css架构模板：

架构说明：

1. css ：所有通用css
2. vue/公共项目：基于vue的通用css

项目编码：s系列： s1.xxxx

|  |  |
| --- | --- |
| 标题说明 | |
| 标题辨认 | 标题1-6级对应1-6号（字体），7级6号斜体，8级绿色6号，9级6号红 |
| 标题粗细间隔、每级有带色附加标题 酒红 |
| 标题缩进 | 级数-2 |
|  | 标题上下间距间隔 0.4行 |

|  |  |
| --- | --- |
| 标题结构 | |
| 级数 | 1-9 |
| 附加标题级数 | 3-9 **色** |
| 知识点 | 蓝 正文 |
| 提示 链接 小字 | 蓝 正文小一号 |
| 易忘 |  |
| 习题 |  |
| 搜索 关键词 | 灰色背景 |

[一. css 3](#_Toc25314)

[1.概述 3](#_Toc4257)

[2.CSS语法规范 3](#_Toc30802)

[3.CSS基础选择器（8种简单的） 6](#_Toc31909)

[4. 尺寸与边框 8](#_Toc21223)

[5.框模型 13](#_Toc9757)

[6.背景 17](#_Toc27513)

[7.渐变 19](#_Toc23451)

[8.文本格式化属性 22](#_Toc4864)

[9.表格 24](#_Toc15396)

[10.定位-浮动定位（重点） 26](#_Toc24242)

[11.显示 28](#_Toc14667)

[12.列表 29](#_Toc20002)

[13.定位-相对 绝对 固定 30](#_Toc8035)

[二．CSS3高级 32](#_Toc17218)

[三．H5（原生js） 45](#_Toc31728)

[1.公共项目 45](#_Toc25456)

[1.公共页面样式 45](#_Toc15622)

[1.button 45](#_Toc29175)

[2.loading 45](#_Toc30773)

[2.pc 46](#_Toc30648)

[1.项目1 46](#_Toc32299)

[2.项目2 46](#_Toc26823)

[3.app 46](#_Toc18503)

[1.项目1 46](#_Toc16449)

[2.项目2 46](#_Toc30615)

[4.移动h5 46](#_Toc2996)

[1.项目1 46](#_Toc6445)

[2.项目2 46](#_Toc27332)

[四．jq 47](#_Toc16808)

[1.标题2.1 47](#_Toc27891)

[1.标题3.1 （3级目录红） 47](#_Toc25180)

[2.附加3.2 47](#_Toc18104)

[1.标题4.1 47](#_Toc3211)

[2.附加4.2 47](#_Toc24652)

[1.标题5.1 47](#_Toc15238)

[1.标题6.1 47](#_Toc5074)

[1.标题7.1 47](#_Toc17583)

[1.标题8.1 47](#_Toc24004)

[1.标题9.1 法沙发撒飞洒发发 47](#_Toc17588)

[五．vue 47](#_Toc6605)

[1.公共项目 47](#_Toc23418)

[2.pc 47](#_Toc30586)

[1.项目1 47](#_Toc21652)

[2.项目2 47](#_Toc28007)

[3.app 47](#_Toc2106)

[1.项目1 47](#_Toc26319)

[2.项目2 47](#_Toc24852)

[4.移动h5 48](#_Toc1708)

[1.项目1 48](#_Toc5470)

[2.项目2 48](#_Toc29897)

[六．react 48](#_Toc22063)

[1.公共项目 48](#_Toc3190)

[2.pc 48](#_Toc3558)

[1.项目1 48](#_Toc16826)

[2.项目2 48](#_Toc20231)

[3.app 48](#_Toc31029)

[1.项目1 48](#_Toc24344)

[2.项目2 48](#_Toc12316)

[4.移动h5 48](#_Toc25909)

[1.项目1 48](#_Toc31980)

[2.项目2 48](#_Toc17013)

[七．anglar 49](#_Toc30400)

[1.公共项目 49](#_Toc2653)

[2.pc 49](#_Toc16508)

[1.项目1 49](#_Toc1003)

[2.项目2 49](#_Toc7720)

[3.app 49](#_Toc8197)

[1.项目1 49](#_Toc28197)

[2.项目2 49](#_Toc30796)

[4.移动h5 49](#_Toc27953)

[1.项目1 49](#_Toc16807)

[2.项目2 49](#_Toc9034)

# css

## 1.概述

1.什么是CSS?

CSS：Cascading Style Sheets 层叠样式表，级联样式表，简称：样式表。

2.作用

设置HTML元素的样式

3.CSS与HTML之间的关系

HTML：构建网页的结构内容

CSS：构建网页的样式效果

4.HTML属性与CSS属性的使用原则

w3c尽量使用CSS属性去取代HTML的属性来修饰元素。

## 2.CSS语法规范

1.使用CSS样式表的方式

1.内联样式

又称为“行内样式”

特点：将CSS样式定义在HTML开始标记中

语法：

<ANY style="CSS的样式声明"></ANY>

样式声明：

1.由样式属性和值组成；

2.属性名与值之间用冒号连接；

3.多个样式声明之间用分号分隔。

常用的样式属性和值：

1.设置文本颜色的属性和值

属性：color

值：合法的颜色值（颜色英文单词red）

2.设置背景颜色的属性和值

属性：background

值：合法的颜色值

3.设置字体大小的属性和值

属性：font-size

值：数字+px（像素）

css简单练习：

1.在网页中新建一对p元素，内容随意;

2.设置p元素中文本颜色为紫色(purple)，背景颜色为橙色orange)，字体大小为34px。

2.内部样式

在网页的头元素中增加一对<style></style>标记，在<style>标记中声明该网页要用到的所有样式。

语法：

<head>

<style>

样式规则

</style>

</head>

样式规则：由选择器和样式声明组成

选择器：规范了页面中哪些元素能够使用声明好的样式。其实就是把声明好的样式匹配给页面中的元素。

元素选择器：由元素名称作为选择器

样式规则：

选择器{样式声明}

div{

color:red;

}

标签选择器练习：

在网页中添加若干3级标题元素，内容随意

用内部样式设置3级标题元素中文本颜色为黄色，背景颜色为红色。

3.外部样式

独立于任何网页的位置处，声明一个样式文件（以.css为后缀），在css文件保存样式规则。

使用步骤:

1.创建独立的.css文件

2.在css文件中保存样式规则

3.在网页中引入css文件

<head>

<link rel="stylesheet" href="css文件的url">

</head>

创建css样式文件练习：

在网页中添加一对span元素，内容随意，使用外部样式为span元素设置文本颜色为绿色，背景颜色为灰色，字体大小为40px;

2.CSS的样式特征

1.继承性

必须是父子关系结构

大部分的样式可以直接继承给子元素（子继承父）

2.层叠性

可以为一个元素设置多个样式规则，规则中的属性不冲突时，可以同时都应用到元素上的。

3.优先级

如果样式声明冲突时，则按照默认的优先级来应用样式。

默认优先级由低到高：

浏览器的默认设置 低

外部样式表和内部样式表 中(就近原则)

内联样式 高

优先级练习：

在页面中用内部样式为div元素设置文本颜色为红色，字体大小为35px，用外部样式设置文本颜色为蓝色，引入外部样式后查看页面字体显示效果，然后交换外部和内部样式的位置，再查看页面效果。

4.调整显示优先级

调整显示的优先级

使用!important规则改变优先级

将!important放在属性值之后，与值之间用空格隔开。

ex:

color:red !important;

## 3.CSS基础选择器（8种简单的）

1.通用选择器

语法：\*{样式声明}

作用：可以匹配页面中的任意元素。

注意：效率较低，尽量少用

2.元素选择器

语法：元素名称{样式声明}

作用：设置页面上某种（类）元素的默认样式

ex:

div{},p{},a{},header{},span{}...

