CSS

CSS概述

什么是CSS？

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

- CSS的作用

  设置HTML网页元素的样式（修饰元素）

  HTML：构建网页的结构

  CSS：构建网页元素的样式

- HTML与CSS之间使用原则

  W3C建议尽量使用CSS属性来取代HTML属性修饰元素。

- **CSS的语法规范**

  使用CSS的样式的方式

  - **内联样式** (行内样式)

    特点：将CSS样式定义在HTML的开始标记中

    语法：`<any style="样式声明1;样式声明2;..."></any>`

    样式声明：

    由CSS样式属性和值来组成；

    属性名与值之间用冒号连接；

    多个样式声明之间用分号隔开

  常用属性：

  设置 **文本颜色** 的属性和值

  属性：color

  取值：blue, red....

  设置 **背景颜色** 的属性和值

  属性：background

  取值：颜色的英文单词

  设置 **文字大小** 的属性和值

  属性：font-size

  取值：以px为单位的数字（20px）

  - **内部样式**

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

  - **外部样式**

    独立于任何网页的位置处，声明一个样式表文件（.css为后缀）在.css文件中保存样式规则

使用步骤：

创建样式表文件（.css结尾）

在独立的样式表文件中编写样式规则

在网页中引入样式表文件 `<link rel="stylesheet" href="cssUrl">`

CSS的样式特征

- 继承性

  必须是父子（有层级嵌套关系）结构

  大部分的CSS样式是可以直接继承给子元素

- 层叠性

  可以为一个元素定义多个样式规则，样式规则不冲突时，可以同时都应用到元素上

- 优先级

  样式的优先级默认从低到高：

浏览器默认设置	低
外部样式表和内部样式表	中（就近原则）
内联样式	高

- 调整显示的优先级

  语法：样式属性值 `!important`;

  将 `!important` 加在谁的后面，就优先使用谁。

CSS 选择器

选择器	示例	示例说明	CSS
.class	.intro	选择所有class="intro"的元素	1
#id	#firstname	选择所有id="firstname"的元素	1
*	*	选择所有元素	2
element	p	选择所有<p>元素	1
element,element	div,p	选择所有 元素和<p>元素	1
element element	div p	选择 元素内的所有<p>元素	1
element>element	div>p	选择所有父级是 元素的 <p> 元素	2
element+element	div+p	选择所有紧接着 元素之后的<p>元素	2
[attribute]	[target]	选择所有带有target属性元素	2
[attribute=value]	[target=-blank]	选择所有使用target="-blank"的元素	2
[attribute~=value]	[title~=flower]	选择标题属性包含单词"flower"的所有元素	2
[attribute|=language]	[lang|=en]	选择一个lang属性的起始值="EN"的所有元素	2
:link	a:link	选择所有未访问链接	1
:visited	a:visited	选择所有访问过的链接	1
:active	a:active	选择活动链接	1
:hover	a:hover	选择鼠标在链接上面时	1
:focus	input:focus	选择具有焦点的输入元素	2
:first-letter	p:first-letter	选择每一个<P>元素的第一个字母	1
:first-line	p:first-line	选择每一个<P>元素的第一行	1
:first-child	p:first-child	指定只有当<p>元素是其父级的第一个子级的样式。	2
:before	p:before	在每个<p>元素之前插入内容	2
:after	p:after	在每个<p>元素之后插入内容	2
:lang(language)	p:lang(it)	选择一个lang属性的起始值="it"的所有<p>元素	2
element1~element2	p~ul	选择p元素之后的每一个ul元素	3
[attribute^=value]	a[src^="https"]	选择每一个src属性的值以"https"开头的元素	3
[attribute$=value]	a[src$=".pdf"]	选择每一个src属性的值以".pdf"结尾的元素	3
[attribute*=value]	a[src*="runoob"]	选择每一个src属性的值包含子字符串"runoob"的元素	3
:first-of-type	p:first-of-type	选择每个p元素是其父级的第一个p元素	3
:last-of-type	p:last-of-type	选择每个p元素是其父级的最后一个p元素	3
:only-of-type	p:only-of-type	选择每个p元素是其父级的唯一p元素	3

