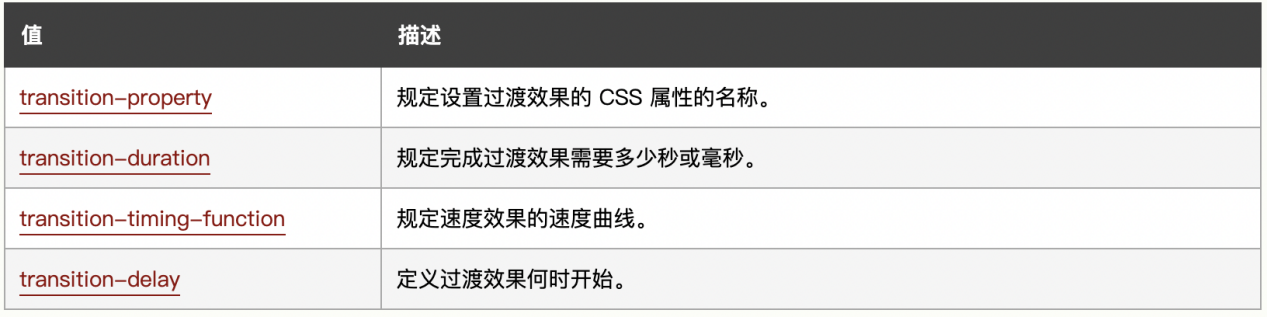
{

width:100px;

transition: width 2s;

}

div:hover {width:300px;}

2.animation使用keyframes来实现动画属性，不需要事件触发

效果：div从左往右移动，5s内移动200px，infinite定义这个动画无限循环

div

{

width:100px;

height:100px;

background:red;

position:relative;

animation:mymove 5s infinite;

}

@keyframes mymove

{

from {left:0px;}

to {left:200px;}

}

keyframes定义关键帧，from代表0%，to代表100%

3.transform

对元素进行缩放，旋转，移动和倾斜

常用scale、rotate、translate、skew

区别：

1.transition需要事件触发，animation不需要

2.transition为两帧，animation可以定义很多帧

3.transition只触发一次，animation可以循环无数次

## 1.4 display:none与visibility:hidden的区别

二者都是让元素隐藏

1. display：none会导致元素从渲染树中消失（DOM树中仍然存在），而visibility：hidden则不会
2. display:none不在页面上占据位置，visibility：hidden会占据位置

2.display会导致子节点也从渲染树中消失，visibility：hidden的子元素会继承氟元素的visibility属性，导致子元素隐藏

## 1.5 对line-height 的理解及其赋值方式

1.行高是指两行文本基线之间的距离

2.行高和height都可以撑开容器高度，若没有设置height，容器高度由行高决定

3.行高=height，可以使得单行文字垂直居中

4.带单位：px

纯数字：按比例计算，这个比例是根据当前字体的大小来计算的，例如，line-height=1，font-size=16px，那么行高为16px\*1=16px

百分比：百分比值乘以当前元素的字体大小

5.height>line-height，文字偏上，height<line-height，文字靠下

## 1.6 伪元素

**伪元素,使用::before和::after在元素之前/之后插入一个在DOM中并不存在的节点，添加的内容默认是inline元素；**这个伪元素的content属性，表示伪元素的内容,设置::before和::after时必须设置其content属性，否则伪元素就不起作用。

优点：不影响html的结构，代码简洁。常用于清除浮动

缺点：无法用js控制

## 1.7 z-index属性在什么情况下会失效

z-index值越大越往上，z-index属性在下列情况下会失效。

1.父元素position为relative时，子元素的z-index失效

2.元素position属性为static属性

3.元素浮动

## 1.8 position属性

**1absolute** ：绝对定位；脱离文档流的布局，遗留下来的空间由后面的元素填充。

**2.relative** ：相对定位；不脱离文档流的布局，只改变自身的位置，在文档流原先的位置遗留空白区域。

**3.fixed** ：固定定位；类似于absolute，但不随着滚动条的移动而改变位置。

**4.static** ：默认值；默认布局。 此时top、bottom、left、right、z-index无效

**5.sticky**：粘性布局；位于页面中，表现为relative，位于页面外，表现为fixed

sticky效果受父元素影响，fixed父元素为根元素，不受原本父元素影响

## 1.9 Flex布局

Flex是”弹性布局”，采用Flex布局的元素，称为Flex容器，简称”容器”。子元素称为Flex项目，简称”项目”。



容器默认存在两根轴：水平的主轴（main axis）和垂直的交叉轴（cross axis）。项目默认沿主轴排列。项目占据的主轴空间叫做main size，占据的交叉轴空间叫做cross size。

