CSS3属性笔记

属性选择器

* E[attr]只使用属性名，但没有确定任何属性值 eg: input[type]{}
* E[type="text"]指定属性名，并指定了该属性的属性值  eg: input[type=“text”]{}
* E[attr~="value"]指定属性名，并且具有属性值，此属性值是一个词列表，并且以空格隔开，其中词列表中包含了一个value词，而且等号前面的“～”不能不写(属性中包含这个value的都会选中)
* E[attr^="value"]指定了属性名，并且有属性值，属性值是以value开头的（以value开头的都会被选）
* E[attr$="value"]指定了属性名，并且有属性值，而且属性值是以value结束的
* E[attr\*="value"]指定了属性名，并且有属性值，而且属值中包含了value
* E[attr |="value"]指定了属性名，并且属性值是value或者以“value-”开头的值（比如说zh-cn）

        结构性伪类

* E:nth-child(n)  表示E父元素中的第n个字节点
* ·p:nth-child(odd){background:red}/\*匹配奇数行\*/
* ·p:nth-child(even){background:red}/\*匹配偶数行\*/
* ·p:nth-child(2n){background:red}
* E:nth-last-child(n) 表示E父元素中的第n个字节点，从后向前计算
* E:nth-of-type(n)  表示E父元素中的第n个字节点，且类型为E
* E:nth-last-of-type(n)表示E父元素中的第n个字节点，且类型为E,从后向前计算
* E:empty 表示E元素中没有子节点。注意：子节点包含文本节点

       body p:nth-child(2)  找到父级第二个子节点 并且子节点还得是p标签（文本节点）

       body p:nth-of-type(2)  找到父级第二个p元素

* E:first-child 表示E元素中的第一个子节点
* E:last-child 表示E元素中的最后一个子节点
* E:first-of-type 表示E父元素中的第一个子节点且节点类型是E的
* E:last-of-type 表示E父元素中的最后一个子节点且节点类型是E的
* E:only-child表示E元素中只有一个子节点。注意：子节点不包含文本节点
* E:only-of-type 表示E的父元素中只有一个子节点，且这个唯一的子节点的类型必须是E。注意:子节点不包含文本节点

伪类

* E:target 表示当前的URL片段的元素类型，这个元素必须是E
* E:disabled 表示不可点击的表单控件
* E:enabled 表示可点击的表单控件
* E:checked 表示已选中的checkbox或radio
* E:first-line 表示E元素中的第一行
* E:first-letter 表示E元素中的第一个字符
* E::selection表示E元素在用户选中文字
* E::before 生成内容在E元素前
* E::after 生成内容在E元素后
* E:not(s) 表示E元素不被匹配
* E~F表示E元素毗邻的F元素
* Content 属性

**rgba颜色模式**

* r  Red       红     0-255
* g  Green   绿     0-255
* b  Blue     蓝      0-255
* a  Alpha   透明    0-1

实例: 背景透明，文字不透明

问题：注意边框颜色透明有问题

**文字阴影**

**text-shadow：x y blur color, …**

**参数**

* **x** **横向偏移**
* **y** **纵向偏移**
* **blur** **模糊距离**
* **color** **阴影颜色**

**文本阴影如果加很多层，会很卡很卡很卡**

文字阴影应用

        最简单用法

            text-shadow:2px 2px 4px black

**阴影叠加**

            text-shadow:2px 2px 0px red, 2px 2px 4px green;

        先渲染后面的，再渲染前面的

        几个好玩的例子

        层叠： color:red; font-size:100px; font-weight:bold; text-shadow:2px 2px 0px white, 4px 4px 0px red;

光晕：color:white; font-size:100px; text-shadow:0 0 10px #fff, 0 0 20px #fff, 0 0 30px #fff, 0 0 40px #ff00de, 0 0 70px #ff00de, 0 0 80px #ff00de, 0 0 100px #ff00de, 0 0 150px #ff00de;

