注意事项：

导航栏一般不直接用a，而是用li包含链接（li+a）的做法。

设置字体时中文名称用unicode编码来代替，如黑体（\9ED1\4F53）、宋体（\5B8B\4F53）、微软雅黑（\5FAF\8F6F\96C5\9ED1）

CSS中字符串用单引号

a包含有宽高的元素时，需转换为块级元素

定位可以拉伸元素

width：100%和auto：width为100%的元素占整个页面宽时，若指定padding，会出现横向滚动条。而width为auto的元素不会

* 1. 各网页元素会有自带的内外边距，可用\*选择器设置margin和padding为0来清除。
  2. 行内元素为了照顾兼容性，尽量只设置左右内外边距，而不设置上下内外边距。

# 综述

1. CSS（层叠样式表）：实现了结构（HTML）与样式（CSS）分离
2. 网页布局整体思路：
   1. 确定版心；分析行模块，以及行模块中的列模块（页面布局第一准则）；列模块常用浮动，先确定大小再确定位置（页面布局第二准则）；先有结构再有样式
3. CSS引入方式：
   1. 内部样式表（放入style标签内，style标签可以放在文档的任意地方，但一般放在head标签中）控制范围为整个页面
   2. 行内样式表（在元素标签内部的sytle属性内设置CSS样式）只控制当前的标签样式
      1. <div sytle=”color: red; font-size: 12px; ”>
   3. 外部样式表（新建css文件，将CSS代码放入其中，再在html文件中使用link标签引入）
      1. <link rel=”stylesheet” herf=”css文件路径”>
4. CSS三大特性：
   1. 层叠性：给相同的选择器的相同样式设置不同属性会覆盖（后执行的代码覆盖先执行的代码）
   2. 继承性：子标签会继承父标签的部分属性（text-，font-，line-和color）。如：p从div中继承color和font-size
   3. 优先级：不同选择器给同一元素的相同样式设置不同属性，按优先级设置。
      1. 单选择器的优先级：
         1. !important > 行内样式style=”” > ID选择器 > 类选择器，伪类选择器，属性选择器 > 元素选择器、伪元素选择器 > 继承或\*
      2. 复合选择器的优先级：
         1. 分别统计该复合选择器中的行内样式、ID选择器、类选择器、元素选择器的个数。按优先级顺序依次比较数量，数量多的优先级高。
5. CSS属性书写顺序：
   1. 1、布局定位属性：display / position / float / clear / visibility / overflow
   2. 2、自身属性：width / height / margin / padding / border / background
   3. 3、文本属性：color / font / text-decoration / text-align / vertical-align / white-space / break-word
   4. 4、其他属性（CSS3）：content / cursor / border-radius / box-shadow / text-shadow / background：linear-gradient。。。
6. CSS初始化：
   1. 所有标签内外边距清除、em与i的文字不倾斜、去除li的小圆点、设置图片边框为0（照顾兼容性）及vertical-align：middle（消除图片底侧空白缝隙）、设置按钮cursor：pointer、设置链接颜色并取消下划线、设置body文字样式及抗锯齿（-webkit-font-smoothing：antialiased；）、清除浮动。
7. 元素显示模式：
   1. 块元素：
      1. 独占一行；高度、宽度、内外边距可控制；宽度默认是容器（父级宽度）的100%；
      2. 是一个容器及盒子，可以放行内元素或块级元素（文字类标签如p、h1-6内不能放块级元素）。如<h1-6><p><div><ul><ol><li>
   2. 行内元素：
      1. 相邻行内元素在一行上，一行可以显示多个；高宽无法直接设置；默认宽度为它本身内容的宽度；行内元素只能容纳文本或其他行内元素
      2. 链接内不能再放链接，但特殊情况下可以放块级元素
   3. 行内块元素：同时具有行内元素和块元素的特点
      1. 可和相邻的行内块元素在一行上，但他们之间会有空白缝隙，一行可以显示多个；
      2. 默认宽度为它本身内容的宽度；高度，行高，内外边距可控制。如img、input、td
   4. 元素显示模式的转换：
      1. 转换为块元素： display：block；
      2. 转换为行内元素： display：inline；
      3. 转换为行内块元素： display：inline-block；
8. 块级格式化上下文（Block Formatting Contexts，BFC)：是一个环境，里面的元素不会影响外面的元素
   1. 布局规则：只要元素满足下面任一条件即可触发 BFC 特性
      1. body 根元素 浮动元素：float 除 none 以外的值
      2. 绝对定位元素：position (absolute、fixed) display 为 inline-block、table、inline-table及各种table-...、flex
      3. overflow 不为 visible 以外的值 (hidden、auto、scroll) 等等。。。
   2. BFC特性：
      1. 多个BFC之间不会发生外边距合并、BFC可以包含浮动的元素且高度不塌陷（用于清除浮动）、BFC不与浮动元素重叠（两栏自适应布局）