1.flex-direction定义了flex元素的排列方式

flex-direction：row（默认值）主轴为水平方向，起点在左端。

flex-direction：column 主轴为垂直方向，起点在上沿。

2.flex-wrap 主轴满了是否换行

flex-wrap ：nowrap 不换行压缩宽度

flex-wrap ：wrap 换行

3.

justify-content属性定义了项目在主轴上的对齐方式 justify-content：center水平居中

align-items属性定义项目在交叉轴上如何对齐，align-items：center垂直居中

## 1.10 display、float、position的关系

float元素可以在dom中找到

1. （display:none position：any float：any ） float和position被忽略
2. （Display：!none Position：absolute/fixed float：any） float被忽略，display转为table或者block
3. （Display：!none Position：relative/static float：left/right） 元素浮动，display转为table或者block

## 1.11 absolute与fixed共同点与不同点

**共同点**

使元素脱离普通文档流，不再占据文档物理空间。

**不同点**

1.根元素不同，absolute相对于父元素进行定位，fixed相对浏览器进行定位

2.滚动页面时，absolute会跟着父元素进行移动，fixed固定在页面的具体位置。

## 1.12 对 sticky 定位的理解

sticky 粘性定位。语法：**position: sticky**

粘性定位依赖于用户的滚动，在 **position:relative** 与 **position:fixed** 定位之间切换。 而当页面滚动超出目标区域时，会变成固定定位

## 1.13 设置小于12px的字体

在谷歌下css设置字体大小为12px及以下时，显示都是一样大小，都是默认12px。

1.修改浏览器内核的私有CSS属性

2.使用css3的transform缩放

## 1.14 png、jpg、gif

png是一种无损压缩的文件格式.

