CSSday02

一.尺寸和边框

1.尺寸属性

2.溢出处理

overflow

overflow-x(x 轴生成)　overflow-y(y 轴生成)

当内容多，元素区域小的时候，就会产生溢出效果，默认是纵向溢出。

取值：

1.visible 可见的，默认值，溢出可见

2.hidden 隐藏的，溢出的内容不可见

3.scroll 显示滚动条，不管是否溢出，都添加横 向和纵向滚动条，(不溢出不可用)

4.auto 溢出的部分，显示滚动条，并可用

3.边框属性

border: width　style　color;

宽度　样式　颜色

a.width: 边框的宽度，以 px 为单位的数字

b.style: 边框的样式

取值　solid 实线

dotted 虚线(点点)

dashed 虚线(短线)

double 双实线

c.color:边框的颜色，合法的颜色值（透明色 transparent)

d.取消边框 border:none/0;

这种写法，同时设置 4 个边框，是一种简写方式

练习

创建一个 div，id=content 尺寸 200*200

设置 4 个方向的边框，2px 实线 橙色

单边设置

border-top/right/bottom/left

```
border-top:5px solid orange;
border-right:3px dotted #f00;
border-bottom:7px dashed #0ff;
border-left:11px double #f0f;
```

单属性设置

border-属性: 属性值;

```
border-style:dotted;
border-width:10px;
```

单边单属性

border-方向-属性(共 12 个)

```
border-top-style:double;
border-top-color:#44f;
```

边框的倒角(圆角)

将直角倒成圆角 border-radius:

取值：以 px 为单位的数字

% 设置圆形(50%)

单角设置

使用两条边去确定一个角，先写上下后写左右

border-top-left-radius:左上

border-top-right-radius:右上

border-bottom-left-radius:左下

border-bottom-right-radius:右下

边框阴影

box-shadow: h-shadow v-shadow blur spread color

h-shadow 水平方向阴影偏移，必须值

v-shadow 垂直方向偏移量，必须值

以下可选参数

blur：阴影的模糊距离，可选值，越大模糊距离越明显

spread：阴影的尺寸，指定要在基础阴影上扩出的大小

color：阴影颜色

inset：把默认的外部阴影设置为内部阴影

练习：

设置一个圆形发光的效果(太阳，日食)

6.轮廓

轮廓指的是边框的边框，绘制与边框外边的线条

outline: width　style　color;

宽度　样式　颜色

练习:尝试完成一个乒乓球拍

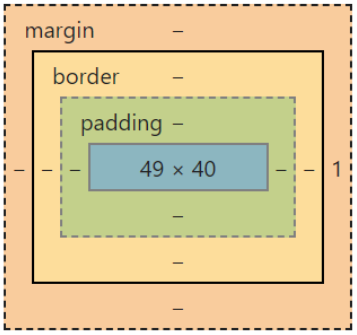
二.框模型（盒子模型）

页面元素皆为框（盒子）

定义了元素框处理元素内容尺寸，内边距，外边距以及边框的一种计算模式

定义了一个元素占地大小的计算方式

盒子模型默认的计算方式：



元素实际占地宽度=左外边距+左边框的宽度+左内边距+内容区域的宽度+右内边距+右边框的宽度+右外边距
元素实际占地高度=上外+上边框+上内+内容区域高度+下内+下边框+下外
内边距：元素边框与内容之间的空白间距
外边距：围绕在元素边框外的空白区域(元素与元素之间的距离)

1.外边距

①围绕在边框周围空白区域，外边距一定是透明的。
margin 定义 4 个方向外边距
margin-top/right/bottom/left 定义某一个方向的外边距
取值
1.以 px 为单位的数字
为正数 top，元素往下移动
left，元素往右移动
为负数 top,元素往上移动
left,元素往左移动
2.取值为%，相对于父元素尺寸的百分比
3.auto，自动计算块级元素的外边距，控制块级元素在水平方向居中对齐
margin 的效果，改变元素在页面上的位置，多用于微调

简写方式
margin:value; 设置 4 个方向外边距
margin：v1 v2; v1:上下外边距 v2：左右外边距
margin：0 auto; 控制块级元素在水平方向居中对齐（也可以直接写 auto，不推荐）
margin:v1 v2 v3; v1:上 v2:左右 v3：下
margin:v1 v2 v3 v4; 上 右 下 左(顺时针)