# 语法

* 1. 选择器：选择器 {属性：值；...} CSS中的值一般情况不加引号
     1. 基础选择器：
        1. 标签选择器：将标签名作为选择器，选择所有该标签
        2. 类选择器：将 .类名 作为选择器，选择所有class属性为类名的标签。标签可以有多个类名
        3. id选择器：将 #id名 作为选择器，选择**第一个**id属性为id名的标签
        4. 通配符选择器：将 \* 作为选择器，选择页面中所有元素
     2. 复合选择器：
        1. 后代选择器：父标签 标签0 标签1... 作为选择器，选择所有符合条件（前者为后者的父标签）的标签
        2. 子选择器：元素1 > 元素2 选择元素1中的所有元素2
        3. 并集选择器：元素1，元素2，... 选择元素1和元素2
        4. 交集选择器：h3.class 选择类名为class的h3标签
        5. 相邻兄弟选择器：h1+h2 若h1和h2为兄弟，选择所有h1后面的第一个h2标签
        6. 后续兄弟选择器：h1~h2 若h1和h2为兄弟，选择所有h1后面的所有h2标签
        7. 伪类选择器：
           1. 链接伪类选择器：

a：link 选择所有未被访问的链接

a：visited 选择所有已被访问的链接

a：hover 选择鼠标指针位于其上的链接

a：active 选择活动链接（鼠标按下未弹起的链接）

为了确保生效，按照上述自上而下的顺序声明，一般只需改变a和a：hover的样式

* + - * 1. ：focus伪类选择器：选取焦点，一般只有input才能获取

input：focus 选择获得输入光标的表单元素

* 1. 变量：（完全不兼容ie）
     1. html｛--color：#bfa；｝ 创建了一个名为color的变量，值为#bfa
     2. color：var（--color） ； 引用了变量
  2. 函数：
     1. width：calc（1000px/2）； 计算，完全不兼容ie

