简介

CSS3 模块重要部分包括：

- 选择器
- 框模型
- 背景和边框
- 文本效果
- 2D/3D 转换
- 动画
- 多列布局
- 用户界面

颜色

1.使用预设了值：background-color: red;

2.使用颜色拾取器：

2.1 #c9ffa6;

2.2 rgb(红，绿，蓝)

2.3 hsl(颜色(0~360)，饱和度(0%~100%)，明度(0%~100%))；明度默认是50%,一般建议保留50的值

3.h5中的颜色设置

3.1 rgba(红色，绿色，蓝色，透明度) 透明度：0代表完全透明 1代表完成不透明 不会影响子元素

3.2 hsl(颜色(0~360)，饱和度(0%~100%)，明度(0%~100%)，透明度)

补充样式

opacity:

通过opacity设置透明度:如果设置父容器，那么父容器中的所有子元素也会透明

text-shadow:

文本阴影 text-shadow:offsetX offsetY blur color

box-shadow:

边框阴影：box-shadow:h v blur spread color inset

border-radius:

1.设置一个值：四个角的圆角值都一样

2.设置两个值:第一个值控制左上/右下，第二个值控制右上/左下

3.设置三个值：第一个值控制左上，第二值控制右上/左下，第三个值控制右下

4.设置四个值：左上 右上 右下 左下

5.添加/是用来设置当前个不同方向的半径值 水平x方向/垂直y方向

overflow:

溢出隐藏：overflow: hidden；

溢出转动：overflow: scroll；

盒模型

box-sizing：content-box、border-box

content-box：默认盒模型情况，设置的width属性仅仅是内容的宽度，盒子的最终宽高值在width的基础上在加上padding和border的宽度

border-box：设置的width属性值就是盒子的最终宽度，包含border和padding和内容。如果添加了border和padding，内容的区域会减小。优点，稳固布局。

渐变

背景

1.添加背景颜色：background-color: skyblue;

2.添加背影图片：background-image: url("../images/bg-img.jpg");

如果图片大于容器，那么默认就从容器左上角开始放置

如果图片小于容器，那么图片默认会平铺

3.设置背景平铺：background-repeat: space;

属性值：repeat,no-repeat,repeat-x,repeat-y,round,space

round:会将图片进行缩放之后再平铺

space:图片不会缩放平铺，只是会在图片之间产生相同的间距值

4.设置在滚动容器的背景的行为：background-attachment: local;

fixted，固定：背景图片的位置固定不变，

scroll，local跟随滚动：当滚动容器的时候，背景图片也会跟随滚动

local和scroll的区别：前提是滚动当前容器的内容

local:背景图片会跟随内容一起滚动

scroll:背景图片不会跟随内容一起滚动

5.设置背景图片的大小：background-size: 300px;

1.属性值为像素或百分比（设置百分比，是参照父容器可放置内容区域的百分比），在使用这个属性之前衔确定宽高比与容器的宽高比是否一致，否则会造成图片失真变形；

2.设置contain:按比例调整图片大小，使用图片宽高自适应整个元素的背景区域，使图片全部包含在容器内

图片大于容器：有可能造成容器的空白区域,将图片缩小

图片小于容器：有可能造成容器的空白区域，将图片放大

3.cover:与contain刚好相反，背景图片会按比例缩放自 适应整个背景区域，如果背景区域不足以包含所有背景图片，图片内容会溢出

图片大于容器：等比例缩小，会填满整个背景区域，有可能造成图片的某些区域不可见

图片小于容器：等比例放大，填满整个背景区域，图片有可能造成某个方向上内容的溢出

6.设置背景坐标的原点：background-origin: content-box;

border-box:从border的位置开始填充背景，会与border重叠

padding-box:从padding的位置开始填充背景，会与padding重叠

content-box:从内容的位置开始填充背景

7.设置内容的裁切:设置的是裁切，控制的是显示：background-clip: content-box;

border-box:其实是显示border及以内的内容

padding-box:其实是显示padding及以内的内容

content-box:其实是显示content及以内的内容

边框图片

过渡

通过过渡transition，我们可以在不使用Flash动画或JavaScript的情况下，当元素从一种样式变换为另一种样式时为元素添加效果。

必须元素：

1.规定希望把这个效果添加到哪个CSS属性上，

2.规定效果的时长。

简写语法：

transition：property duration timing-function delay

参数说明：

transition-property：添加所需要效果的样式属性名称

transition-duration：过渡效果的耗时 以秒为单位

transition-timing-function：设置事件函数，控制运动的速度

linear：规定以相同速度开始至结束的过渡效果（cubic-bezier(0,0,1,1)).