火焰文字：text-shadow: 0 0 20px #fefcc9, 10px -10px 30px #feec85, -20px -20px 40px #ffae34, 20px -40px 50px #ec760c, -20px -60px 60px #cd4606, 0 -80px 70px #973716, 10px -90px 80px #451b0e; font-family:Verdana, Geneva, sans-serif; font-size:100px; font-weight:bold; color:white;

**文字描边：**

**-webkit-text-stroke:宽度 颜色**

**新增文本功能：**

**Direction** **定义文字排列方式(全兼容)**

* **Rtl** **从右向左排列**
* **Ltr** **从右向左排列**
* **注意要配合unicode-bidi** **一块使用**

**Text-overflow** **定义省略文本的处理方式**

* **clip** **无省略号**
* **Ellipsis** **省略号** **(注意配合overflow:hidden和white-space:nowrap一块使用)**

**弹性盒模型：**

注意在使用弹性盒模型的时候 父元素必须要加display:box 或 display:inline-box（要加**-webkit-**）

**Box-orient** **定义盒模型的布局方向**

            Horizontal 水平显示

            vertical 垂直方向

**box-direction 元素排列顺序**

            Normal 正序

            Reverse 反序

**box-ordinal-group 设置元素的具体位置**

**Box-flex 定义盒子的弹性空间**

            子元素的尺寸=盒子的尺寸\*子元素的box-flex属性值 / 所有子元素的box-flex属性值的和

**box-pack** **对盒子富裕的空间进行管理**

            Start 所有子元素在盒子左侧显示，富裕空间在右侧

            End 所有子元素在盒子右侧显示，富裕空间在左侧

            Center 所有子元素居中

            Justify 富余空间在子元素之间平均分布

**box-align** **在垂直方向上对元素的位置进行管理**

            Star 所有子元素在据顶

            End 所有子元素在据底

            Center 所有子元素居中

**盒模型阴影：**

**用法**

                box-shadow:[inset] x y blur [spread] color

                参数

                    nset：投影方式

                            inset：内投影

                            不给：外投影

                    x、y：阴影偏移

                    blur：模糊半径

                    spread：扩展阴影半径

                            先扩展原有形状，再开始画阴影

            color

**其他盒模型新增属性：**

**box-reflect** **倒影**

 direction  方向     above|below|left|right;

                    距离

                    渐变（可选）      **-webkit-linear-gradient(red 0,blue 100%);**

**resize** **自由缩放**

Both 水平垂直都可以缩放

                    Horizontal 只有水平方向可以缩放

                    Vertical 只有垂直方向可以缩放

**注意：一定要配合overflow:auto** **一块使用只有水平方向可以缩放**

**box-sizing 盒模型解析模式**

                    Content-box  标准盒模型 width/height=border+padding+content

            Border-box 怪异盒模型 width/height=content

**CSS3分栏布局：（加-webkit-）**

**column-width** **栏目宽度**

**column-count 栏目列数**

**column-gap   栏目距离**

**column-rule  栏目间隔线**

**CSS3响应式布局：**

**媒体类型**

                    all 所有媒体

                    braille 盲文触觉设备

                    embossed 盲文打印机

                    print 手持设备

                    projection 打印预览

                    screen 彩屏设备

                    speech '听觉'类似的媒体类型

                    tty 不适用像素的设备

            tv  电视

**关键字**

                    and

                    not      not关键字是用来排除某种制定的媒体类型

            only     only用来定某种特定的媒体类型

**媒体特性**

                    (max-width:600px)

                    (max-device-width: 480px)  设备输出宽度

                    (orientation:portrait)  竖屏

                    (orientation:landscape)  横屏

                    (-webkit-min-device-pixel-ratio: 2) 像素比

**样式引入**

                    <link rel="stylesheet" type="text/css" href="../css/print.css" media="print" />

                    @import url("css/reset.css") screen;

                    @media screen{

                          选择器{

                               属性：属性值；

                             }

               }

**移动端meta：**

**<meta name="viewport" content="" />**

                    width [pixel\_value | device-height]

                    height [pixel\_value | device-height]

                    user-scalable 是否允许缩放 （no||yes）

                    initial-scale 初始比例

                    minimum-scale 允许缩放的最小比例

                    maximum-scale 允许缩放的最大比例

                 target-densitydpi [dpi\_value | device-dpi | high-dpi | medium-dpi | low-dpi]