# 属性

1. 继承属性：某属性：inherit 规定应该从父元素继承该属性的值
2. 字体属性：
   1. 字体系列：font-family：“Microsoft yahei”, Arial , “宋体”； 字体中自带空格的加引号，优先使用前面的字体，若无法显示则下一个
   2. 字体大小：font-size：20px；
   3. 字体粗细：font-weight：normal（正常显示，400）| bold（粗体，700）| bolder | lighter | number（100、200。。。900）；
   4. 文字样式：font-style：normal（默认值）| italic（斜体）
   5. 字体复合属性：font：font-style font-weight font-size/-height font-family 不设置的属性可忽略，但必须保留size和family属性
3. 段落属性：
   1. 对齐：text-align: center; 将子元素的行内元素或行内块元素水平居中
   2. 首行缩进：text-indent: 2em; 缩进2字符
   3. 修饰：text-decoration: none | underline | line-through； 无修饰/增加下划线/增加删除线
   4. 文字行高：line-height：40px | 1.5； 若不带单位，表示为文字大小的1.5倍。可以实现单行文字垂直居中
   5. 换行：
      1. word-break：normal | break-all（允许单词内换行） | keep-all
      2. word-wrap：normal | break-word（允许长单词内部换行） 仅在遇到比容器宽度更长的长单词时有效，另起一行并在单词内部换行
4. 列表属性：
   1. 去掉li前面的项目符号（小圆点）：list-style：none；
5. 背景属性：
   1. 背景颜色：background-color：transparent； 设置背景色为透明，默认为透明
   2. 背景图片：background-image：none | url（路径）； 便于控制位置，用于装饰性小图片和超大背景图片,背景图片显示在背景颜色之上
   3. 背景平铺：background-repeat：repeat | no-repeat | repeat-x | repeat-y
   4. 背景图片位置：background-position：x y； xy可用百分比，50%表示居中
      1. x或y可用方位名词（left、top等）替代。若x和y都被替代，则两个值前后顺序无关。若只指定一个方位名词，第二个值默认居中
   5. 背景图像固定：background-attachment：scroll | fixed 设置图片是否随其他元素一起运动，可用于实现视差滚动
   6. 背景属性复合写法：background：背景颜色 背景图片地址 背景平铺 背景图像滚动 背景图片位置； 可以省略任意项
   7. 背景颜色半透明（CSS3）：background：rgba（0， 0， 0， 不透明度（0-1）） 如0.3可将0省略，只写.3
6. 元素的可见性：
   1. display：none | block； 不占位置，触发回流和重绘，有株连性（子孙节点全隐藏）
   2. visibility：hidden | visible； 占原先所在位置，触发重绘，伪株连性（可以给子孙节点设置visibility：visible可见）
7. 元素是否可用：
   1. pointer-events：none 设置元素不可用（阻止hover、active触发事件，阻止js点击动作触发事件等）
8. 盒子模型：
   1. 边框：位于盒子的外侧
      1. border：border-width（px） border-style border-color； 设置所有边框没有顺序要求
         1. border-style：none、solid（实线）、dashed（虚线）、dotted（点线）
         2. border-top（left、bottom、right）：border-width（px） border-style border-color； 设置上边框
      2. 设置相邻边框合并在一起：
         1. border-collapse：collapse；（仅表格有效） 或 设置margin-left：-1px；使得右元素的边框压住左元素的边框
   2. 内容：
      1. 盒子的width和height属性为内容的宽高，实际盒子的大小还要加上各边框和内边距。可利用此特性实现按字数变大小的盒子
      2. 行内或行内块元素可以通过设置padding并且不设置width的方法来通过内容撑大盒子
      3. overflow：visible（默认，超出部分可见）| hidden（超出部分隐藏）| scroll（不管是否超出，添加滚动条）| auto（超出则添加滚动条）
   3. 内边距：
      1. padding-right（-left、-top。。。）：20px 单独设置各内边距
      2. padding：各边内边距；或 上下 左右；或 上 左右 下；或 上 右 下 左； 单位为px
   4. 外边距（margin）：基本设置同内边距
      1. 使块级盒子水平居中：在给盒子指定了宽度的情况下，设置左右外边距为auto。（margin：0 auto；或分别设置左右）
      2. 行内元素水平居中：给其父元素设置text-align：center
      3. 外边距合并：同一个BFC中，相邻块级元素外边距会自动合并，取大值。
      4. 外边距塌陷：子元素指定垂直外边距时，子元素不生效，父元素的该外边距变为父子元素中该外边距的较大值。
         1. 解决方案：为父元素添加边框；父元素添加内边距；为父元素添加overflow：hidden；若子盒子为浮动，则无此问题
   5. 圆角边框（CSS3）：border-radius：10px | 百分比；
      1. 百分比为盒子宽高乘以指定百分比所得的值，可用于生成椭圆
      2. 若border-radius指定两个值，分别为左上右下、右上左下的圆角大小；
      3. 若border-radius指定三个值，分别为左上、右上左下、右下的圆角大小；
      4. 若border-radius指定四个值，分别为左上、右上、右下、左下的圆角大小；
      5. 分别指定：border-top（-bottom）-left（-right）-radius：。。。；
   6. 盒子阴影（CSS3）：box-shadow：h-shadow v-shadow blur（模糊距离） spread（阴影尺寸） color inset（若写，则表示内阴影）；
      1. 第一二属性必填，分别表示水平和垂直阴影的位置，允许负值。前四个属性为px。
   7. 文字阴影（CSS3）：text-shadow：h-shadow v-shadow blur color 属性参考盒子阴影

