1一. CSS3边框

边框属性：

border-radius：创建圆角边框

box-Shadow：附加一个或者多个下拉框的阴影

border-image（不支持IE）：使用图像创建一个边框

CSS3圆角

可以使用CSS3 圆角制作器来制作一个css3圆角，可以直接生成代码。

指定border-radius的每个圆角：

如果只指定一个值，则生成四个圆角；若在每个角上指定，则使用如下规则：

四个值：四个值分别指定左上角、右上角、右下角、左下角。

三个值：第一个值为左上角，第二个值为右上角和左下角，第三个值为右下角。

两个值：第一个值为左上角和右下角，第二个值为右上角和左下角。

一个值：四个圆角值相同。

创建椭圆边角：

border-radius： 50px/15px;

border-radius: 50%;

圆角属性：

border-radius

所有四个边角属性的缩写

border-top-left-radius

左上角的弧度

border-top-right-radius

右上角的弧度

border-bottom-left-radius

左下角的弧度

border-bottom-right-radius

右下角的弧度

1. CSS3背景

四种背景属性：

background-image：添加背景图片。不同的背景图像和图像用逗号隔开。

background-size：指定背景图像的大小，可指定像素或者百分比的大小，是相对于父元素的宽度和高度的百分比的大小。

background-origin：指定了背景图像的位置区域

background-clip：背景裁剪属性是从指定位置开始绘制。

1. CSS3渐变

渐变可以让我们在两个或者多个颜色之间显示平稳的过渡。

两种类型的渐变：

Linear Gradients：线性渐变-向下/向上/向左/向右/对角方向

Radial Gradients：径向渐变，由它们的中心定义

线性渐变：

语法：background：linear-gradient(direction, color-stop1, color-stop2, …);

线性渐变默认情况是从上到下。

从左到右的线性渐变：

#grad {

background: -webkit-linear-gradient(left, red, blue);

background: -o-liner-gradient(right, red, blue);

background: -moz-linear-gradient(right, red, blue);

background: linear-gradient(to right, red, blue);/\*标准语法\*/

}

从左上角开始到右下角的渐变：

设置方向为：left top/bottom right/to bottom right(标准语法下)

使用角度对方向进行设置：

这里的角度是指水平线和渐变线之间的角度。So 0deg将创建一个从下到上的渐变，90deg将创建一个从左到右的渐变。

使用透明度：

支持透明度，用于创建减弱变淡的效果。

为了添加透明度，使用rgba()函数来定义颜色结点。rgba()函数中的最后一个参数可以是从0到1的值，它定义了颜色的透明度：0表示完全透明，1表示完全不透明。

重复的线性渐变：

repeating-linear-gradient()函数用于重复的线性渐变。

径向渐变：

我们可以指定渐变的中心、形状（圆形或者椭圆形）、大小，默认情况下，渐变的中心是center（表示在中心点），渐变的形状是ellipse（表示椭圆形），渐变的大小是farthest-corner（表示到最远的角落）。

语法: background: radial-gradient(center, shape size, start-color, …, last-color);

颜色结点分布不均匀的径向渐变：

#grad{

background: -webkit-radial-gradient(red 5%, green 15%, blue 60%);

}

Size参数定义了渐变的大小，可以是以下四个值：

closest-side/farthest-side/closest-corner/farthest-corner

重复的径向渐变：

Reperting-radial-gradient()函数用于重复径向渐变。

CSS3文本效果

包含属性：

text-shadow：适用于文本阴影，可以指定是水平阴影还是垂直阴影、模糊的距离以及阴影的颜色。

box-shadow：适用于盒子阴影。同上。可以在::after和::before两个伪元素中添加阴影效果。

text-overflow：文本溢出属性指定应向用户如何显示溢出内容

有两个属性：ellipsis（以省略号的形式隐藏内容）和clip（直接截断内容）

word-wrap：强制换行，但是单词中间不换行

word-break：单词可进行拆分然后换行

CSS3字体

@font-face规则，若使用，首先定义字体的名称，然后指向该文字字体。

通过font-family属性来引用字体的名称。

列举出所有的字体描述和里面的@font-face规则定义

描述符

值

描述

font-family

name

必需，规定字体的名称。

src

URL

必需，定义字体文件的URL

font-strench

normal

condensed

ultra-condensed

extra-condensed

semi-condensed

expanded

semi-expanded

extra-expanded

ultra-expanded

可选，定义如何拉伸字体，默认值是nomal

font-style

nomal

Italic

oblique

可选，定义字体的样式

font-weight

nomal

bold

100

200

......