3.类选择器

作用：定义某个或某些元素的样式（谁想用谁就可以引用）

特点：通过元素的class属性进行引用

语法：

1.声明类选择器

.类名称{样式声明}

注意：

1.类名不能以数字开头

2.不能包含特殊字符（除\_,-外）

3.点 不能省略

2.引用

<ANY class="类名称"></ANY>

特殊用法：

1.多类选择器

让元素引用多个类选择器

语法：<ANY class="类名1 类名2 类名3 ..."></ANY>

2.分类选择器

将元素选择器和类选择器联合使用

语法:元素名称.类名{样式声明}

ex:

p.test{color:red;}

练习：

在页面中创建div和p元素，内容随意

1.用类选择器设置所有元素字体颜色为红色

2.用分类选择器为p元素设置背景颜色为黑色

4.id选择器（私人订制）

作用：设置指定id元素的样式

语法：#id值{声明样式}

#map{color:red;}

<div id="map"></div>

练习：

1.新建一个div元素，设置id为main

2.通过id选择器设置文本颜色为黄色，字体大小为32px,斜体（font-style:italic;）

5.群组选择器

作用：将多个选择器放在一起进行样式的声明定义

语法：选择器1,选择器2,选择器3{样式声明}

ex：

div,p,a.text,#one,span{color:gray;}

div{color:gray;}

p{color:gray;}

a.text{color:gray;}

#one{color:gray;}

span{color:gray;}

6.后代选择器

作用：通过元素的后代关系匹配元素

后代：一级或者多余一级的嵌套

语法：选择器1 选择器2 选择器3{样式声明}

7.子代选择器

作用：通过元素的子代关系匹配元素

语法：选择器1>选择器2{样式声明}

8.伪类选择器

作用：匹配元素不同状态的选择器

语法：所有的伪类选择器都是以:作为开始

选择器:伪类选择器{样式声明}

1.链接伪类

:link 匹配元素尚未访问时的状态

:visited 匹配元素访问后的状态

2.动态伪类

:hover 匹配鼠标悬停在元素上时的状态

:active 匹配元素被激活时的状态

:focus 匹配元素获取焦点时的状态

:load 加载完成

伪类选择器课后作业：

页面中新建一个a标记，增加以下样式：

1.未被访问时颜色为黑色，没有下划线(text-decoration:none;)

2.鼠标悬停时，文本颜色为红色

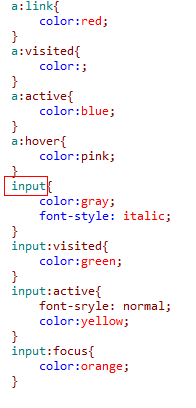
3.被激活时，文本颜色为橙色

4.访问后，颜色变为绿色

页面中增加一个input文本框，默认效果颜色为灰色，，字体为斜体（font-style:italic;）

1.被激活时，字体为非斜体（font-style:normal;）

2.获取焦点时，文本颜色为红色。



css day02

9.基础选择器的优先级

权值：标识当前选择器的重要程度，权值越大优先级越高。

元素选择器 1

类选择器 10

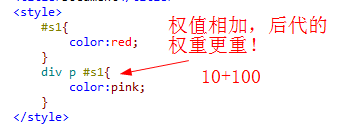
伪类选择器 10

ID选择器 100

内联样式 1000

选择器的权值加到一起，大的优先

权值相同 以后定义的为主



## 4. 尺寸与边框

1.单位

1.尺寸单位

1.px 像素 1024\*768px

2.in 英寸 1in=2.54cm

3.pt 磅 1pt=1/72in 用于描述文字的单位

4.cm 厘米

5.mm 毫米

6.em 相对于父元素乘以倍数（12px\*2em=24px）

7.rem 相对于根元素（body）

8.% 百分比

2.颜色单位（颜色取值）

1.rgb(r,g,b)

r:red

g:green

b:blue

r,g,b:取值0-255

2.rgb(r%,g%,b%)

3.rgba(r,g,b,alpha)

alpha:透明度，0-1之间的小数，值越大越不透明。

ex:

color:rgb(0,0,0,0.3)

4.#rrggbb

由0-9和a-F组成

#ffffff

#000000

#dddddd

5.#rgb是上面的缩写

#ffffff-->#fff

#000000-->#000

#dddddd-->#ddd

6.颜色的英文单词

red,green,gray,purple,blue...

2.尺寸

1.作用

改变元素的宽度和高度

2.语法

属性：

宽度：width

min-width 最小宽度

max-width 最大宽度

高度：height

min-height 最小高度

max-height 最大高度

3.注意：页面中哪些元素能修改尺寸

1.所有的块级元素都允许修改尺寸

div,p,h1,h2...,结构标记，ul,ol,dl,dt,dd

2.行内块元素允许设置尺寸

表单控件元素（单选按钮和复选框除外）

3.本身具备width和height属性的元素

img,table

大部分的行内元素是无法修改尺寸的。

4.溢出处理

当内容多，元素区域小的时候，就会产生溢出的效果，默认都是纵向溢出。

属性：

overflow,overflow-x,overflow-y

取值：

1.visible 可见的，默认值，溢出可见

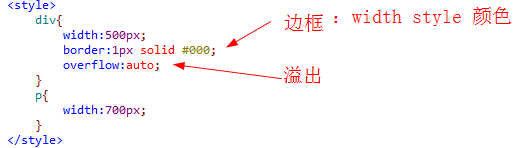
2.hidden 隐藏的，溢出的内容全部隐藏，不可见

3.scroll 显示滚动条，溢出时，可用

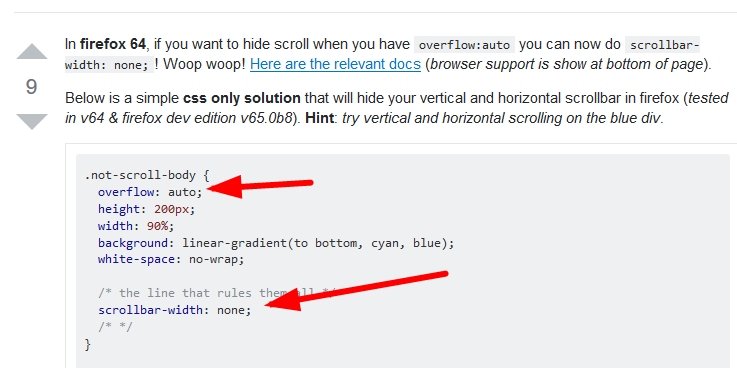
4.auto 自动，溢出时才显示滚动条并可用

溢出练习：

边框 border:1px solid #000;



\*隐藏火狐滚动条



3.边框

1.边框简写方式

属性：

border:width style color;

width:边框的宽度（粗细）

style:边框的样式

取值：solid 实线

dotted 虚线（圆点）

dashed 虚线（线条）

color:边框颜色（transparent透明色）

以上方式是同时设置四条边框的样式。

取消边框：border:0/none;

边框练习：

网页中创建一个div元素，设置高宽各300px,并且设置四条边框为1px的宽度，橙色（orange）的实线条。

2.单边框定义

设置四条边框中的某一个边框。

属性：

border-top/bottom/left/right:width style color;

3.单属性定义

设置四条边框的某一个属性

属性：

border-width/style/color:值;

4.单边单属性

设置某一条边的某一个属性

border-方向-属性：值;

方向：top/bottom/left/right

属性：width/style/color

ex:

border-top-color:red;

练习：

1.设置四个方向的边框样式都为虚线

2.设置下边框的颜色为黄色

4.边框倒角



将元素的直角倒成圆角。

属性：border-radius

取值：

1.以px为单位的数字

2.百分比 x%(设置圆形 50%)