# 三种网页布局方式

* 1. 普通流（标准流/文档流）：标签按照规定好默认方式排列（块元素、行内元素、行内块元素）
  2. 浮动：float属性用于创建浮动框，将其移动到一边，直到左边缘或右边缘触及包含块或另一个浮动框的边缘
     1. float：属性值（none默认、left、right）；
     2. 浮动特性：
        1. 会脱离标准流（脱标，不保留原先位置，漂浮在后面的标准流之上）、
        2. 会一行内显示并且元素顶部对齐（父级宽度不够则另起一行）、
        3. 会具有行内块元素的特性
     3. 清除浮动：浮动的元素脱离标准流，不占位置，若父元素不设置高度，则无法匹配浮动元素的高度。
        1. 本质：clear：属性值； 元素的指定方向不出现浮动元素，属性值：left、right、both
        2. 清除浮动的方法：
           1. 额外标签法（隔墙法）：在最后的浮动元素后面添加空标签（必须是块元素），此标签设置clear属性
           2. 父级添加overflow（应用BFC特性）：给父级添加overflow属性，属性值设置为hidden、auto或scroll
           3. after伪元素法，给父元素添加：

.clearfix: after{ content: “”; display: block; height: 0; clear: both; visibility: hidden; }

.clearfix { \*zoom: 1; } IE6、7兼容用

* + - * 1. 双伪元素清除浮动，给父元素添加：

.clearfix::before, .clearfix::after { content: “”; display: table; }

.clearfix::after { clear: both; }

.clearfix { \*zoom: 1; }

* + 1. 浮动的元素会压住下面标准流的盒子，但不会压住盒子里的文字或图片
  1. 定位：自由地在某个盒子内移动位置或固定在屏幕中的某个位置，并且可以压住其他盒子。
     1. 定位=定位模式+边偏移
        1. 定位模式：决定元素的定位方式position：属性值；
           1. 属性值：

static（静态定位）：默认，无定位（标准流）。

relative（相对定位）：边偏移相对于原来的位置移动，移动后仍保留原先位置。

absolute（绝对定位）：相对于祖先元素来移动

如果没有祖先元素或祖先元素没有定位，则相对于浏览器移动

如果祖先元素有定位，会选择最近一级有定位的祖先元素移动

不再占有位置

fixed（固定定位）：固定于浏览器可视区的位置，以浏览器的可视窗口为参照点移动，不随滚动条滚动，不占有位置

固定在版心右侧位置：left：50%（浏览器宽度的一半）；margin-left：xx px（版心宽度的一半）；

sticky（粘性定位）：开始时随页面滚动，随后不随页面滚动。不兼容IE

以浏览器的可视窗口为参照点移动、占原先位置、必须添加至少一个边偏移才有效（边偏移条件满足时，不随页面滚动）

* + - 1. 边偏移：定位的盒子移动到的最终位置。有top、bottom、left和right属性（值为px）。分别表示距离父元素各边线的距离
    1. 定位叠放顺序：
       1. z-index：1； 数值越大越靠上，可以压住数值小的盒子，默认值为auto（小于1）。属性值相同，则后来居上
    2. 绝对定位居中对齐：left：50%（浏览器宽度的一半）；margin-left：-xx px（自身宽度的一半）；
    3. 行内元素添加了绝对定位或固定定位，可以直接给宽高；块级元素添加绝对或固定，不给宽高则默认为内容大小
    4. 定位不同于浮动，会压住下面标准流的文字或图片

# 其他界面样式

1. 字体图标：以icomoon下载的包为例
   1. fonts文件夹放入根目录，设置CSS（从包的style.css中复制）：@font-face {。。。}，从demo.html中复制字符，并给字符设置font-family
   2. 追加图标：import icon，选择要追加的图标，下载后覆盖原字体文件
2. CSS三角形（用i标签做）：设置div宽高、行高、字体大小为0，设置边框大小和颜色。
   1. 若设置四边框大小相等，生成四个等腰直角三角形；若设置下边框为0，上边框大于左右边框，生成两个直角三角形和一个等腰三角形
3. 用户界面样式：
   1. 鼠标样式：cursor：属性值； 设置在对象上移动鼠标指针的光标形状
      1. 属性值：default 默认 pointer 小手（链接） move 十字箭头 text 工字 not-allowed 禁止
   2. 轮廓线：outline：none； 去除表单选中时的轮廓线
   3. 文本域大小：resize：none； 禁止文本域大小的拖拽
   4. 行内、行内块元素的垂直对齐：vertical-align：属性值；
      1. 属性值：baseline 默认，元素位于父元素的基线（英文字母上往下的第三行）上 top 元素顶部与行中最高元素的顶部对齐
      2. middle 元素对齐于父元素的中部 bottom 元素底部与行内最低元素的底部对齐
      3. 消除图片底部空白缝隙：vertical-align：top / middle / bottom 或将图片转换为块级元素
   5. 溢出文字省略号显示：
      1. 单行文本：white-space：nowrap（禁止换行）；overflow：hidden；text-overflow：ellipsis（省略号替代超出的部分）；
      2. 多行文本（兼容性，适用于webKit浏览器或移动端，一般由后台人员写）：
         1. overflow：hidden；text-overflow：ellipsis；display：-webkit-box；-webkit-line-clamp：2（行数）；-webkit-box-orient：vertical；
   6. 常见布局技巧：
      1. 边框合并：
         1. 设置margin-left：-1px；使得右元素的边框压住左元素的边
         2. 此时若要添加hover高亮边框，需要使目标处于所有元素之上（定位或z-index）
      2. 文字环绕图片：结构为div>img+p 给img添加浮动和右侧外边距即可
      3. 行内/行内块元素水平居中：给父元素添加text-align：center；
      4. 禁用滚动条：overflow-y：hidden