练习

05_ex 创建两个 div 300*300，随意设置两个不同背景颜色
两个 div 之间的间距为 50px
第二个 div 水平居中显示

②自带外边距的元素有哪些
h1~h6,p,body,ol,ul,dl,dd,pre
开发时候由于不同浏览器对默认外边距解析不同，通常要清除这些元素的自带外边距

③外边距的特殊效果
a.外边距的合并
当两个垂直外边距相遇时，他们将合并成一个外边距
最终的取值，取决于两个外边距中距离大的值

b.行内元素以及行内块元素的外边距
行内元素垂直外边距无效(img 除外)，水平外边距相遇两个值会相加
行内块，设置垂直外边距，整行元素跟着发生变化

c.外边距溢出
在特殊条件下，给子元素设置的上外边距，会作用到父元素上
特殊条件：父元素没有上边框
只有第一个子元素设置上边距时
解决方案：
1.为父元素增加上边框；弊端：元素实际占地高度变大
2.为父元素添加上内边距；弊端：元素实际占地高度变大
3.在父元素的第一个子元素位置添加一个空的<table></table>

练习

06_ex 创建两个 div，#d1 #d2 200*200 不同的背景颜色
在#d2 中添加一个 div#d3 100*100 添加不同背景颜色
最后设置 d3 的上外边距为 50px.查看页面效果

2.内边距

不会影响其他元素，但是会改变元素自己的占地尺寸，会变大

padding:value;设置 4 个方向的内边距
padding-top/right/bottom/left 定义单方向的内边距
取值：以 px 为单位的数字
%
简写方式
padding:value; 设置 4 个方向的内边距
padding:v1 v2; v1: 上 下 v2: 左 右
padding 没有 auto
padding:v1 v2 v3: v1:上 v2:左右 v3:下
padding:v1 v2 v3 v4;上 右 下 左

3.box-sizing 属性

设置盒子模型的计算方式
默认计算方式：
元素实际占地宽度=左外边距+左边框+左内边距+内容区域宽度+右内边距+右边框+右外边距
元素实际占地高度=上外边距+上边框+上内边距+内容区域高度+下内边距+下边框+下外边距

box-sizing 取值
content-box 默认值，会采用默认计算方式
border-box 元素的尺寸(定义宽高)，包含border,padding 和内容区域宽高
元素实际占地宽度=左右外边距+设置的宽度width
元素实际占地高度=上下外边距+设置的高度height

块级元素，行内元素，行内块

块级元素：独立成行，可以设置宽高，margin 上下有效

行内元素：与其他行内元素和行内块，共用一行
宽高无效，margin 上下无效

行内块：与其他行内元素和行内块，共用一行
可以设置宽高，margin 上下有效
一行行内元素，其中一个设置了上下margin,整行都跟着移动

CSSday03

一.背景

1.背景颜色

background-color:
取值
合法的颜色值
transparent;透明
注意：背景色的填充，默认从边框位置处开始

2.背景图片

background-image:url(资源路径)

3.背景图片的平铺

background-repeat:
取值：
1.repeat 默认值，平铺
2.repeat-x 水平方向平铺
3.repeat-y 垂直方向平铺
4.no-repeat 不平铺，只显示一张

4.背景图的定位

background-position：x y
取值:
1.以 px 为单位的数字
指的是，图片在水平和垂直方向的偏移量
2.x% y%
50% 50%中间
0% 0% 左上
100% 100% 右下
3.关键字
x：left/center/right
y：top/center/bottom

5.背景图的尺寸

background-size: width height
取值：
1.以 px 为单位的数字
2.%（强烈支持，响应式开发）
3.cover，填满容器，图片的宽高比**不变**，铺满整个容器的宽高，图片溢出的部分会被裁减掉将背景图同比缩放，直到背景图完全覆盖容器，不留一丝空白，图片溢出的部分会被裁减掉，图片可能会显示不完整，但是容器完全被背景图充满。
4.contain 图片自身宽高比不变，但是，要缩放到图片能够完全展示出来，所以会有空白区域。

6.背景图的固定

background-attachment:
将背景图固定在网页某个位置，一直在可视的区域中显示，不会随着网页滚动条改变位置
取值：
scroll 默认值，会随着网页滚动条改变位置。
fixed 固定，不会随着网页滚动条改变位置。