900

可选，定义字体的粗细

unicode-range

unicode-range

可选，定义字体支持的UNICODE字符范围

CSS3 2D转换

2D变换方法：

translate()：平移

rotate()：在一个给定度数顺时针旋转的元素

scale()

skew()

matrix()

CSS3过渡

为了添加某种效果可以从一个样式转变到另一个的时候，无需使用flash动画或JavaScript，只需要鼠标移动到上面就可以。

我们要实现这样的效果，必须规定两项内容：

指定要添加效果的CSS属性和效果的持续时间

所有的过渡属性：

transition：简写属性，用于在一个属性中设置四个过渡属性

transition-property：规定应用过渡的CSS属性的名称

transition-duration：定义过渡效果花费的时间，默认是0

transition-timing-function：规定过渡效果的时间曲线，默认是‘ease‘

transition-delay：规定过渡时间何时开始，默认是0

div {

transition-property: width;

transition-duration: 1s;

transition-timing-function: linear;

transition-delay: 2s;

}//等价于div{ transition: width 1s linear 2s; }

CSS3动画

创建动画，可以取代许多网页动画图像，Flash动画和JavaScript.

CSS3@keyframes规则

该规则是创建动画。@keyframes规则内指定一个CSS样式和动画将逐步从目前的样式更改为新的样式。

当@keyframes创建动画时，把它绑定到一个选择器，否则不会有任何效果。

指定至少两个CSS3的动画属性绑定向一个选择器：规定动画的名称和动画的时长。

div {

width: 100px;

height: 100px;

background: red;

animation: myfirst 5s;

}

@keyframes myfirst {

from { background: red;}

to { backgroung: yellow;}

}

我们可以改变任意多的样式任意多的次数。请用百分比来规定变化发生的时间，或用关键词“from”和“to”，等同于0%和100%。0%是动画的开始，100%是动画的完成。

所有的动画属性：

CSS3多列

多列属性：

column-count：指定元素的列数应分为几列。

div {

-moz-column-count:3;/\*Firefox\*/

}

column-gap：指定的列之间的差距

column-rule：设置列之间的宽度、样式和颜色

column-rule-color：指定的列之间颜色规则

column-rule-style：指定的列之间的样式规则

column-rule-width：指定的列之间的宽度规则

column-width：指定列的宽度

column-span：指定一个元素应该横跨多少列

columns：缩写属性设置列宽和列数

column-fill：指定如何填充列

CSS3用户界面

新增加了一些新的用户界面特性来调整元素尺寸、框尺寸和外边框。

用户界面属性：

resize：指定一个元素是否应该由用户去调整大小

box-sizing：允许以确切的方式定义适应某个区域的具体内容

outline-offset：对轮廓进行偏移，并在超出边框边缘的位置绘制轮廓。轮廓与边框不同，轮廓不占用空间，而且可能是非矩形。

CSS3盒子模型

所有HTML元素可以看作盒子。CSS盒模型本质上是一个盒子，封装周围的HTML元素，包括：边距、边框、填充和实际内容。

不同部分的说明：

Margin：外边距，清除边框外的区域，外边距是透明的。

Border：边框，围绕在内边距和内容外的边框。

Padding：内边距，清除内容周围的区域，内边距是透明的。

Content：盒子的内容，显示文本和图像。

总元素的宽度 = 宽度 + 左填充（padding） + 右填充 + 左边框 + 右边框 + 左边距 + 右边距

两种盒模型的比较：

盒模型分为W3C的标准盒模型和IE盒模型。

标准盒模型的宽度就是content的宽度，高度是content的高度，而IE盒模型的宽度是content+padding+border的总和。

切换盒模型的方法：设置box-sizing属性。

box-sizing: content-box //W3C盒子模型

box-sizing:border-box //IE盒子模型

CSS3弹性盒子（Flex Box）

说明：弹性盒子模型这一重点知识阮一峰老师讲的很详细，图文结合很容易理解，文章开头我附加了阮一峰老师这部分内容的链接。除了一些基本的语法之外，还有很多实用的实例。一定要掌握这一部分的知识点，面试中会经常问到。