**圆角：**

            border-radius: 1-4个数字 / 1-4个数字

                前面是水平，后面是垂直

                不给“/”则水平和垂直一样

                        border-radius: 10px/5px;

**参数**

                各种长度单位都可以：px，%，…

                %有时很方便

                        但宽高不一致时不太好

**用法**

        border-radius: 一样

        border-radius: 左上&右下    右上&左下

        border-radius: 左上    右上&左下    右下

        border-radius: 左上    右上    右下    左下

**边框：（-webkit-）**

    边框图片 border-image

            border-image-sourceg 引入图片

            border-image-slice 切割图片

            border-image-width 边框宽度

            border-image-repeat 图片的排列方式

                round 平铺,repeat 重复,stretch拉伸

    边框颜色 border-colors

**线性渐变：**

**线性渐变格式**

        linear-gradient([<起点> || <角度>,]? <点>, <点>…)

        只能用在背景上

        IE                        filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffffff',endColorstr='#ff0000',GradientType='1');

**参数**

        起点：从什么方向开始渐变  默认：top

                left、top、left top

        角度：从什么角度开始渐变

                xxx deg的形式

        点：渐变点的颜色和位置

 black 50%，位置可选

**线性渐变实例**

        最简单

            red, green

            从上到下

        起点位置

            left top, red, green

            30deg, red, green

                逆时针

**repeating-linear-gradient**

**线性渐变实例(2)**

        加入点的位置

            top, red 40%, green 60%

            top, red 50%, green 50%

                同一个位置两个点——直接跳变

            也可以用px

**配合rgba**

                top, rgba(255,255,255,1), rgba(255,255,255,0)

**加入背景图片**

                background: -webkit-linear-gradient (top, rgba(255,255,255,1) 30%, rgba(255,255,255,0)), url(a.gif)

**径向渐变**

radial-gradient([<起点>]? [<形状> || <大小>,]? <点>, <点>…);

    起点：可以是关键字（left,top,right,bottom），具体数值或百分比

    形状： ellipse、circle

    大小 :具体数值或百分比，也可以是关键字（最近端，最近角，最远端，最远角，包含或覆盖 (closest-side, closest-corner, farthest-side, farthest-corner, contain or cover)）;

**注意firefox目前只支持关键字**

radial-gradient([<起点>]? [<形状> || <大小>,]? <点>, <点>…);

    起点：可以是关键字（left,top,right,bottom），具体数值或百分比

    形状： ellipse、circle

    大小 :具体数值或百分比，也可以是关键字（最近端，最近角，最远端，最远角，包含或覆盖 (closest-side, closest-corner, farthest-side, farthest-corner, contain or cover)）;

**背景：**

多背景

    逗号分开

            background: url(a.jpg) 0 0, url(b.jpg) 0 100%;

背景尺寸

    background-size:x y

          background-size:100% 100%

          Cover 放大

                  Contain 缩小

    background-origin ： border | padding | content

        border-box： 从border区域开始显示背景。

        padding-box： 从padding区域开始显示背景。

        content-box： 从content区域开始显示背景。

    background-clip

        border： 从border区域向外裁剪背景。

        padding： 从padding区域向外裁剪背景。

        content： 从content区域向外裁剪背景。

        no-clip： 从border区域向外裁剪背景。

**遮罩：**

    Mask-image

    Mask-position

Mask-repeat

**Transition过渡:**

transition-property  要运动的样式  （all || [attr] || none）

transition-duration 运动时间

transition-delay 延迟时间

transition-timing-function 运动形式

    ease：（逐渐变慢）默认值

    linear：（匀速）

    ease-in：(加速)

    ease-out：（减速）

    ease-in-out：（先加速后减速）

 cubic-bezier 贝塞尔曲线（ x1, y1, x2, y2 ）

**过渡完成事件**

Webkit内核： obj.addEventListener('webkitTransitionEnd',function(){},false);

firefox: obj.addEventListener('transitionend',function(){},false);