7.背景的简写方式

在一个属性中，指定背景多个值
属性：background
取值：color url repeat attachment position
background:#faa url('08.png') no-repeat center center;

1.先写结构布局 2.定义宽高，大体的位置
3.背景颜色图，边框所有 4.文字的所有
5.微调
从外往里写，从左往右，从上往下写

二.渐变 gradient

1.什么是渐变

多种颜色平缓变化的一种显示效果

2.渐变的主要因素---色标

色标，一种颜色以及他出现的位置
一个渐变至少有两个色标

3.渐变的分类

1.线性渐变，以直线的方向来填充渐变色
2.径向渐变，以圆形的方式实现渐变色
3.重复渐变，将线性渐变或径向渐变 重复几次

4.线性渐变

background-image
取值：
新版本：linear-gradient(方向，色标 1，色标 2.....)
色标的写法：颜色的合法值 位置的百分比
方向取值：to top/right/bottom/left
角度 按顺时针计算
0deg to top
90deg to right
180deg to bottom
270deg to left
旧版本兼容性写法：-浏览器内核-linear-gradient(方向,色标 1,色标 2....)
方向取值：top/right/bottom/left

5.径向渐变

background-image:radial-gradient(半径 at 圆心,色标 1,色标 2.....);
半径：以 px 为单位的数字
圆心：x y，以 px 为单位的具体数值
x% y% 元素的宽高占比
关键字 x:left/center/right
y:top/center/bottom

6.重复线性渐变

repeating-linear-gradient(to right,#faa 0px, #ffa 50px);
注意在重复渐变中，色标的位置，要给绝对值数字，不要用%（可能出问题）

7.重复的径向渐变

repeating-radial-gradient(50px at center,#060606 0px,#af8bd8 20px,#292727 35px);
注意在重复渐变中，色标的位置，要给绝对值数字，不要用%（可能出问题）

8.浏览器兼容性

各个浏览器新版本都渐变
对于不支持的旧版本，可以添加浏览器内核前缀的方式，做兼容，让浏览器支持
chrome&safari: -webkit-
firefox:-moz-
ie:-ms-
opera: -o-
background: -webkit-linear-gradient(left, #006699 0%,#f00 25%, #990066 50%,#ff0 75%,#996600 100%);

三.文本格式化（重点******）

1.字体属性

1.指定字体的类型
font-family:"类型 1","类型 2"....
例：font-family:"宋体","黑体"....

2.字体大小
font-size
以 px、pt 为单位的数字
em rem

3.字体加粗（权重）
font-weight:
bold/normal/lighter/bolder
无单位的数字 100-1000 整百数 常用 400-900

4.字体样式
font-style
normal
italic（尽量不用标签斜体，不方便维护）

5.小型大写字母
font-variant:
normal
small-caps;

6.字体属性的简写方式
font:style variant weight size family
font:italic small-caps bold 30px "黑体";
简写方式最少要有 size 和 family.

练习：

创建 04_ex，一个 div 内容随意，最好中英结合，设置以下样式
1.字体为黑体
2.大小为 32px
3.加粗并斜体显示
4.所有小写字母变成小型大写字母

2.文本格式化

1.文本颜色 color:合法的颜色值
2.文本的水平对齐方式 text-align 取值：left/center/right/justify(两端对齐)
3.行高 定义一行数据的高度 如果行高大于字体本身的大小，该行文本在行高内成垂直居中的显示效果 line-height: 以 px 为单位的数字，一般与容器的高相同 无单位数字，line-height:2（是字体大小倍数）
4.线条的修饰 text-decoration 1.none 没有线条，去掉 a 标签的下划线 2.overline 上划线 3.underline 下划线 4.line-through 删除线

5.首行缩进

text-indent: 以 px 为单位的数字

6.文本的阴影

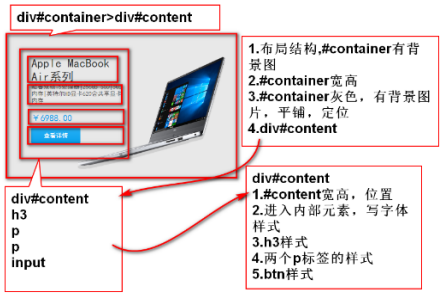
text-shadow: h-shadow v-shadow blur color;