是一种当页面需要适应不同的屏幕大小以及设备类型时确保元素拥有恰当的行为的布局方式。

弹性盒子由弹性容器和弹性子元素组成。弹性容器通过设置display属性的值为flex或inline-flex（行内元素设置为Flex布局）将其定义为弹性容器。

注：设为了Flex布局以后，子元素的float、clear和virtical-align属性将失效。

弹性容器内包含一个或多个弹性子元素。

注意：弹性容器外及弹性子元素内是正常渲染的，弹性盒子只定义了弹性元素的如何在弹性容器内布局。弹性子元素通常在弹性盒子内一行显示，默认情况每个容器只有一行。

修改排列方式：rtl(right to left)

body { direction: rtl; }

弹性容器上的属性共有6个：

flex-direction

flex-wrap

flex-flow

justify-content

align-items

align-content

①flex-direcion：决定了主轴的方向（即弹性子元素在父容器中的排列方向）。

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

row：横向从左到右排列。默认的排列方式。

row-reverse：反转横向排列（右对齐，从后往前排，最后一项排在最前面）

column：纵向排列。

column-reverse：反转纵向排列，从后往前排，最后一项排在最上面。

②flex-wrap：默认情况下，项目都排在一条线上，该属性定义如果一条轴线排不下，如何换行。

语法：flex-wrap: nowrap | wrap | wrap-reverse

③flex-flow：该属性是flex-direction属性和flex-wrap属性的简写形式，默认值是row nowrap。

④justify-content：定义了项目在主轴上的对齐方式，把弹性项沿着弹性容器的主轴线对齐。

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

flex-start：左对齐

flex-end：右对齐

center：居中

space-between：两端对齐，项目之间的间隔都是相等的

space-around：每个项目两侧的间隔相等，所以项目之间的间隔比项目与边框的间隔大一倍。

⑤align-items：定义项目在交叉轴上如何对齐。

语法：align-items: flex-start | flex-end | center | baseline | strench

baseline：项目的第一行文字的基线对齐

strench（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。

⑥align-content：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。

语法：align-content: flex-start | flex-end | center | space-between | space-around | strench(默认值)

弹性子元素上的6个属性：

order

flex-grow

flex-shrink

flex-basis

flex

align-self

①order :定义弹性子元素的排列顺序。数值越小，排列越靠前，默认为0。

语法：order: <integer>

②flex-grow: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。

语法：flex-grow: <number>

③flex-shrink:定义项目的缩小比例，默认为1，即如果空间不足，该项目将缩小。负值对该属性无效。

语法：flex-shrink: <number>

④flex-basis：定义了在分配多余空间之前，项目占据的主轴空间。默认值为auto

⑤flex属性是flex-grow、flex-shrink、flex-basis的简写，默认值是 0 1 auto。建议优先使用这个属性。该属性有两个快捷值： auto(1 1 auto) 和none(0 0 auto)。

⑥align-self：允许单个弹性子元素有与其他子元素不一样的对齐方式。可覆盖align-items属性，默认值为auto，表示继承弹性容器的align-items属性，如果没有父元素，则等同于strench。

CSS3多媒体查询

CSS中我们可以根据不同的媒体类型（包括显示器、便携设备、电视机等）来设置不同的样式规则。

在CSS3中根据设置自适应显示。

媒体查询可用于检测很多事情：

Viewport（视图）的宽度和高度

设备的宽度和高度

朝向（智能手机横屏，竖屏）

分辨率

---------------------

作者：瘦肉\_

来源：CSDN

原文：https://blog.csdn.net/weixin\_42734488/article/details/81429549

版权声明：本文为博主原创文章，转载请附上博文链接！