# CSS3新增特性（IE9+）

* 1. 新增选择器：
     1. 属性选择器：
        1. 元素[属性] 选择具有某属性的元素
        2. 元素[属性=“值”] 选择有属性且属性值等于值的元素
        3. 元素[属性^=“值”] 选择有属性且属性值开头为值的元素
        4. 元素[属性$=“值”] 选择有属性且属性值结尾为值的元素
        5. 元素[属性\*=“值”] 选择有属性且属性值包含值的元素
     2. 结构伪类选择器：section div：nth-child（2） 元素类型可忽略不写
        1. 元素类型：first-child 若一个元素是它父元素的第一个元素，则选择
        2. 元素类型：last-child 若一个元素是它父元素的最后一个元素，则选择
        3. 元素类型：nth-child（n） 若一个元素是它父元素的第n个元素，则选择
           1. n可以是数字、关键字（even偶数、odd奇数）和公式（公式中的n从0开始递增1直到结束，如n+5为第五个到最后一个）
        4. 元素类型：first-of-type 若该元素是它父元素的第一个指定元素，则选择
        5. 元素类型：last-of-type 若该元素是它父元素的最后一个指定元素，则选择
        6. 元素类型：nth-of-type（n） 若该元素是它父元素的第n个指定元素，则选择
        7. 选择器1：not（选择器2） 对选择器1选中的元素应用选择器2，将其中不被选择器2选中的元素作为结果
     3. 伪元素选择器：利用CSS创建新标签元素
        1. ：：before｛｝ 在父元素内部的前面插入内容
        2. ：：after｛｝ 在父元素内部的后面插入内容
        3. 创建的元素为行内元素、此元素无法在文档树中找到、必须有content属性（content：‘’；）
  2. 弹性盒flex：用来代替浮动完成页面布局，使元素具有弹性，使其跟随页面的大小改变大小
     1. 弹性容器：display：flex / inline-flex 将元素设置为块级/行内弹性容器
        1. flex-direction：可选值 指定容器中弹性元素的排列方式（改变主轴和辅轴）
           1. row：默认值。弹性元素在容器内水平从左向右排列
           2. row-reverse：水平从右向左
           3. column：纵向自上向下
           4. column-reverse：纵向自下向上
        2. flex-wrap：可选值 设置弹性元素在弹性容器中是否自动换行（当容器大小不足时）
           1. nowrap（不换行） / wrap（沿辅轴方向换行） / wrap-reverse（沿辅轴反方向换行）
        3. flex-flow：flex-direction flex-wrap 复合设置两个属性
        4. justify-content：可选值 设置如何分配主轴上的空白空间
           1. flex-start（元素沿主轴起边排列）/ flex-end（沿主轴终边排列）/ center（元素居中排列）
           2. / space-around（空白分配到元素两侧）/ space-evenly（空白分配到元素单侧，最后一个元素两侧都有，兼容性问题）
           3. / space-between（空白分配到元素间）
        5. align-items：可选值 设置元素在辅轴上如何对齐
           1. stretch（将同一行元素的高度设置为相同的值）/ flex-start（元素不会拉伸，沿辅轴起边对齐）/ flex-end / center
           2. / baseline（基线对齐）
        6. align-content：可选值 同justify-content 设置如何分配辅轴上的空白空间（辅轴将一行视为一个元素）
     2. 弹性元素（弹性项）：是弹性容器的子元素，弹性元素也可以是弹性容器
        1. flex-grow：数值 指定弹性元素伸展的系数
           1. 默认值为0，将父元素剩余的空间按子元素系数大小分配后给子元素
        2. flex-shrink：数值 指定弹性元素收缩的系数
           1. 默认值为1。指定为0时，父元素装不下子元素时，该元素不收缩
        3. flex-basis：像素值（默认值为auto） 指定元素在主轴上的基础长度，若指定了具体值，会覆盖width或height
        4. flex：flex-grow flex-shrink flex-basis 复合指定样式
           1. 也可指定为initial（0 1 auto）、auto（1 1 auto）、none（0 0 auto）
        5. order：整数 指定弹性元素的排列顺序，越小越靠前，默认值为0
        6. align-self：可选值同align-items 设置单个元素的align-items