**2D变换：**

    transform

            rotate()  旋转函数 取值度数

                deg  度数

                Transform-origin 旋转的基点

    skew() 倾斜函数 取值度数

            skewX()

            skewY()

    scale() 缩放函数 取值 正数、负数和小数

            scaleX()

            scaleY()

    translate() 位移函数

            translateX()

            translateY()

**Transform 执行顺序问题 — 后写先执行**

**matrix(a,b,c,d,e,f) 矩阵函数**

        通过矩阵实现**缩放**

              x轴缩放 a=x\*a    c=x\*c

              y轴缩放 b=y\*b   d=y\*d

        通过矩阵实现**位移**

              x轴位移: e=e+x

              y轴位移: f=f+y

        通过矩阵实现**倾斜**

              x轴倾斜: c=Math.tan(xDeg/180\*Math.PI)

              y轴倾斜: b=Math.tan(yDeg/180\*Math.PI)

**matrix(a,b,c,d,e,f) 矩阵函数**

              通过矩阵实现旋转

                    a=Math.cos(deg/180\*Math.PI);

                    b=Math.sin(deg/180\*Math.PI);

                    c=-Math.sin(deg/180\*Math.PI);

                    d=Math.cos(deg/180\*Math.PI);

**变换兼容IE9以下IE版本只能通过矩阵来实现**

                  filter: progid:DXImageTransform.Microsoft.Matrix( M11= 1, M12= 0, M21= 0 , M22=1,SizingMethod='auto expand');

                  IE下的矩阵没有E和F两个参数 M11==a; M12==c; M21==b; M22==d

**3D变换：**

    transform-style（preserve-3d） 建立3D空间

            Perspective 景深

            Perspective- origin 景深基点

            Transform 新增函数

                rotateX()

                rotateY()

                rotateZ()

                translateZ()

        scaleZ()

**animation：**

    关键帧——keyFrames

                类似于flash

                    只需指明两个状态，之间的过程由计算机自动计算

            关键帧的时间单位

                数字：0%、25%、100%等

                字符：from(0%)、to(100%)

            格式(1)

                    @keyframes 动画名称

                            {动画状}

            格式(2)

                    @keyframes  miaov\_test

                            {from { background:red; }

                              to { background:green; } }

                    可以只有to

        调用动画：

            调用的标签(#div1、xxx:hover之类的)

                必要属性

                    animation-name  动画名称（关键帧名称）

                    animation-duration  动画持续时间

                例如：

                        -webkit-animation-name: ‘miaov';

                        -webkit-animation-duration: 4s;

                例子：进度条

                animation-play-state 播放状态（running 播放 和paused 暂停 ）

                可选属性

                    animation-timing-function  动画运动形式

                        linear  匀速。

                        ease  缓冲。

                        ease-in  由慢到快。

                        ease-out  由快到慢。

                        ease-in-out  由慢到快再到慢。

                        cubic-bezier(number, number, number, number)：  特定的贝塞尔曲线类型，4个数值需在[0, 1]区间内

                可选属性(2)

                    animation-delay  动画延迟

                            只是第一次

                    animation-iteration-count  重复次数

                            infinite为无限次

                    animation-direction  播放前重置

                            动画是否重置后再开始播放

                            alternate  动画直接从上一次停止的位置开始执行

                            normal  动画第二次直接跳到0%的状态开始执行

           与JS结合：

                    通过class

                            在class里加入animation的各种属性

                             直接给元素加-webkit-animation-xxx样式

                    animation的问题

                            写起来麻烦

                            没法动态改变目标点位置

                    obj.addEventListener('webkitAnimationEnd', function (){}, false);

                    实例1：无缝滚动

CSS3边框

border-radius：25px;  圆角

box-shadow :(10px 10px 10px #888888) 阴影

border-image:边界图片      (不支持ie浏览器，其他浏览器，一般要加前缀-webkit- 欧朋浏览器加-o-)