单角设置：垂直再水平

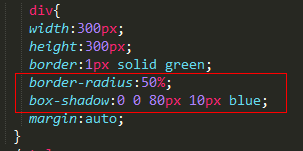
border-top-left-radius:左上角

border-bottom-left-radius:左下角

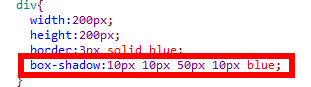
border-top-right-radius:右上角

border-bottom-right-raidus:右下角

5.边框阴影 （6属性）

day02 4

横竖0模糊80阴影10蓝



属性：box-shadow

取值：h-shadow v-shadow blur spread color inset（先水平）

h-shadow:阴影的水平偏移距离，必须值

取值为正，阴影向右偏移

取值为负，阴影向左偏移

v-shaodw:阴影的垂直偏移距离，必须值

取值为正，阴影向下偏移

取值为负，阴影向上偏移

blur:阴影模糊距离（可选值）

spread:阴影尺寸（可选值）

color:阴影颜色（可选值）

inset:将默认的外阴影改为内阴影（可选值）



box-shadow: 0 1px 6px rgba(0,0,0,.2);

练习：

1.设置圆形的外发光效果。

2.在溢出练习基础上，给每幅图设置5px的圆角，并且鼠标悬停在图片上时，设置向右下方5px的阴影效果，阴影颜色#ddd。

6.边框轮廓

轮廓指的边框的边框，绘制于边框外围的条线。

属性：

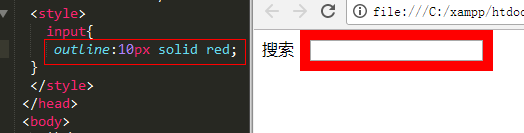
outline：width style color;

outline:0/none; 取消轮廓



练习：

在页面中创建一个文本框，尝试着取消轮廓，再观察效果。



一般为outline:none

## 5.框模型

1.什么是框模型

页面的元素皆为框

框模型：box model盒子模型，定义了元素框处理元素的内容，内边距，外边距以及边框的一种计算方式。

内边距：元素边框与内容区域之间的空白间距

外边距：元素与元素之间的空白间距

框模型默认的计算方式：

元素的实际占地宽度=左右外边距+左右边框+左右内边距+width

元素的实际占地高度=上下外边距+上下边框+上下内边距+height

2.外边距 margin

1.什么是外边距

围绕在边框外的空白间距

2.语法

属性：margin 定义四个方向的外边距

margin-top/bottom/left/right单边定义

取值：

1.以px为单位的数字

2.取值为负

margin-left

取值为正，元素往右移动

取值为负，元素往左移动

margin-top

取值为正，元素往下移动

取值为负，元素往上移动

3.百分比 %

外边距的值，占父元素的高或宽的比例

4.取值为auto

自动计算外边距的值

控制块级元素在水平方向居中对齐。

练习：

页面中设置两个div尺寸为200\*200，两个div之间的间距为50px,然后设置第二div水平居中显示。

3.简介写法

1.margin:value; 四个方向的外边距

2.margin:v1 v2;

上下 左右

3.margin:v1 v2 v3;

上 左右 下

4.margin:v1 v2 v3 v4;

上 右 下 左

4.自带外边距的元素

h1~h6,p,ul,ol,dl,pre,body

可通过CSS Reset（CSS重写）的方式，来重置具备外边距的元素

body,h1,h2,h3,h4,h5,h6,p,ul,ol,dl,dd,dt,pre{margin:0;}

5.外边距的特殊效果

1.外边距合并

当两个垂直外边距相遇时，他们将合并成一个，最终的距离取决于两个外边距中距离较大的那个。

2.上外边距溢出

在某些特殊的条件下，为子元素设置上外边距时，有可能会作用到父元素上。

1.父元素没有上边框

2.为第一个子元素设置上外边距时

解决方案：

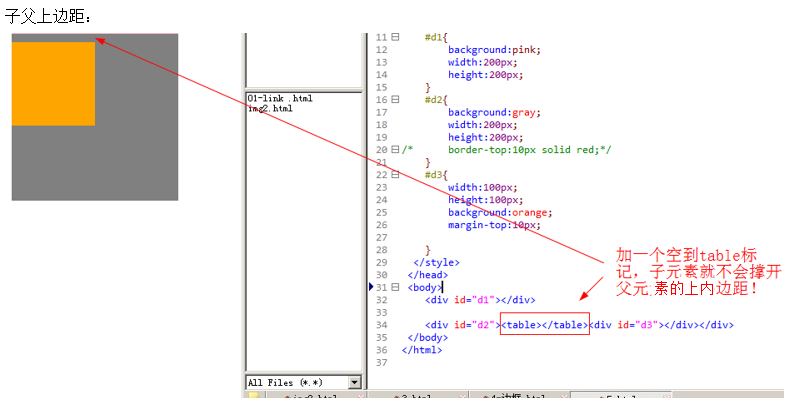
1.为父元素增加上边框

弊端：对父元素的高度有影响

2.使用父元素的上内边距取代子元素的上外边距

弊端：对父元素的高度有影响

3.在父元素的第一个子元素位置处，增加空<table></table>标记



3.行内元素以及行内块元素的外边距（两种情况）

1.行内元素垂直外边距无效（img除外），左右外边距相遇时会相加

2.行内块元素，设置垂直外边距时，整行元素都跟着发生改变

练习：

1.创建两个div元素，d1和d2,分别设置不同的背景颜色，尺寸都为200\*200；

2.在d2中再添加一个子元素div,id为d3,设置d3的尺寸为100\*100，并设置背景颜色；

3.设置d3的上外边距为50px,观察结果。

3.内边距

1.什么是内边距

元素边框与内容之间的空白间距

内边距会扩大元素边框的占地区域

2.语法

属性：

padding:value; 四个方向的内边距

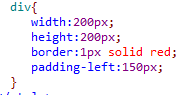
padding-top/bottom/left/right:value; 单边设置

取值：

1.以px为单位的数字

2.以%为单位的数字

3.简洁写法



padding:value

padding:v1 v2

上下 左右

padding:v1 v2 v3;

上 左右 下

padding:v1 v2 v3 v4;

上 右 下 左（顺时针）

练习：

创建两个div，d1和d2,尺寸200\*200，并设置边框为2px solid #f00;

为d1设置四个方向的内边距为30px，观察两个div效果。

4. box-sizing属性



（包括内外边距）

作用：指定框模型的计算方式

取值：

1.content-box

默认值，采用默认的计算方式。

实际占地宽度=左右外边距+左右边框+左右内边距+width

实际占地高度=上下外边距+上下边框+上下内边距+height

2.border-box

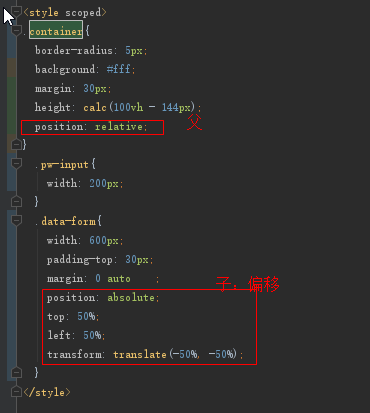
元素的尺寸，会包含border以及padding的值。

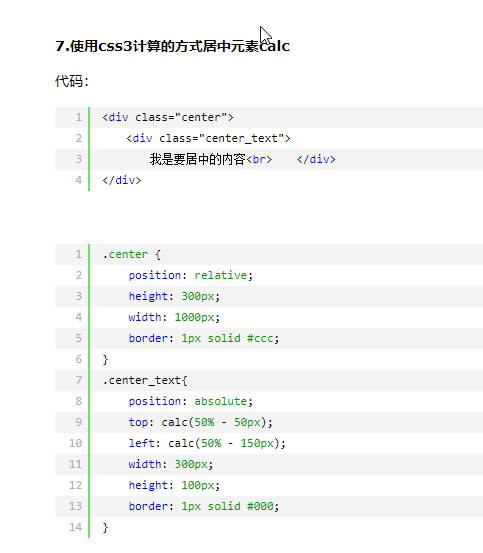
实际占地宽度=width(包含border和padding)+margin

实际占地高度=height(包含border和padding)+margin

5. 水平垂直居中 双向居中

设置子元素左上角top：50%，left：50%，transform向左向上偏移半个子元素的宽高





css day03

## 6.背景

1.背景色

属性：background-color

取值：合法的颜色值

注意：背景颜色和背景图片默认都从边框位置处开始填充

2.背景图片

属性：background-image

取值：url(图片的路径);

ex:

background-image:url(a.jpg);