练习

在 04_ex 中，创建 div#d2,内容随意，添加以下样式 1.文本水平两端对齐 2.有下划线 3.行高是字体大小的 1.5 倍 4.首行缩进 50px

作业：

```
transition: 1s;
```



CSSday04

一.表格的样式

1.表格的常用属性

之前学习的样式，在表格中，基本都可以使用 1.边框属性 table 和 td 2.边距属性 table 都能使用，td 没有 margin 3.尺寸属性 4.文本格式属性 font-* text-* line-height 5.背景属性 颜色 图片 渐变 6.vertical-align:top/middle/bottom 要 写在 td 才生效

2.表格特有的属性

1.边框合并 border-collapse a.separate 默认，边框分离的模式 b.collapse 边框合并	
2.边框的边距 设置单元格与单元格之间的距离,必须保证边框是分离状态 border-collapse:separate; border-spacing a.取值一个 value,垂直和水平距离相同 b.取值两个 value1 value2. value1:水平距离，value2:垂直距离	
3.标题的位置 caption-side 取值：top/bottom	
4.表格的显示规则,告诉浏览器，我这张表格如果渲染，td 尺寸的计算方式 table-layout: a.auto 默认值，自动表格布局，列的尺寸实际是由内容决定的 b.fixed 固定表格的布局，列的尺寸由设置的值决定 需要 table 有尺寸，td 有尺寸	
自动布局 auto	固定布局 fixed
单元格的大小会适应内容	单元格尺寸取决于设定的值
表格复杂时，加载速度慢(先加载，再渲染)	任何情况下会加速加载表格(优点)
布局会比较灵活(优点)	布局会比较死板（缺点）
适用于不确定每列大小，并且表格不太繁琐的时候	当确定每列尺寸的时候使用

二.定位(重点**********)

1.什么是定位

元素在页面中的位置

2.分类

1.普通流定位　默认文档流
2.浮动定位
3.相对定位
4.绝对定位
5.固定定位

3.普通流定位

又称文档流定位，页面中所有元素默认显示方式
a.每个元素在页面中都有自己空间(盒子模型)
b.每个元素都是从父元素的左上角开始显示的
c.块级元素默认按照从上往下的方式逐个排列，每个元素独占一样
d.行内元素是多个元素在一行中显示，从左往右排列

4.浮动

浮动特点
a.元素一旦浮动起了，脱离文档流
不占页面空间，其它未浮动元素会上前补位
b.浮动元素会停靠在父元素的左边或者右边
或其它已经浮动元素的左/右的边缘
c.浮动解决多个块级元素在同一行内显示的问题（作用）
d.元素只会在当前行浮动
e.浮动元素依然包含在框内

float
取值：left：左浮动，元素浮动起来后，停靠在父元素的左侧，或者往左挨着已浮动元素
right：右浮动，元素浮动起来后，停靠在父元素的右侧，或者往右挨着已浮动元素

5.浮动元素引发的特殊情况

1.当父容器横向显示不下所有浮动元素的时候，最后一个元素将换行显示。
a.默认情况下，最后一个元素会优先显示在最高的位置，再往左排列。
b.会发生**浮动元素占位**的情况，浮动元素根据**浮动方向**，占据方向之上的位置，不允许最后一个元素占用。

2.浮动对默认宽度的影响
块级元素不写宽度，宽度默认是父级元素的100%，元素一旦浮动起来，宽度以内容为准。前题，不设置 width。

3.元素一旦浮动起来，就变为块级元素，允许修改尺寸，设置垂直外边距。

4.文本，行内元素，行内块，是不会被浮动元素压在下方，而会巧妙避开，环绕着浮动元素显示。

6.清除浮动

元素一旦浮动之后，会对后续的元素带来一定的影响，后续元素会上前补位，如果不希望后续上前补位，可以给后续元素添加清除浮动的属性 clear:
取值：left:清除左浮动元素对我带来的影响
right:清除右浮动元素对我带来的影响
both: 清除所有浮动元素带来的影响

7 浮动元素对父元素高度带来的影响(高度坍塌)