  3. 盒子模型box-sizing：
     1. box-sizing：content-box 默认，盒子大小为width+padding+border
     2. box-sizing：border-box 盒子大小为width
  4. 滤镜：filter：blur（5px）； 模糊处理，数值越大越模糊
  5. 执行计算：calc（表达式）； 例如width：calc（100%-80px）； 表达式中可用+-\*/
  6. 不透明度：opacity：1； 1表示不透明 0表示全透明
  7. 过渡（一般与hover一起用）：
     1. transition：要过渡的属性 花费时间（单位s，必写） 运动曲线（默认ease，可忽略） 触发延迟时间（单位s，默认0s，可省略）；
        1. transition写于变化前的元素中，在hover中写变化后的属性
        2. 要同时修改多个属性，在transition冒号后面写多个表达式，用逗号隔开；要让所有属性同时变化，则在要过渡的属性处写all
        3. 运动曲线：linear匀速、ease渐慢、ease-in加速、ease-out减速、ease-in-out先加速后减速
  8. 2D转换transform：处于标准流而不影响其他元素的位置
     1. 平移：transform：translate（x，y）；或transform：translateX（n）；或transform：translateY（n）；
        1. translate给百分比单位时参照自身宽高移动（left：50%；transform：translateX（-50%）；实现水平居中）
        2. 平移对行内元素无效
     2. 旋转：transform：rotate（x deg）； deg为度数，顺时针为正，默认按中心点旋转
        1. 设置旋转中心：transform-origin：x y； 可以设置px或方位名词，默认为50% 50%
     3. 缩放：transform：scale（x，y）； x和y为放大倍数，没有单位；可以只写一个参数表示两个参数一样；设置中心点方式同上
     4. 同时添加多种效果，在函数之间加空格隔开，需要按平移 旋转 缩放的顺序写，否则会影响坐标轴方向
  9. 动画：设置关键帧控制动画效果，页面加载后自动播放，默认不循环
     1. 定义动画：使用keyframes选择器
        1. @keyframes 动画名称 { 0% {width：100px；} 100%{width：200px；}}
        2. 可用关键词from和to分别代替0%和100%
        3. 若有多个状态，可用多个关键帧定义。例如：0%、25%、50%、75%、100% 百分比需要整数
     2. 使用动画：
        1. 用选择器选择元素，内写属性：animation-name：动画名称；animation-duration：持续时间；
        2. 可选属性：
           1. animation：动画名称 持续时间 运动曲线 何时开始 播放次数 是否反方向 动画状态；所有动画属性的简写方式

前两项必写，持续时间在何时开始之前，其他无顺序关系

* + - * 1. animation-timing-function：运动曲线；默认为ease渐慢、steps（指定步长，即动画过程中的总帧数）
        2. animation-delay：动画何时开始；默认为0
        3. animation-iteration-count：动画被播放的次数；默认1，可规定infinite无限
        4. animation-direction：是否在下一周期逆向播放；默认为normal，可规定alternate逆播放
        5. animation-play-state：运行状态；默认为running，可规定paused
        6. animation-fill-mode：结束后状态；forwards保持、backwards回到初始
      1. 元素可以添加多个动画，用逗号隔开
  1. 3D转换：z轴垂直屏幕向外为正
     1. 平移：translateZ（x px）； 一般不用百分比
     2. 3D复合平移：translate3d（x，y，z）； 不能省略xyz任意一项
     3. **透视（视距）**：perspective：x px； 眼睛到屏幕的距离，单位为像素。该属性写在被观察元素的父元素上
     4. 3D旋转：旋转方向按左手准则
        1. rotateX：（n deg）； 沿x轴旋转n度，yz同理。旋转中心默认在元素中央
        2. rotate3d：（x，y，z，n deg）； xyz表示旋转轴的矢量，（1，1，0，60deg）表示按照45°直线旋转60°
     5. 3D呈现：控制子元素是否开启三维立体环境
        1. transform-style：flat | preserve-3d； 属性写给父级元素。设为flat会把子元素投影到父元素所在平面，不遮挡元素