3.背景图片平铺

属性：background-repeat

取值：

1.repeat 平铺

2.no-repeat 不平铺，只显示一次

3.repeat-x 只在横向平铺

4.repeat-y 只在纵向平铺

4.背景图片尺寸

属性：background-size

取值：

1.value1(宽) value2(高)

2.x% y%

3.cover 将背景图等比放大，直到背景图完全覆盖到元素的所有区域为止。

4.contain 将背景图等比放大，直到背景图碰到元素的某一个边缘为止。

5.背景图片的固定

属性：background-attachment

取值：

1.scroll 默认值，滚动

2.fixed 固定

6.背景图片定位

作用：改变背景在元素中的位置

属性：background-position

取值：

1.x y

x：背景图在水平方向的偏移距离

取值为正，向右移动

取值为负，向左移动

y：背景图在垂直方向的偏移距离

取值为正，向下移动

取值为负，向上移动

2.x% y%

0% 0% 背景在左上角

100% 100% 背景在右下角

50% 50% 背景在正中间

3.关键词

x: left/center/right

y: top/center/bottom

7.▲背景的简写属性(5 属性：颜色 url 平铺 固定 定位)（尺寸另写）

在一个属性中指定多个属性值

background:color url() repeat attachment position;

ex:

background:red;

background:url(a.jpg) no-repeat fixed center center;

## 7.渐变

1.什么是渐变

渐变指的是多种颜色平缓变化的一种显示效果。

2.渐变的主要因素

1.色标：一种颜色 及其 出现的位置

一个渐变是由多个色标组成（至少两个）

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

以圆形的方式来填充渐变色

3.重复渐变

将线性渐变 或 径向渐变 重复几次实现渐变填充

4.线性渐变linear-gradient

background-image:linear-gradient(60deg,red 0%,yellow 40%,pink 80%);

属性：backgrond-image

取值：linear-gradient(angle,color-point1,color-point2,...);

angle:表示渐变填充的方向或角度

取值：

1.关键词

to top 从下向上填充渐变色

to bottom 从上向下填充渐变色

to left 从右向左填充渐变色

to right 从左向右填充渐变色

2.角度值

0deg 从下向上(等同于to top)

180deg 从上向下(等同于to bottom)

color-point:色标

表示颜色及其位置

取值：颜色和位置的组和，中间用空格隔开。

ex:

red 0px;

blue 100px;

green 200px;

5.径向渐变radial-gradient

background-image:radial-gradient(200px at top left,blue 0,yellow 100px,green 200px);

属性：background-image

取值：radial-gradient([size at position],color-point1,color-point2,...);

size:半径，以px为单位的数字

position:圆心所在位置

1.x y:具体数字

2.x% y%:元素宽和高的占比

3.关键词

x:left/center/right

y:top/center/bottom

6.重复渐变

1.重复线性渐变

background-image:repeating-linear-gradient(angle,color-point,...);

color-point:位置一定要给绝对数值（px），不要用百分比。

2.重复径向渐变

backgrond-image:repeating-radial-gradient([size at position],color-point);

练习：

给一个div设置高宽各400px,然后设置重复线性渐变背景色。

字体渐变 图标颜色填充

<https://segmentfault.com/a/1190000012797252>



.font-block {

font-size: 48px;

background-image: linear-gradient(to bottom,#F80, #2ED);

-webkit-background-clip: text;

-webkit-text-fill-color: transparent;

}

7.浏览器兼容性-渐变

各浏览器的新版都支持渐变属性

对于不支持的浏览器版本，可以通过增加浏览器前缀的方式，让浏览器支持渐变。

Firefox:-moz-

Chrome&Safari:-webkit-

IE:-ms-

opera:-o-

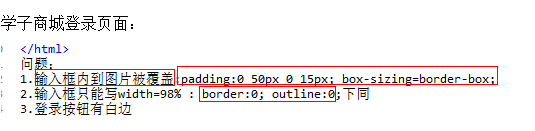
ex：

background-image:-webkit-linear-gradient();

兼容性查询网址：[www.caniuse.com](http://www.caniuse.com)

背景图片-学子商城登录页面 练习：





浏览器兼容：

1. 渐变属性加前缀：

（1） Firefox:-moz-

Chrome&Safari:-webkit-

IE:-ms-

opera:-o-

ex：

background-image:-webkit-linear-gradient();

（2）@-moz-keyframes

兼容性查询网址[www.caniuse.com](http://www.caniuse.com)

2.浏览器默认的margin和padding不同。解决方案是加一个全局的\*{margin:0;padding:0;}来统一。

3. 父子标签间用margin的问题，子前加空table

4. 父子关系的标签，子标签浮动导致父标签不再包裹子标签。子末尾加块级元素，clear：both

5. IE8 api兼容isArray

6.查：浏览器内核名称版本 排布情况

## 8.文本格式化属性

1.字体属性

1.指定字体

属性：font-family

取值：用逗号隔开字体名称列表

ex:

font-family:"微软雅黑","黑体",Arial;

2.字体大小

属性：font-size

取值：以px或pt为单位的数值

3.字体加粗

属性：font-weight

取值：

1.bold 加粗(b标记)

2.normal 正常

3.value 整百倍的数字 400-900

4.字体样式

属性：font-style

取值：

1.italic 斜体(i标记)

2.normal 正常

5.小型大写字母

将小写字符变为大写，但大小与小写字符一致。

属性：font-variant

取值：

1.normal 正常

2.small-caps 小型大写字符

练习：

在网页中定义一个div元素，并设值id为main,内容随意，中英文结合

设置以下格式：

1.字体：微软雅黑

2.文字大小:24px

3.加粗并且倾斜显示文本

4.将所有小写字符转为小型大写字符

6.▲字体属性的简写方式（6连写）

属性：

font:style variant weight size/lineheight family;

注意：使用简写方式时，必须要设置family的值，否则无效。

ex:font:24px;//错误

font:24px "微软雅黑";//正确

2.文本格式化

1.文本颜色

属性：color

取值：合法的颜色值

2.文本排列

作用：指定文本（行内，行内块）在水平方向的对齐方式。

属性：text-align

取值：left/center/right/justify(两端对齐)

3.线条修饰

属性：text-decoration

取值：

1.none 无任何线条修饰

2.underline 下划线

3.overline 上划线

4.line-through 删除线

4.行高

作用：一行数据所占的高度

如果行高大于字体本身的大小，那么改行文本将在指定的行高内呈现垂直居中的效果。

属性：line-height

取值：以px为单位的数值

5.首行文本缩进

属性：text-indent

取值：以px为单位的数值

6.文字阴影

属性：text-shadow

取值：h-shadow v-shadow blur color;





练习：

在上面文本练习的基础上，增加以下效果：

1.文本排列方式为两端对齐

2.文本的修饰为下划线

3.行高设为100px,并观察设完行高后的文本效果

4.文本阴影，水平和垂直偏移0px，模糊距离为3px，颜色红色

7.文字不可选中

.noneSelect {  
 -webkit-user-select:none;  
 -moz-user-select:none;  
 -ms-user-select:none;  
 user-select:none;  
}