jpg是一种失真压缩方法，是一种破坏性的压缩

gif可以实现动画效果。

## 1.15 overflow

overflow 属性规定当内容溢出元素框时所采取的行为

overflow:auto 不溢出不显示滚动条，溢出显示滚动条

overflow:hidden 溢出部分隐藏

overflow:scroll 添加滚动条

overflow:visible 框外显示内容

## 1.16 float属性

浮动可以理解为让某个div元素脱离标准流，漂浮在标准流之上

# 2 常考元素布局方法

## 2.1 常见的实现左右两栏布局的方法

·通过float浮动效果

·通过flex 实现左右两栏布局

·通过position absolute 绝对定位元素。

## 2.2 隐藏元素的方法有哪些

·**display: none**：渲染树不会包含该渲染对象，该元素不会在页面中占据位置

·**visibility: hidden**：元素在页面中仍占据空间，但是不会响应绑定的监听事件

·**opacity: 0**：将元素的透明度设置为 0，以此来实现元素的隐藏。

·**position: absolute**：使用绝对定位将元素移到可视区域之外

·**z-index: 负值**：来使其他元素遮盖住该元素，以此来实现隐藏

## 2.3 判断元素是否到达可视区域

window.innerHeight：浏览器可视区域的高度

document.body.scrollTop：浏览器滚动过的距离

div.offsetTop：元素顶部到文档起始的距离

元素到达可视区域：div.offsetTop<window.innerHeight+document.body.scrollTop

## 2.4 常见的CSS布局单位

常用的布局单位包括像素（px），百分比（%），em，vw/vh。

**像素**（px）,一个像素表示屏幕所能显示的最小的区域

**百分比**（%），相对于父元素的宽高百分比

**em：** em=当前元素字体尺寸fontsize。rem=根元素字体大小尺寸,默认16px

vw/vh：视图窗口的宽高

## 2.5 清除浮动

**浮动溢出的定义：**容器不设高度且子元素浮动时，容器高度不能被内容撑开

清除浮动的方法：

1.父元素给height

2.父元素设置overflow:hidden

3.父元素后追加::after伪元素，并设置clear:both样式

4.子元素后追加div，并设置clear:both样式

## 2.6 元素的层叠顺序

由下到上分别是：

（1）背景和边框：建立当前层叠上下文元素的背景和边框。

（2）负的z-index：当前层叠上下文中，z-index属性值为负的元素。

（3）块级盒：文档流内非行内级非定位后代元素。

（4）浮动盒：非定位浮动元素

（5）行内盒：文档流内行内级非定位后代元素。

（6）z-index:0：层叠级数为0的定位元素。

（7）正z-index：z-index属性值为正的定位元素

## 2.7 常见居中方案

·水平居中：text-align：center。行内内容（例如文字）如何相对它的块父元素对齐。也可以在块元素中使用，来将其中的行内元素对齐

·垂直居中：vertical-align：middle 只对行内、行内块元素生效，块级元素不行

·flex布局：设置justify-content：center水平居中，align-items：center 垂直居中

·绝对定位+margin/padding来实现居中

## 2.8 常见的inline元素、block元素、inline-block元素

**常见的inline内联元素：**

span、a

**常见的block块级元素：**

div、p、h1…h6

**常见的inline-block内联块元素：**

img、input

# 3 CSS基础知识

## 3.1 CSS三大特性——优先级

1.!important

2.内联样式

3.id选择器

4.类选择器

5.标签选择器

6.通配符选择器

7.继承的属性

## 3.2 CSS三大特性——继承级

文字属性(text-)、字体属性(font-)、行高、元素可见性visibility等属性可以继承

## 3.3 CSS三大特性——层叠级

两个选择器设置同一个元素的属性，根据就近原则进行层叠

## 3.4 伪元素和伪类的区别和作用

伪元素：在元素前后插入在DOM树中不存在的节点

伪类：在当前的元素上添加新的状态

## 3.5 CSS盒模型

标准盒模型的width和height属性的范围只包含了content

IE盒模型的width和height属性的范围包含了border、padding和content

## 3.6 CSS3中有哪些新特性

1.弹性布局flex

2.盒模型：content-box和border-box

3.旋转：Transform：scale(x,y) 定义缩放 rotate(angle) 旋转

4.新的选择器：not选择器，除了xxx以外的所有选择器。nth-child选择器

5.文字属性：Text-shadow阴影；Text-decoration:overline/underline 下划线

6.圆角边框：border-radius

## 3.7 CSS 优化和提高性能的方法有哪些

1.css压缩：可以减小文件体积

2.使用单一样式：设置margin-left，不要设置margin

3.减少使用@import，无法并行加载

4.少使用后代选择器，降低效率

## 3.8 CSS预处理器/后处理器是什么

预处理器：sass，less，css之上的抽象层，为css添加动态语言的特性，例如函数、变量、继承

后处理器：postCss，给css添加浏览器私有前缀，解决兼容性问题

# 4 实战篇字节

## 4.1 用CSS实现一个三角形

将一个div的宽度和高度设置为0，然后设置边框样式

[复制代码](javascript:void(0);)

.triangle{

width: 0;

height: 0;

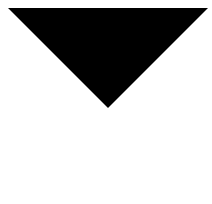
border: 50px solid transparent;

border-top-color: black ;

}

[复制代码](javascript:void(0);)

结果如下图所示：



## 4.2 transition、animation、transform是否会引起重排

transform不会，使用css3硬件加速，可以让transform、opacity、filters这些动画不会引起重排

transition和animation，要看是否修改了width、height、position等影响页面布局的属性，如果修改了，会重排。

## 4.3重排和重绘

重绘

元素外观改变，但并没有影响到布局。比如，仅修改DOM元素的字体颜色

重排

因为元素的大小和位置改变，引起部分渲染树重新构建，称为回流(reflow)。每个页面至少需要一次回流，就是在页面第一次加载的时候。

触发重排的条件：任何页面布局和几何属性的改变都会触发重排：

1.页面渲染初始化(无法避免)