块级元素不写高度，他的高度是根据内容判定的如果内部元素都浮动了，内部元素都脱了默认文档流，父级元素就失去了高度------高度坍塌
解决方案：
a.给父级元素添加高度　弊端：不是每次都知道具体的高度是多少
b.设置父级元素也浮动　弊端：会影响父元素后面的其它元素
c.给父元素设置 overflow:hidden/auto
弊端：如果内容想要溢出显示，显示不了。
d.在父元素中追加一个空的块级元素，只设置 clear:both

三.显示

1.显示方式

决定了元素在网页中表现形式(块级，行内，行内块，table)
语法 display
取值：
a.none 不显示元素，隐藏
b.block 让元素表现为块级
块级元素特征，独占一行，可以设置尺寸以及上下外边距
c.inline 让元素表现为行内
行内元素特征，多个元素一行显示，不能设置宽高以下上下外边距
d.inline-block 让元素表现为行内块
行内块特征，多个元素一行显示，可以设置尺寸以及上下外边距
e.table 让元素表现为 table
table 的特征：尺寸以内容为准，独占一行，允许修改尺寸

2.显示的效果

属性：visibility
取值　visible/hidden

visibility:hidden 和 display:none 的区别

visibility:hidden 隐藏，不脱离默认文档流，依然占位

display:none 隐藏，脱离了默认文档，不占页面位置

3.透明度

opacity:0~1　　1 是不透明，0 是全透明

opacity 和 rgba 的区别

opacity 作用与元素，当一个元素设置了 opacity 之后，这个元素本身，以及所有子元素，与颜色相关的属性，都会受到影响

rgba(0~255,0~255,0~255,0.5)只会改变你设定的这个颜色透明度

4.垂直对齐

vertical-align

使用在 table 中,取值 top/middle/bottom

使用在 img 中，取值 top/middle/bottom/baseline(影响图片两边文字与图片对齐方式，控制图片与两边文字垂直对齐方式)

5.光标

改变鼠标悬停在元素时的样式
cursor:
取值
default 箭头　　pointer 小手
crosshair +　　text I
wait 等待　　help 帮助

四.列表

列表标识
list-style-type
取值：
none/disc/circle/square

列表标识的位置
list-style-position:outside/inside

列表项引用图片
list-style-image:url()

简写方式
list-style:type url() position;
常用的方式：
list-style:none;

CSS3 高级 01

一.相对/绝对/固定定位 position（重点）

position
取值：
static 静态，默认值。文档流定位
relative 相对定位
absolute 绝对定位
fixed 固定定位
注意：将元素的 position 设置为 relative/absolute/fixed 中的任意一个，此元素被称为**已定位元素**

偏移属性

top/right/bottom/left　距离哪个方向多远
取值是以 px 为单位的数字

1.相对定位

元素相对于它原来的位置偏移
在元素位置做自身微调的时候使用（与 margin 不同，margin 会挤其他元素）**/配合绝对定位使用**
position:relative;配合偏移属性使用
相对定位不脱离文档流，不会让后面元素补位，不会变成块级元素。

练习：

ul 中 5 个 li，横向显示，每个 li 中有图片，鼠标放到 img 上，图片往左移动 8 像素，往上移动 8 个像素

2.绝对定位

position:absolute;配合偏移属性使用
特点：
脱离文档流，不占页面空间，后面元素要补位
绝对定位的元素，会相对于"最近的""已定位的""祖先元素"去做偏移量，若没有"已定位的""祖先元素"，相对 body 偏移。
绝对定位会让元素变为块级
出现堆叠效果，就使用绝对定位）
对其使用 margin 时，除了 auto 其他均可使用

3.固定定位

将元素固定在页面的某个位置，位置不会随着滚动条发生变化，一直固定在可视区域。
固定定位始终相当于 body 去实现位置偏移。
固定定位脱离文档流，不占页面空间，定位的元素会变成块级。
语法：position:fixed;配合偏移属性一起使用

4.堆叠顺序

默认堆叠顺序，后发生定位元素，堆叠层级高。
默认堆叠顺序，不会大于设置 z-index 的顺序
z-index:无单位数字，理论最大值 2^31-1。
父子元素之间，子元素永远在父元素上面
只有已定位元素，才能使用 z-index