1. 字符串过长省略 长字符串省略

****word-break:keep-all;**** /\* 不换行 \*/   
****white-space:nowrap;**** /\* 不换行 \*/     
****overflow:hidden;**** /\* 内容超出宽度时隐藏超出部分的内容 \*/     
****text-overflow:ellipsis;/****\* 当对象内文本溢出时显示省略标记(...) ；需与overflow:hidden;一起使用。\*/

## 9.表格

1.表格的常用属性

1.边距属性：padding

2.边框属性：border

3.尺寸属性：width,height

4.文本格式化属性：font-\*，text-\*,line-height

5.背景属性：颜色，图片，渐变

6.vertical-align

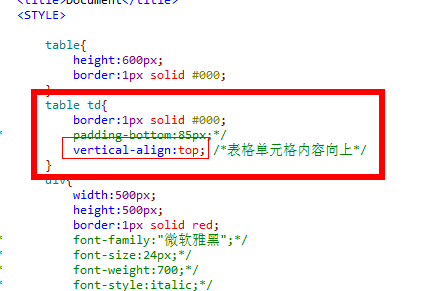
作用：指定单元格数据的垂直对齐方式

取值：

top:上对齐

middle:居中对齐

bottom:下对齐



练习：

创建一个表格尺寸为400\*400px,4行4列

1.创建每个单元格的尺寸为100\*100

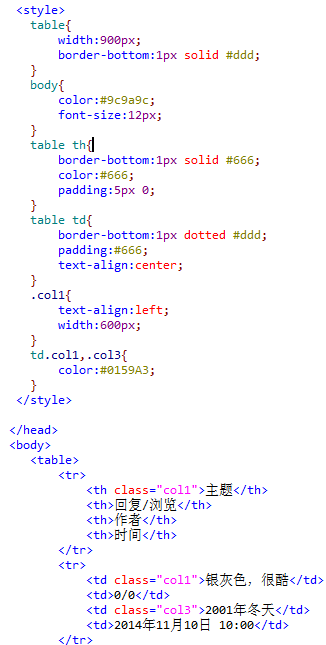
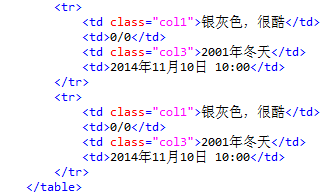
2.设置表格和单元格的边框为1px solid #000

3.尝试着为每个td增加上外边距10px.