2.添加或删除DOM元素

3.改变位置

4.元素大小

5.浏览器窗口缩放

重排必定会引发重绘，但重绘不一定会引发重排。

## 4.4 渲染层与合成层

1.渲染层

处于相同坐标空间（z轴空间）的渲染对象，都将归并到同一个渲染层中，因此根据层叠上下文，不同坐标空间的的渲染对象将形成多个渲染层，以体现它们的层叠关系。

2.合成层

满足某些特殊条件的渲染层，会被浏览器提升为合成层。交由GPU处理

常见的合成层：

1. transforms

2.通过CSS动画实现的 opacity 动画转换

3.video、canvas 等元素

4.position: fixed

合成层元素特点：

1.合成层的位图，会交由 GPU 合成，比 CPU 处理要快得多；

2.当需要 repaint 时，只需要 repaint 本身，不会影响到其他的层；

3.元素提升为合成层后，transform 和 opacity 才不会触发 repaint，如果不是合成层，则其依然会触发 repaint。

## 4.5响应式布局

响应式页面，是指一套页面，能够适配多种终端，如手机、平板、电脑等

常见的实现方法：

1.百分比布局

2.rem布局

3.flex布局

4.vw\vh布局

5.媒体查询@media。可以针对不同的屏幕尺寸设置不同的样式

## 4.6 css中百分比以什么为基准



百分比基本上是以父元素的对应属性为参考的，但是几个例外：

margin、padding是根据父元素的宽度来进行参考的，无论上下左右

border-radius和translate是根绝自身的宽高计算百分比的

## 4.7 border-radius

border-radius可以设置四个值

如果设置1个值，表示4个圆角都使用这个值。border-radius:40px;

如果设置两个值，表示左上角和右下角使用第一个值，右上角和左下角使用第二个值。border-radius: 10px 50px;

如果设置三个值，表示左上角使用第一个值，右上角和左下角使用第二个值，右下角使用第三个值。border-radius: 10px 50px 40px;

如果设置四个值，则依次对应左上角、右上角、右下角、左下角（顺时针顺序）。border-radius: 10px 20px 30px 40px;

若使用百分比，则表示圆形半径或椭圆的半长轴，半短轴。水平半轴相对于盒模型的宽度；垂直半轴相对于盒模型的高度

其中，10px代表以10px为半径，定格放在左上角后，留下的弧度，如图：

因此，画半圆的时候，假如width为200px，height为100px，那么，我们应该设置左上角为100px，那么就可以用100px为半径画圆，定格后得到的弧度就是个半圆。因为圆心在这个div的下边框中心

## 4.8 flex布局子元素属性

1. flex-basis 定义了该元素的空间大小，flex容器里除了元素所占的空间以外的富余空间就是可用空间 available space。 该属性的默认值是 auto 。如果元素设定了宽度为100px，那么 flex-basis 的值为100px

2.flex-grow 会使得 flex 元素以 flex-basis 为基础，沿主轴方向增长尺寸。这会使该元素延展，并占据此方向轴上的可用空间。如果有其他元素也被允许延展，那么他们会各自占据可用空间的一部分。

3.flex-shrink属性是处理flex元素收缩的问题。flex 元素仅在默认宽度之和大于容器的时候才会发生收缩，其收缩的大小是依据 flex-shrink 的值

4.flex 简写形式允许三个数值按顺序书写 — flex-grow，flex-shrink，flex-basis

例子：

flex布局，父项目div：display：flex width：1000px

子项目div1：flex：1 0 100px

子项目div2：flex：2 0 300px

求各个子项目宽度

首先，flex-shrink为0，说明不缩放。flex-basis 分别为100和300，那么可用空间就是1000-100-300=600。flex-grow分别为1和2，说明会以1:2的比例平分可用空间600px，div1分得200px，div2分得400px。因此div的宽度=100px+200px=300px；div2的宽度=300px+400px=700px

## 4.9 left、right、top、bottom

CSS left属性定义了定位元素的左外边距边界与其包含块（父元素）左边界之间的偏移，非定位元素设置此属性无效。其他属性类似

## 4.10 transform一定会开启GPU渲染图层吗

不一定

1.在animation中应用transform，才会开启GPU渲染

2.transform中适应translate3d、translateZ等

## 4.11 position:absolute带来的性能问题

如果频繁修改绝对定位元素的位置、大小，会引起页面的重绘，导致页面卡顿。

但是absolute的元素脱离了普通文档流，他不再会影响到普通文档流里的元素了，不会导致重排

这个问题可以通过transform来解决，transform运行在合成层，合成层可以通过GPU来进行硬件加速渲染，而且合成层的元素不会引起页面的重绘或者重排，不会有卡顿问题。

## 4.12 transform的详细取值

matrix：定义 2D 转换，使用六个值的矩阵。

translate(x,y)：定义水平或者垂直方向移动的距离

scale(x,y)：在水平或者垂直方向缩放

rotate(x\*deg)：顺时针旋转45度

skew(deg, deg)：在水平方向和垂直方向上倾斜

## 4.13 行内块元素和块元素区别

相同点：二者都可以设置宽和高

区别：

1.行内块元素同行显示，块元素一行显示

2.行内块宽度由内容撑开，块元素宽度默认为父元素宽度

## 4.14 图片下方有一条白线

vertical-align默认对其方式是基线对齐，基线到底线之间会形成一条白线

解决办法：

1.图片变为块级元素

2.vertical-align:middle