## 媒体查询

* + 1. @media 查询规则｛｝
    2. 媒体类型：all（所有设备）、print（打印设备）、screen（带屏幕的设备）、speech（屏幕阅读器）
       1. 多个媒体类型用逗号分隔，可在媒体类型前加only，表示只有这个媒体类型应用。only主要为了兼容老版本浏览器
    3. 媒体特性：width、height（视口宽高）、min-width、max-width（视口最小最大宽度），max不包括本数，min包括
       1. @media （width：500px）｛｝ 仅当width为500px时生效
       2. 若需指定多个条件，每个条件用（）括起来，条件之间用 逗号（或）/ and（与） 分隔
       3. css选择器写在媒体查询｛｝内部
       4. not用于查询规则的开头，对整个规则取反，需要指定媒体类型
       5. 常用断点：
          1. 0-768：超小屏幕 max-width=768px
          2. 768-992：小屏幕 min-width=768px
          3. 992-1200：中型屏幕 min-width=992px
          4. 1200-∞：大屏幕 min-width=1200px

1. 浏览器私有前缀：兼容老版本用
   1. -moz-、-ms-、-webkit-、-o- 分别代表火狐、ie、safari与chrome、opera
   2. 将前缀写在属性之前

# less

* 1. 一门css的预处理语言，以更少的代码实现更强大的样式
  2. 支持变量、mixin等等、浏览器需要现将less转换为css，然后执行
  3. 单行注释：//
  4. 选择器：
     1. 可将后代的【选择器2｛xx｝】写在父级的【选择器1｛｝】中，相当于【选择器1 选择器2｛｝】
     2. 用：hover时需写【&：hover｛｝】，&表示外层的父元素
  5. 运算符：+ - \* 可以直接使用 / 需要在括号内使用，或改用 ./
     1. @a = ~“xxx” 防止less编译xxx
     2. .col（@b）when （@a > 1）｛｝ 当when后面括号中的条件满足时执行
  6. 引入：@import “xxx.less” 将其他less文件引入
  7. 变量：
     1. 声明变量：@a：100px； 使用变量：width：@a；
     2. 变量可以是像素、颜色、类名。
     3. 为选择器的类名或值的一部分使用时：.@｛a｝｛。。。。。｝ url（“@｛a｝/1.png”）；
     4. 变量有作用域，内部更改变量不影响外部
     5. 引用其他属性的值：width：100px；height：$width；
     6. 变量有延迟加载特性，即先加载完声明变量再赋值到样式，可以看做把声明部分提前到作用域最前
  8. 扩展：
     1. 方法一：选择器1：extend（选择器2）｛。。。｝ 自动将选择器2的属性复制到选择器1中
        1. 若在选择器2后面加上关键词all，则会把选择器2所选元素的伪类同样添加给选择器1
     2. 方法二：选择器1｛选择器2（）；。。。｝ 同上，mixin混合，性能较上法差
  9. 混合函数：
     1. 在选择器后面添加（），这时就创建了一个mixins（混合函数），仅能被混合引用，不会生效
     2. 在括号内可以设置变量.test（@w，@h）｛width：@w；height：@h｝ 使用：.test（200px，300px）；
     3. 传参：可按顺序传，也可按名字传：（@h：300px，@w：200px）。传递时必须严格按参数数量传递
  10. 函数：
      1. average（颜色1，颜色2） 返回两个颜色的中间值
      2. darken（颜色，百分数） 返回加深的颜色
      3. percentage（数字） 将数字换为百分比数
  11. 循环：用递归实现
  12. 媒体查询：
      1. @media screen {
      2. @media (min-width: 1023px) {
      3. .selector {
      4. color: blue;
      5. }
      6. }
      7. }