css-表格-学子商城数据练习：

标注顺序，样式有节凑，代码少！



 +

css day04

2.表格css属性



1.边框合并

属性：border-collapse

取值：

1.separate 默认值，即边框分离模式



2.collapse 边框合并

2.边框边距

作用：设置单元格之间或单元格与表格之间的距离。

属性：border-spacing

取值：

1.给一个值：水平和垂直的间距相同

2.给两个值：

第一个值表示水平间距

第二个值表示垂直间距

注意：

只有在分离边框模式下，边框边距才有效，即border-collapse:separate;时，border-spacing才有效果。

3.表格标题位置

属性：caption-side

取值：

1.top 默认值，标题在表格的内容之上

2.bottom 标题位于表格内容之下

4.显示规则--表格尺寸固定

作用：用来帮助浏览器指定如何布局一张表，也就是指定td尺寸的计算方式。

属性：table-layout

取值：

1.auto 默认值，即自动表格布局，列的尺寸实际上是由内容来决定的。

2.fixed 固定表格布局，列的尺寸由设定的值为准。

自动布局VS固定布局：

1.自动布局

1.单元格的大小会适应内容

2.表格复杂时，加载速度较慢（缺点）

3.自动布局会比较灵活（优点）

4.适用于表格不太复杂并且不确定每一列大小时使用

2.固定布局

1.单元格的尺寸取决于你设定的值

2.任何情况下都会加速显示表格（优点）

3.固定布局不够灵活（缺点）

4.适用于确定每列大小时使用

## 10.定位-浮动定位（重点）

1.定位

定位：指的是改变元素在页面中的默认位置（出现在应该出现的位置）。

2.定位的分类

按照定位的效果，可以分成以下几种：

1.普通流定位（默认定位方式）

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

普通流定位，又称为“文档流定位”，页面中元素们的默认显示方式。

1.每个元素在页面中有自己空间

2.默认都是在父元素的左上角开始显示

3.块级元素都是按照从上到下的方式逐个排列，每个元素独占一行。

4.行内元素以及行内块都是按照从左往右的方式排列，多个元素在一行中显示。

问题：如何让多个块级在一行中显示？

4.浮动定位（重点）

1.什么是浮动&特点

如果将元素的定位方式设置为浮动后，元素将具备以下特点：

1.元素将被排除在文档流之外（脱离文档流），不再占据页面空间，未浮动元素上前补位。

2.浮动元素只能在当前行浮动。

3.浮动元素停靠在其父元素的左边或右边，或其他已经浮动的元素的边缘上。

2.语法

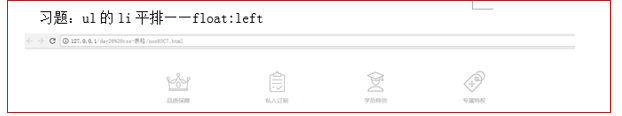
属性：float

取值：

1.left 左浮动，让元素停靠在父元素的左边，或挨着左侧已有的浮动元素。

2.right 右浮动，让元素停靠在父元素的右边，或挨着右侧已有的浮动元素。

3.none 默认值，无浮动



3.浮动引发的特殊效果

1.当父元素显示不下所有以浮动元素时，最后一个将换行，但是，有可能被卡主。

2.元素一旦浮动后，宽度将以内容为主（未指定宽度情况下）

3.元素一旦浮动起来后，将变为块级元素

允许修改尺寸；

能正常处理垂直外边距。

4.文本，行内元素，行内块元素采用环绕的方式来排列，是不会被浮动元素压在地下 的，而是巧妙避开。

练习：

添加两个span元素，内容随意，让sapn浮动起来，并设置高宽属性，查看效果

4.清除浮动带来的影响

元素一旦浮动起来后，就会对后续元素带来一定的影响（后续元素上前补位），如果后续元素不想被影响（不想占位），那么就可以通过清除浮动来解决。

属性：clear

取值：

1.left 清除当前元素前面元素的左浮动带来的影响

2.right 清除当前元素前面元素的右浮动带来的影响

3.both 清除当前元素任何一种浮动带来的影响

4.none 默认值，不做任何清除操作

5.浮动元素对父元素的影响

元素的高度，都是以未浮动元素的高度为准的，浮动元素是不占高度的。

解决父元素高度的方案：

1.直接设置父元素的高度

弊端：不是每次都知道父元素的高度

2.设置父元素也浮动

弊端：不是任何时候父元素都需要浮动，而且浮动会影响后续元素

3.为父元素设置overflow

取值：hidden或auto

弊端：如果有内容需要溢出显示，也会一同被隐藏

4.在父元素中，追加一个空子元素(块级)，并设置其clear:both。

## 11.显示

1.显示方式

1.什么是显示方式？

决定了元素在页面中的表现形式（行内，块级，行内块，table）

2.语法

属性：display

取值：

1.none 不显示元素-隐藏

脱离文档流，不占页面空间。

2.block 让元素表现的与块级元素一致

允许设置高宽

3.inline 让元素表现的与行内元素一致

4.inline-block 让元素表现的与行内块元素一致

特点：允许修改尺寸

多个元素在一行中显示

5.table 显示为表格

特点：尺寸以内容为准

每个元素独占一行

允许修改尺寸

练习：

在页面中创建5个a标记，并写内容和href属性

1.分别为a标记设置不同的显示方式

none,block,inline,inline-block,table

2.给所有的a元素设置高宽各200px,然后查看效果。

2.显示效果

1.显示/隐藏

属性：visibility

取值：

1.visible 默认值，元素可见

2.hidden 隐藏，元素不可见

面试题：dispaly:none和visibility:hidden区别？

dispaly:none 会脱离文档流，不占页面空间

visibility:hidden 不脱离文档流，占据空间

3.透明度--（元素）

属性：opacity

取值：0.0-1.0

注意：rgba() 与 opacity的区域

opacity作用于元素，以及元素内的所有内容的透明度。而rgba只作用于某一个属性的透明度（元素颜色或其背景颜色）。

4.垂直方向对齐方式

属性：vertical-align

场合：

1.表格中使用

取值：top/middle/bottom

2.图片中使用

作用：控制图片两边的文本的垂直对齐方式

取值：

top 顶部

middle 居中

bottom 底部

baseline 默认值，基线对齐

注意：

编写网页时，通常都会将所有图片的默认对齐方式更改为除baseline以外的对齐方式。

5.光标（鼠标）鼠标样式

1.作用

改变鼠标悬停在元素上时，鼠标的状态

2.语法

属性：cursor

取值：

1.default 箭头

2.pointer 小手

3.text I

4.wait 等待

5.crosshair +

6.help 帮助

待查：如何自定义鼠标光标？

## 12.列表

1.列表项标识

属性：list-style-type

取值：

1.none

2.disc

3.circle

4.square

2.列表项图像

作用：使用自定义图像作为列表标识

属性：list-style-image

取值：url(图片路径);

3.列表项的位置

属性：list-style-position

作用：将默认的列表项标识的位置放到li的里面。

取值：

1.outside 默认值，将标识放置li外面

2.inside 将标识放置于li里面

4.列表的简写方式

属性：list-style

取值：type url() position;

常用方式：

list-style:none;

练习：

完成如图所示效果

## 13.定位-相对 绝对 固定

1.定位属性

属性：position

取值：

1.static 静态的，默认值

2.relative 相对定位

3.absolute 绝对定位

4.fixed 固定定位

2.偏移属性

top/bottom/left/right

以上偏移属性取值均为数字(px)

ex:

top:100px; 向下偏移100px

top:-50px; 向上偏移50px

left:150px; 向右偏移150px

right:-150px; 向右偏移150px

3.定位方式

1.相对定位

1.什么是相对定位

元素会相对于它原来的位置偏移某个距离

2.使用场合

在做位置微调时使用

3.语法

position:relative;

配合偏移属性来实现位置的微调。

练习：4-7

将上面列表练习中图片加上相对定位，向左上方偏移10px.

2.绝对定位

1.语法：

position:absolute;

配合偏移属性 实现位置的修改

2.使用场合

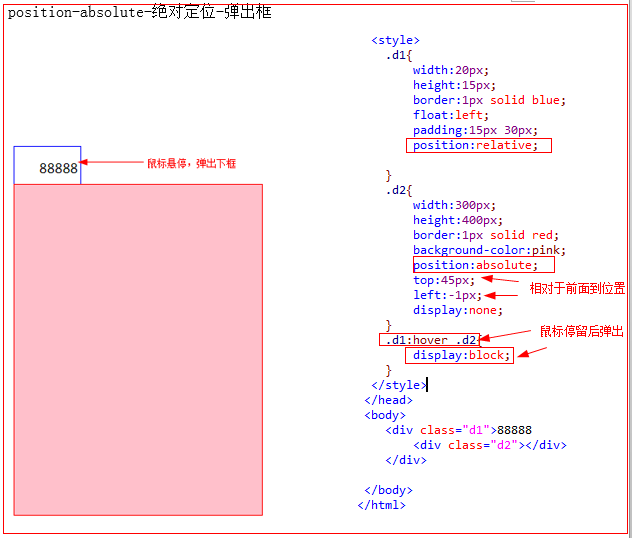
有堆叠顺序的元素

弹出菜单

3.特点

1.绝对定位的元素会相对于离他最近的 已定位的祖先元素 去实现位置的初始化，如果没有已定位的祖先元素，那么元素就相对于body去实现位置的初始化。

2.元素定位以后会脱离文档流，不占据页面空间。





css day05

3.堆叠顺序 5 1

一旦将元素变为已定位元素的话，元素们则有可能出现堆叠的效果。

如何改变堆叠顺序？

属性：z-index

取值：无单位的数字，数字越大越靠上。

注意：

1.父子元素间，z-index无效，永远都是子元素压在父元素上。

2.只有已定位元素才能使用z-index属性。

4.固定定位5 2

1.什么是固定定位

将元素固定在页面中的某个位置，位置不会随着滚动条而发生改变，始终固定在可视化区域。

2.语法

position:fixed;

配合偏移属性使用

3.特点

1.固定定位的元素会变成块级元素

2.脱离文档流，不占页面空间

3.固定定位的元素永远都是相对于body去实现位置的初始化。

练习：

在网页的右下角固定一张图片，距离底部100px,距离右边50px,页面滚动时，图片不动。

=======================

# 二．CSS3高级

1.复杂选择器

4.兄弟选择器5 3

兄弟元素：具备相同父元素的平级元素称之为兄弟元素。

ex:

<div>

<p></p>

<p></p>

<p></p>

<p></p>

</div>

1.相邻兄弟选择器+

作用：获取紧紧挨在某元素后的一个兄弟元素

语法：选择器1+选择器2{}

2.通用兄弟选择器~

作用：获取某元素后所有满足条件的兄弟元素

语法：选择器1~选择器2{}

练习：通过兄弟选择器设置以下样式

1.h3与div之间有30px的外边距；

2.从第二个div开始，每个div都有10px的上外边距。

<h3>静夜思</h3>

<div>窗前明月光</div>

<div>疑是地上霜</div>

<div>举头望明月</div>

<div>低头思故乡</div>

5.属性选择器5 4

1.作用

允许通过元素的属性及其值匹配页面中的元素。

2.语法

1.基本语法

[attr]

作用：匹配页面中所有附带attr属性的元素

ex：

[id] 匹配页面中所有附带id属性的元素

[class] 匹配页面中所有附带class属性的元素

2.elem[attr]

elem:表示任意元素

attr:表示任意属性

作用：匹配页面中所有附带attr属性的elem元素

ex:

div[id] 匹配页面中所有附带id属性的div元素

3.[attr1][attr2]...

作用：匹配同时附带多个属性的元素

ex:

[id][class]匹配页面中同时附带id和class属性的元素。

4.[attr=value]

作用：匹配attr属性值为value的元素

ex:

[id="map"]匹配id属性为map的元素

[class="c1"]

6.伪类选择器

已学过的伪类：

链接伪类： :link , :visited

动态伪类： :hover , :active , :focus

1.目标伪类 选择器:target{...} 5 2/5 4

作用：突出显示活动的HTML锚点元素，匹配被激活时的锚点。

语法：

选择器:target{...}

ex:

div:target{}

练习：

用目标伪类选择器完成如图所示效果

2.结构伪类

作用：通过元素的结构关系来匹配元素

1.:first-child其父元素的第一个子

匹配的元素是属于其父元素的第一个子元素

<div>

<p>p1</p>

<p>p2</p>

<p>p3</p>

<p>p4</p>

</div>

ex:

p:first-child{}

2.:last-chlid其父元素中的最后一个子

匹配的元素是属于其父元素中的最后一个子元素

ex:

p:last-child{}

练习：

创建一个4\*4的表格，并填充内容

设置第一行数据的背景颜色为黄色

设置最后一行数据的背景颜色为红色

3.:nth-child(n) 第n个子

匹配的元素是属于其父元素中第n个子元素

ex:

tr:nth-child(3)