综合练习

二.css3 core

1.复杂选择器

1.兄弟选择器
作用于兄弟元素，只能往后找，**不能往前找**
兄弟元素：具备相同父级元素的平级元素之间，成为兄弟元素
a.相邻兄弟选择器
选择器 1+选择器 2{}
在选择器 1 的后面，紧跟着一个元素，这个元素还得匹配到选择 2
b.通用兄弟选择器
选择器 1~选择器 2{}
获取某元素后面所有满足条件的兄弟元素

2.属性选择器
id class name type value style titile ....
允许通过元素所带的属性以及值来匹配元素
①基本语法
[attr]{} ex:[id]{}匹配有 id 属性的标签
②elem[attr]
div[title]{} 匹配有 title 属性的 div 标签
③[attr1][attr2]
[class][title]{color:#f00;} 匹配同时具有 class 属性和 title 属性
④[attr=value]{}
[id=p2]{} 匹配 id 值为 p2 的元素
a.[attr*=value]属性值，包含指定值的元素
b.[attr~=value]属性值，包含指定单词元素
c.[attr^=value] 属性值,以 value 开头元素
d.[attr$=value] 属性值,以 value 结尾元素

3.伪类选择器
①目标伪类
突出显示获得的 HTML 锚点元素，匹配被激活的锚点
选择器:target{} div:target{}
②结构伪类
a.:first-child{} 匹配的元素是其父元素的第一个子元素（找大哥）
b.:last-child{} 匹配的元素是其父元素的最后一个子元素（找最小兄弟）
c.:nth-child(n){} 匹配的元素是其父元素的第 n 个子元素，n 从 1 开始（找第 n 个兄弟）
d :empty 匹配没有子元素的元素
div:empty{} 匹配没有子元素的 div 元素
e.:only-child
p:only-child{} 匹配属于其父元素的唯一子元素
③否定伪类 :not(selector)
div:not(:empty){} 匹配非空的 div

练习

07_ex 创建 4*4 的表格，400px*400px 内容随意
第一行背景颜色为#aaf
最后一行背景颜色为#ffa
第三行第二列的背景颜色为#faa

练习

在上面表格基础上，
设置除了第一列以外的其他所有单元格字体都 24px，加粗，倾斜

4.伪元素选择器

h2::first-letter{color:#f00;}
匹配某个元素的首字字符
h2::first-line{color:#aaf}
匹配某个元素的首行字符，首行与首字符冲突，优先首字符
h2::selection{color:#ffa}（必须双冒号）
匹配用户选取的内容，只能修改文本颜色和背景颜色

练习

08_ex 创建一个 div#d1 内容随意
将 d1 中的首字符设置为加粗红色，大小为 20px
右边距 15px，首行文本设置为橙色并倾斜。
用户选取内容是将文字颜色设置为黄色文字，红色背景

4.内容生成

使用 css 动态的向元素中插入一段内容
1.:before 或者::before 匹配元素的内容区域之前（肚子里），插入的伪元素。
2.:after 或者::after 匹配元素的内容区域之后（肚子里），插入的伪元素

```
<div id="d1">
    (::before)
    我是要成为海贼王的男人
    (::after)
</div>
```

应用：
可以解决外边距溢出：
父元素:before{ content:"";display:table;}
和高度坍塌。
父元素:after{ content:"";display:block}

练习

子曾经曰过：
学而时习之，不亦说乎!
----《摘自论语》

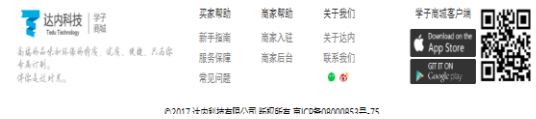
5.内容生成可以解决的问题

1.外边距溢出
父元素没有上边框，给第一个子元素设置上外边距，父元素也会发生作用
解决方案，在第一个元素之前添加一个<table></table>

作业：

1.使用定位完成

2.完成学子底部



3.使用内容生成，解决高度坍塌

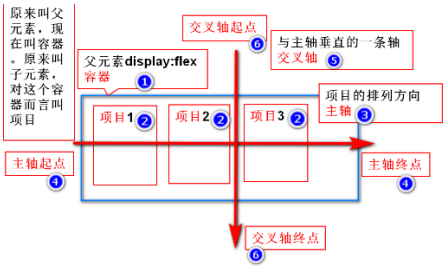
CSS3 高级 02

一.弹性布局（重点**********）

1.什么是弹性布局

弹性布局就是一种布局方式
主要解决某元素中的子元素的布局方式，为布局提供最大的灵活性

2.弹性布局的相关概念和专业术语



a.容器
要布局的子元素的父元素称之为容器，容器中写 display:flex
b.项目
要布局的子元素称之为项目
c.主轴
项目们排列的方向，称之为主轴（水平和垂直）
如果项目们是按照横向排列，x 轴就是主轴
如果项目们是按照纵向排列，y 轴就是主轴
d.交叉轴
与主轴垂直相交的方向轴叫做交叉轴