ease：规定慢速开始，然后变快，然后慢速结束的过渡效果

ease-in：规定慢速开始的过渡效果

ease-out：规定慢速结束的过渡效果

ease-in-out：规定慢速开始和结束的过渡效果

cubic-bezier(n,n,n,n)：该属性中定义自己的值。值在0~1之间。

transition-delay：过渡效果的延迟。

过渡效果执行完毕之后，默认会还原到原始状态。

单个样式添加过渡效果：transition：left 2s linear 0s;

多个样式添加过渡效果：transition：left 2s linear 0s，background-color 5s linear 0s；

所有样式添加过渡效果：

transition：all 2s；

1.所有样式的过渡效果一样；

2.效率低下，它会查询所有添加的样式，遍历浪费效率；

3.建议：以后不这么用

4.step(4)：可以让过渡效果分为指定的几次来完成

兼容处理：-moz- -webkit- -o-

过渡效果只能产生从某一个值到另一个具体的值的过渡，即从一个状态到另外一个状态。

display：none不可实现过渡

转换

二维转换

通过CSS3 transform转换，我们能够对元素进行移动、缩放、旋转、斜切等操作。

移动 translate：

a.移动时参照元素的左上角；

b.执行完毕之后会恢复到原始位置

transform：translate(tx,ty)。

一个参数代表x方向，　　两个参数代表x/y方向。

也可以使用translateX 或translateY

缩放 scale：

1指不缩放，>1.01放大　<0.99缩小　参照元素的几何中心

2.如果只有一个参数，就代表x和y方向都进行相等比例的缩放

3.如果有两个参数，就代表x/y方向

transform: scale(2,1)//transform:scaleX(0.5);//transform:scaleY(0.5);

旋转 rotate:

设置旋转轴心：transform-origin: left top;

1.x y

2.关键字：left top right bottom center

transform:rotate(-90deg);

斜切 skew：

如果角度为正，则往当前轴的负方向斜切，如果角度为负，则往当前轴的正方向斜切

transform:skew(-30deg);//transform:skewX(30deg);//transform:skewY(30deg);

同时添加多个transform属性值：transform: translate(10px,-250px) rotate(-90deg)；

三维转换

三维移动--3D移动：translate3d(X方向的偏移，Y方向的偏移，Z方向的偏移)

三维缩放：scale3d(x方向上的缩放，y方向的缩放，z方向的缩放)

三维旋转：rotate3d(x,y,z,angle):

x:代表x轴方向上的一个向量值

y:代表y轴方向上的一个向量值

z:代表z轴方向上的一个向量值

transform: rotate3d(1,1,1,330deg);

动画

动画是CSS3中具有颠覆性的特征之一，可通过设置多个节点来精确控制一个或一组动画，常用来实现复杂的动画效果。CSS3中的的动画是关键帧动画，

必要元素：

a、通过关键字@keyframes指定动画序列；自动补间动画，确定两个点，系统会自动计算中间过程。这两个点就称为关键帧。我们可以设置多个关键帧。

b、通过百分比将动画序列分割成多个节点。百分比是指整个动画耗时的百分比，from相当于0%，to相当于100%。

c、在各节点中分别定义各属性

d、通过animation将动画应用于相应元素；

参数说明：

1.animation-name:指定动画名称；

2.animation-duration: 2s；设置动画的总耗时；

3.animation-iteration-count: 1;设置动画的播放次数，默认为1次 可以指定具体的数值，也可以指定infinite(无限次)

4. animation-direction: alternate;设置交替动画 alternate:来回交替

5. animation-delay: 2s;设置动画的延迟

6. animation-fill-mode:设置动画结束时的状态：默认情况下，动画执行完毕之后，会回到原始状态

- forwards:会保留动画结束时的状态，在有延迟的情况下，并不会立刻进行到动画的初始状态
- backwards:不会保留动画结束时的状态，在添加了动画延迟的前提下，如果动画有初始状态，那么会立刻进行到初始状态
- both:会保留动画的结束时状态，在有延迟的情况下也会立刻进入到动画的初始状态