练习：

将第二行中第3列的背景颜色设置为蓝色

4.:empty没有子

匹配没有子元素的元素

5.:only-child父中唯一的子

匹配的元素属于其父元素中的唯一的子元素才选。

3.否定伪类

将满足指定选择器的元素给排除在外

语法：

:not(selector){}

tr:not(:first-child){} 不是第一的所有子

练习：

设置每行中除第一列以外的所有列，文字以加粗和斜体的方式显示。

7.伪元素选择器

1.:first-letter或::first-letter元素的首字符

作用：匹配某个元素的首字符

2.:first-line或::first-line首行

作用：匹配某个元素的首行

注意：首字符和首行样式冲突时，优先使用首字符的样式

3.::selection 被选取的内容

作用：匹配被用户选取的内容

注意：只能修改本文颜色和背景颜色

练习：

创建一个div，id为d1,内容随意

1.将div中内容的首字符设置为加粗，红色，大小为20pt,右外边距15px

2.将div中首行文本设置为橙色，斜体

3.将用户选取的内容设置为红色背景黄色字体

8.内容生成5 7

1.什么是内容生成

使用css动态的向某元素中插入一段内容

2.伪元素选择器

1.:before或::before

作用：匹配到某元素的内容区域之前

<p>(:before)锄禾日当午</p>

2.:after或::after

作用：匹配到某元素的内容区域之后

<p>汗滴禾下土(:after)</p>

3.属性：content文字或url图片 display

属性：content

作用：向匹配的位置处增加内容

取值：

1.字符串 用""引起来

2.url() 添加一张图片

练习：

<p>学而时习之，不亦说乎</p>

<p>有朋自远方来，不亦乐乎</p>

输入格式如下：

子曰：学而时习之，不亦说乎

-摘自《论语》

9.解决问题

1.外边距溢出5 8 3 2 （子外边距，父前加空table放溢出）

1.增加一个空table标记-显示为table

2.在父元素的第一个子元素位置处增加(:before)

选择器:before{

content:"";

display:table;

}

2.浮动给父元素高度带来的影响4 2（子浮动，父不浮：父后加清除浮动的空块元素）

解决问题：

选择器:after{

display:block;

content:"";

clear:both;

}

14.弹性布局 5 10

1.什么是弹性布局

弹性布局是一种布局方式，主要是解决某元素中"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“容器”

1.弹性布局的容器

简称“容器”，要实现布局效果的父元素，称为容器。

2.弹性布局的项目

简称“项目”，要实现布局效果的子元素，称为项目

3.主轴

项目们排列方向的一根轴，就称为主轴

如果项目们按x轴排列，那么x轴就是主轴

如果项目们按y轴排列，那么y轴就是主轴

4.交叉轴

与主轴交叉的一根轴称为交叉轴

如果主轴是x轴，那么y轴就是交叉轴

如果主轴是y轴，那么x轴就是交叉轴

3.语法

1.flex的容器display: flex

将元素变为flex容器后，那么所有的子元素都将变成flex项目,都允许按照弹性布局的方式排列。

属性：display

取值：

1.flex 将块级元素变为容器

2.inline-flex 将行内元素变为容器

2.容器的属性 方向 换行 对齐方式\*2

1.flex-direction 方向

作用：指定容器的主轴及其排列方向

取值：

1.row 默认值，即主轴是x轴，起点在左端

2.row-reverse 主轴是x轴，起点在右端

3.column 主轴是y轴，起点在顶端

4.column-reverse 主轴是y轴，起点在底端

2.flex-wrap换行

作用：当一个主轴排列不下所有项目时，换行显示

取值：

1.nowrap 默认值，即空间不够时，也不换行，项目自动缩小。

2.wrap 换行显示

3.wrap-reverse 换行反转

3.flex-flow 缩写:方向+换行

作用：是flex-direction和flex-wrap的缩写形式

取值：

1.row nowrap 默认值

2.direction wrap

4.justify-content 主轴上对齐方式

作用：定义项目在主轴上的对齐方式

取值：

1.flex-start 在主轴的的起点对齐

2.flex-end 在主轴的终点对齐

3.center 在主轴上居中对齐

4.space-between 两端对齐

5.space-around 每个项目两端间距相同

5.align-items 交叉轴上的对齐方式

作用：项目在交叉轴上的对齐方式

取值：

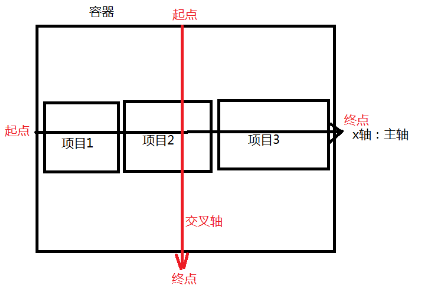
1.flex-start 交叉轴的起点对齐

2.flex-end 交叉轴的终点对齐

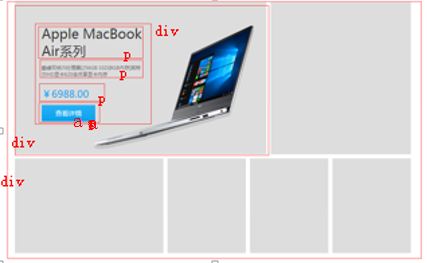
3.center 交叉轴的居中对齐

4.baseline 交叉轴的基线对齐

5.stretch 如果项目未设置尺寸，在交叉轴上占满所有空间。



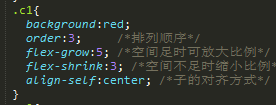
课后作业：完成学子商城 楼层布局效果



css day06

3.子项目的属性6 1

该组属性只能设置在某项目元素上，只控制一个项目，是不影响容器以及其他项目的效果。



1.order 排列顺序

作用：定义项目的排列顺序，值越小，越靠近起点，默认值是0

取值：整数数字，无单位

2.flex-grow 放大比例

作用：定义项目的放大比例，如果容器有足够的空间，项目可以放大。

取值：

整数数字，无单位。默认为0，不放大。

取值越大，占据剩余的空间越多。

3.flex-shrink容器空间不足时缩小比例

作用：定义项目的缩小比例，即容器空间不足时，项目可以缩小。缩小的比例取决于flex-shrink的值。

取值：默认值为1，空间不足时，则等比缩小

取值为0，则不缩小。

4.align-self 子的对齐方式

作用：定义项目在交叉轴上的对齐方式

取值：

flex-start

flex-end

center

baseline

stretch

auto 继承父元素的align-items的效果

14.1. vh布局

1.响应式宽高





15.转换

1.什么是转换

改变元素在页面中的位置，大小，角度以及形状的一种方式。

2D：只在x轴和y轴上发生的转换效果

3D：增加z轴的转换效果

2.转换属性

1.转换属性 transform: translate(x) scale(value) rotate(ndeg) skewX(xdeg);

属性：transform

取值：

1.none 默认值，无任何转换效果

2.transform-function

表示1个或多个转换函数（如果是多个转换函数的话，中间用空格隔开）

2.转换原点 transform-origin: (50%，50%)

属性：transform-origin

取值：数值/百分比/关键字

2个值：表示原点在x轴和y轴上的位置

3个值：表示原点在x轴，y轴和z轴上的位置

默认的原点在元素的中心位置处。

(50%，50%)

(center,center)

3.2D中转换 4个函数 位移 缩放 旋转 倾斜6 3

1.位移transform 6 2

1.什么是位移

改变元素在页面中的位置

2.语法

属性：transform

取值：

1.translate(x)

指定元素在x轴上位移距离

取值为正，元素右移

取值为负，元素左移

2.translate(x,y)

x:同上

y:指定元素在y轴上的位移距离

取值为正，元素下移

取值为负，元素上移

3.translateX(x)

4.translateY(y)

练习：在页面中添加一个div元素，设置元素往右位移120px,往下位移100px。

2.缩放

1.什么是缩放

改变元素在页面中的尺寸

2.语法

属性：transform

取值：

1.scale(value)

value 横向和纵向的缩放比例

value 默认值为1

>1 表示放大

<1 表示缩小

负数：放大（水平和垂直方向都翻转180度）

2.scale(x,y)

x：横向缩放比例

y：纵向缩放比例

3.scaleX(x)