语法

将元素变为弹性容器，他所有的子元素将变成弹性项目，按照弹性布局的方式去排列显示
display
取值：flex， 将块级元素变为容器
inline-flex,将行内元素变为容器

元素设置为 flex 容器之后，子元素一些样式属性会失效
float/clear/vertical-align 失效
子元素允许修改尺寸(项目是行内元素也可以修改尺寸)，容器的对齐方式 text-align 失效

3.弹性容器的样式属性

①flex-direction 主轴方向
取值：
row 默认值 主轴是 x 轴，主轴起点在左端
row-reverse 主轴是 x 轴，主轴起点在右端
column 主轴是 y 轴，主轴起点在顶端
column-reverse 主轴是 y 轴，起点在底部

②flex-wrap
一个主轴排列不下所有项目时,项目的显示方式
取值:
a.nowrap 默认值，空间不够，不换行，项目会自动压缩
b.wrap 空间不够，就换行，项目不压缩
c.wrap-reverse 换行，反转

③flex-flow
是 flex-direction 和 flex-wrap 的缩写
取值：direction wrap;
ex:flex-flow:row wrap;

④justify-content
定义项目在主轴上的对齐方式
取值：space-between 两端对齐
space-around 每个间距相同
flex-start 默认值 在主轴起点对齐
flex-end 在主轴的终点对齐
center 在主轴上居中对齐

⑤align-items（容器高度大于项目高度）
项目们在交叉轴上的对齐方式
取值：flex-start 交叉轴起点对齐
flex-end 交叉轴终点对齐
center 交叉轴居中对齐
baseline 交叉轴基线对齐
stretch 如果项目未设置高度，在交叉轴上充满容器

4.项目的属性

是单独设置给一个项目的，不影响容器和其他项目

a.order
取值为无单位整数，定义项目排列顺序，值越小离起点越近，默认值为 0

b.flex-grow
定义项目放大比例
如果容器有放大空间，项目将按比例放大
取值为无单位整数

c.flex-shrink
定义项目缩小比例，空间不足时，项目该如何缩小
取值为无单位整数
默认值为 1，空间不足等比缩小
0 不缩小
取值越大，占据空间越小

d.align-self 控制当前项目在交叉轴上的对齐方式，与其他项目无关
取值：flex-start 交叉轴起点对齐
flex-end 交叉轴终点对齐
center 交叉轴居中对齐
baseline 交叉轴基线对齐
stretch 如果项目未设置高度，在交叉轴上充满容器
auto 继承容器的 align-items 的效果

5.CSS hack

由于不同的浏览器，对 css 解析认识不同会导致生成的页面效果不同，无法得到我们预期的效果
这个时候我们需要 些不同的 css 让代码同时兼容不同浏览器
这个针对不同浏览器写不同 css 的过程叫做 CSS hack（兼容）
tmooc 有相应视频

二．转换（重点*****）

1.什么是转换

改变元素在网页中的位置 大小 角度 形状的一种方式
2D 转换，在 x 轴和 y 轴上发生转换效果
3D 转换，增加了 z 轴

2.转换属性

a.transform:
取值：transform-function
none
如果有多个转换函数，用空格隔开

b.transform-origin: 转换原点
取值：px 为单位数字/%/关键字
2 个值，表示 x 轴和 y 轴上的位置
3 个值，x,yz 轴。

3.2D 转换函数

a.位移 改变元素在页面中的位置
transform:translate(x)指定元素在 x 轴上的位移距离（+元素往右，-元素往左）
transform:translate(x,y)指定元素在 x 轴和 y 轴上的位移距离
（x 取值同上；y 取值 +往下 -往上）
transform:translateX(x)指定元素在 x 轴上的位移距离
transform:translateY(y)指定元素在 y 轴上的位移距离