7.animation-timing-function:动画的时间函数。参考过渡参数。

8.animation-play-state: 设置动画的播放状态 paused:暂停 running:播放

多列布局

column-count: 属性设置列的具体个数

column-width: 属性控制列的宽度。原则：取大优先

1.如果人为设置宽度更大，则取更大的值，但是会填充整个屏幕，意味最终的宽度可能也会大于设置的宽度--填充满整个屏幕

2.如果人为设置宽度更小，使用默认计算的宽度

column-gap: 两列之间的缝隙间隔

column-rule: 规定列之间的宽度、样式和颜色

column-span: 规定元素应横跨多少列(n:指定跨n列 all:跨所有列)

Flex 布局

Flex 是 Flexible Box 的缩写，意为"弹性布局"，简便、完整、响应式地实现各种页面布局。http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html

指定flex布局

块级及行内块元素：display:flex， display: -webkit-flex;

行内元素：display: inline-flex;

注意，设为 Flex 布局以后，子元素的float、clear和vertical-align属性将失效。

容器属性

flex-direction

决定主轴的方向（即项目的排列方向）

flex-direction: row | row-reverse | column | column-reverse;

flex-wrap

定义，如果一条轴线排不下，如何换行。

flex-wrap: nowrap | wrap | wrap-reverse;

flex-flow

属性是flex-direction属性和flex-wrap属性的简写形式

flex-flow: <flex-direction> || <flex-wrap>;

justify-content

属性定义了项目在主轴上的对齐方式。

justify-content: flex-start | flex-end | center | space-between | space-around;

align-items

属性定义项目在交叉轴上如何对齐。

align-items: flex-start | flex-end | center | baseline | stretch;

baseline: 项目的第一行文字的基线对齐。

stretch（默认值）：如果项目未设置高度或设为auto，将占满整个容器的高度

align-content

属性定义了多根轴线的对齐方式。如果项目只有一根轴线，该属性不起作用。

align-content: flex-start | flex-end | center | space-between | space-around | stretch;

项目属性

order

属性定义项目的排列顺序。数值越小，排列越靠前，默认为0。

order: <integer>;

flex-grow

属性定义项目的放大比例，默认为0，即如果存在剩余空间，也不放大

flex-grow: <number>; /* default 0 */

如果所有项目的flex-grow属性都为1，则它们将等分剩余空间（如果有的话）。如果一个项目的flex-grow属性为2，其他项目都为1，则前者占据的剩余空间将比其他项多一倍。

flex-shrink

属性定义了项目的缩小比例，默认为1，即如果空间不足，该项目将缩小

flex-shrink: <number>; /* default 1 */

如果所有项目的flex-shrink属性都为1，当空间不足时，都将等比例缩小。如果一个项目的flex-shrink属性为0，其他项目都为1，则空间不足时，前者不缩小。

flex-basis

属性定义了在分配多余空间之前，项目占据的主轴空间（main size）。浏览器根据这个属性，计算主轴是否有多余空间。它的默认值为auto，即项目的本来大小。

flex-basis: <length> | auto; /* default auto */

它可以设为跟width或height属性一样的值（比如350px），则项目将占据固定空间。

flex

属性是flex-grow, flex-shrink 和 flex-basis的简写，默认值为0 1 auto。后两个属性可选。

flex: none | [<'flex-grow'> <'flex-shrink'>? || <'flex-basis'>]

该属性有两个快捷值：auto (1 1 auto) 和 none (0 0 auto)。

建议优先使用这个属性，而不是单独写三个分离的属性，因为浏览器会推算相关值。

align-self属性允许单个项目有与其他项目不一样的对齐方式，可覆盖align-items属性。默认值为auto，表示继承父元素的align-items属性，如果没有父元素，则等同于stretch。

align-self: auto | flex-start | flex-end | center | baseline | stretch;

媒体查询

https://www.runoob.com/cssref/css3-pr-mediaquery.html

使用 @media 查询，你可以针对不同的媒体类型定义不同的样式。

@media 可以针对不同的屏幕尺寸设置不同的样式，特别是如果你需要设置设计响应式的页面，@media 是非常有用的。

当你重置浏览器大小的过程中，页面也会根据浏览器的宽度和高度重新渲染页面。

语法：

1.@media *mediatype* and|not|only *(media feature)* { *CSS-Code;* }

2.<link rel="stylesheet" media="*mediatype* and|not|only (*media feature*)" href="*mystylesheet.css*">

常用格式：

@media only screen and (max-width:600px) { *CSS-Code;* }