4.scaleY(y)

3.旋转 6 2

1.什么是旋转

改变元素在页面上角度

2.语法

属性：transform

取值：

1.rotate(ndeg)

n取值为正，顺时针旋转

n取值为负，逆时针旋转

注意：

1.转换原点会影响最后的转换效果

2.旋转是连同坐标轴一同旋转的，会影响旋转后的位移效果。

练习：

1.将d1先旋转45deg，再向右位移50px

2.将d2先位移50px，再旋转45deg，对比效果

4.倾斜

1.什么是倾斜

改变元素在页面中的形状

2.语法

属性：transform

取值：skewX(xdeg)

让元素向着x轴产生倾斜，实际上改变的是y轴的倾斜角度

x取值为正，y轴逆时针倾斜

x取值为负，y轴顺时针倾斜

skewY(ydeg)

让元素向着y轴产生倾斜，实际上改变的是x轴的倾斜角度

y取值为正，x轴顺时针倾斜

y取值为负，x轴逆时针倾斜

skew(x,y)

skew(x) 等同于skewX(x)

练习：

页面上创建一个200\*200的div，设置背景颜色，鼠标悬停时，该元素向右偏移200px,向下偏移200px,旋转135deg,向x轴倾斜45deg,同时放大1.5倍，查看效果。

4.3D转换 6 4

1.透视距离 父元素上perspective:200px;

模拟人的眼睛到3D转换元素之间的距离

属性：perspective

取值：以px为单位的数字

注意：该属性要加在3d转换元素的父元素上

2.3D旋转transform :rotateX(xdeg)

属性：transform

取值：

1.rotateX(xdeg)

以x轴为中心轴，旋转元素的角度

2.rotateY(ydeg)

以y轴为中心轴，旋转元素的角度

3.rotateZ(zdeg)

以z轴为中心轴，旋转元素的角度

4.rotate3D(x,y,z,ndeg)

x,y,z取值为大于0的数字时，表示该轴参与旋转，取值为0则不参与旋转。

16.过渡

1.什么是过渡

使得CSS属性值，在一段时间内平缓变化的效果。

2.过渡属性

1.指定过渡属性

作用：指定哪个属性值在发生改变时需要使用过渡效果来体现。

属性：transition-property

取值：

1.all 能使用过渡的属性，一律用过渡体现

2.具体属性名

ex:

transition-property:background;

transition-property:border-radius;

允许设置过渡效果的属性：

1.颜色属性（背景颜色，字体颜色，边框颜色，阴影颜色）

2.取值为数字的属性

3.转换属性

4.阴影属性

5.渐变属性

6.visibility属性

2.指定过渡时长

作用：指定在多长时间内完成过渡操作

属性：transition-duration

取值：以s或ms为单位的数字

1000ms=1s

3.指定过渡的时间曲线函数

属性：transition-timing-function

取值：

1.ease 默认值，慢速开始，快速变快，慢速结束

2.linear 匀速

3.ease-in 慢速开始，加速结束

4.ease-out 快速开始，慢速结束

5.ease-in-out 慢速开始和结束，中间先加速后减速

4.指定过渡延迟时间

属性：transition-delay

取值：以s或ms为单位的数字

注意：

1.将过渡放在元素声明的样式中

既管去，又管回

2.将过渡放在触发的操作中（hover）

只管去，不管回

5.过渡属性的缩写transition：property duration timing-function delay;属性 时长 变化方式 延迟

ex:

transition:all 2s linear 2s;

练习：

1.在页面中创建一个100\*100的元素

2.鼠标悬停时，5s之内，向右偏移500px同时再旋转3圈。

3.鼠标移出时，还能滚回来。

17.动画

1.什么是动画

使用元素从一种样式逐渐变化为另一种样式

即将多个过渡效果放在一起

动画是通过关键帧，来控制动画的每一步。

"关键帧"：

1.动画执行的时间点

2.在该时间点上的样式是什么

2.动画的使用步骤

1.声明动画

为动画起名

定义关键帧

@keyframes 动画名称{

/\*定义动画所有的关键帧\*/

0%{

/\*动画开始时元素的样式\*/

}

50%{

/\*动画执行到一半时元素的样式\*/

}

100%{

/\*动画结束时的样式\*/

}

}

兼容性：低版本浏览器4种：

firefox:@-moz-keyframes

IE:-ms-

chrome&safari:-webkit-

opera:-o-

2.为元素调用动画

指定元素调用的动画名称

调用动画属性：

1.animation-name

作用：指定调用动画名称

2.animation-duration

作用：指定动画执行一个周期的时长

取值：以s或ms为单位的数字

3.animation-timing-function

作用：指定动画的速度事件曲线函数

取值：ease,linear,ease-in,ease-out,ease-in-out;

4.animation-delay

作用：指定动画的延迟时间

取值：以s或ms为单位的数字

5.animation-iteration-count

作用：指定动画的播放次数

取值：

1.具体数字

2.infinite 无限次播放

6.animation-direction

作用：指定动画的播放方向

取值：

1.normal 默认值，正常播放0%-100%

2.reverse 逆向播放100%-0%

3.alternate 轮流播放

奇数次播放时，正常播放

偶数次播放时，逆向播放

7.动画简写方式

animation:name duration timing-function delay iteration-count direction;

ex:

animation:change 3s;

animation:change 3s linear 2s infinite reverse

动画：名称 时长 方式 延迟 次数 方向

8.animation-fill-mode: forwards 播放后的状态第一帧

作用：规定动画在播放前或播放后的状态

取值：

1.none 默认值

2.forwards 动画播放完之后，将保持在最后一个帧的状态上。

3.backwards 动画播放前，延迟时间内，动画保持在第一个帧的状态上。

4.both forwards和backwrads的结合体

9.animation-play-state:paused/running 暂停/播放

作用：指定动画是处于播放还是暂停的状态上

取值：

1.paused 暂停

2.running 播放

18.Rem布局 (王蒙：淘宝支配方案）

①

下载引用：flexible.js

然后按设计尺寸除以75就是我的单位长度

举例：750宽度： 就是10rem

② 如果觉得每个尺寸都除75麻烦，可以下载VScode插件：px2rem，把基准设置为75就行，直接写设计尺寸，会有提示，帮你计算好。

1. 尺寸计算

width: **calc**(100% - 91px);

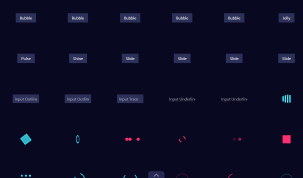
# 三．H5（原生js）

## 1.公共项目

### 1.公共页面样式

#### 1.button

<https://cssfx.dev/>



#### 2.loading

<https://cssfx.dev/>



## 2.pc

### 1.项目1

### 2.项目2

## 3.app

### 1.项目1

### 2.项目2

## 4.移动h5

### 1.项目1

### 2.项目2

# 四．jq

## 1.标题2.1

### 1.标题3.1 （3级目录红）

2.附加3.2

#### 1.标题4.1

2.附加4.2

##### 1.标题5.1

###### 1.标题6.1

1.标题7.1

1.标题8.1

1.标题9.1 法沙发撒飞洒发发

# 五．vue

## 1.公共项目

## 2.pc

### 1.项目1

### 2.项目2

## 3.app

### 1.项目1

### 2.项目2

## 4.移动h5

### 1.项目1

### 2.项目2

# 六．react

## 1.公共项目

## 2.pc

### 1.项目1

### 2.项目2

## 3.app

### 1.项目1

### 2.项目2

## 4.移动h5

### 1.项目1

### 2.项目2

# 七．anglar

## 1.公共项目

## 2.pc

### 1.项目1

### 2.项目2

## 3.app

### 1.项目1

### 2.项目2

## 4.移动h5

### 1.项目1

### 2.项目2