:only-child	p:only-child	选择每个p元素是其父级的唯一子元素	3
:nth-child(*n*)	p:nth-child(2)	选择每个p元素是其父级的第二个子元素	3
:nth-last-child(*n*)	p:nth-last-child(2)	选择每个p元素的是其父级的第二个子元素，从最后一个子项计数	3
:nth-of-type(*n*)	p:nth-of-type(2)	选择每个p元素是其父级的第二个p元素	3
:nth-last-of-type(*n*)	p:nth-last-of-type(2)	选择每个p元素的是其父级的第二个p元素，从最后一个子项计数	3
:last-child	p:last-child	选择每个p元素是其父级的最后一个子级。	3
:root	:root	选择文档的根元素	3
:empty	p:empty	选择每个没有任何子级的p元素（包括文本节点）	3
:target	#news:target	选择当前活动的#news元素（包含该锚名称的点击的URL）	3
:enabled	input:enabled	选择每一个已启用的输入元素	3
:disabled	input:disabled	选择每一个禁用的输入元素	3
:checked	input:checked	选择每个选中的输入元素	3
:not(*selector*)	:not(p)	选择每个并非p元素的元素	3
::selection	::selection	匹配元素中被用户选中或处于高亮状态的部分	3
:out-of-range	:out-of-range	匹配值在指定区间之外的input元素	3
:in-range	:in-range	匹配值在指定区间之内的input元素	3
:read-write	:read-write	用于匹配可读及可写的元素	3
:read-only	:read-only	用于匹配设置 "readonly"（只读）属性的元素	3
:optional	:optional	用于匹配可选的输入元素	3
:required	:required	用于匹配设置了 "required" 属性的元素	3
:valid	:valid	用于匹配输入值为合法的元素	3
:invalid	:invalid	用于匹配输入值为非法的元素	3

选择器优先级排序：

关于选择器的优先级，Google 资深web开发工程师Steve Souders

对CSS选择器的效率从高到低做了一个排序(效率排序，和权重不是一回事)：

1.id选择器（#myid）

2.类选择器（.myclassname）

3.标签选择器（div,h1,p）

4.相邻选择器（h1+p）

5.子选择器（ul < li）

6.后代选择器（li a）

7.通配符选择器（*）

8.属性选择器（a[rel="external"]）

9.伪类选择器（a:hover,li:nth-child）

上面九种选择器中ID选择器的效率是最高，而伪类选择器的效率则是最低

计算指定选择器的优先级:重新认识CSS的权重(权重排序，和效率不是一回事)

1.通配选择符的权值 0,0,0,0

2.标签的权值为 0,0,0,1

3.类的权值为 0,0,1,0

4.属性选择的权值为0,0,1,0

5.伪类选择的权值为 0,0,1,0

6.伪对象选择的权值为 0,0,0,1

7.ID的权值为 0,1,0,0

8.important的权值为最高 1,0,0,0

归纳为!important > 内联 > ID > 类 > 伪类 | 属性选择 > 标签 | 伪对象 > 通配符 > 继承

尺寸和边框

尺寸单位

1. px 像素 1024*768
2. pt 磅 (1/72in)字体大小
3. in 英寸 1in=2.54cm
4. cm 厘米
5. mm 毫米
6. em 相对单位（相对于父元素乘以倍数 2em）
7. rem 相对单位（相对于根元素字体大小乘以倍数 2rem）
8. % 相对单位（百分比）

颜色单位（取值）

- 颜色英文单词

  blue,red,black....
- rgb(r,g,b)

r:0-255；g:0-255；b:0-255

- rgba(r,g,b,alpha)

  alpha:透明度，取值0-1之间的小数

- #rrggbb / #rgb

尺寸属性

- 作用：改变（设置）元素的宽度和高度
- 语法
  - 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
```

页面中允许设置尺寸的元素

- 所有的块级元素都允许设置尺寸

  div,p,h1-h6,ul,ol,dl,dt,dd ...

- 行内块元素允许设置尺寸

  表单元素（单选按钮和复选框除外）

- 本身具备width和height属性的元素

  img,table

注意：大部分行内元素不允许设置尺寸

溢出处理

- 场景：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：
  - `hidden` 隐藏的，溢出的内容全部隐藏，不可见
  - `scroll` 显示滚动条，溢出时可用
  - `auto` 自动，溢出时才显示滚动条，可用
  - `visible` 可见的，默认值，溢出可见

边框属性

- 语法：`border: width style color;`
- 属性：
  - width: 边框的宽度，以px为单位的数字
- style: 边框的样式

  取值
  - solid 实线条
  - dotted 虚线条（点）
  - dashed 虚线条（线）
    - color: 边框的样色，合法的颜色值，也可以取值为transparent(透明色)
- 取消边框：
  - border:none/0;

边框倒角

- 作用：将直角倒成圆角
- 属性：`border-radius`
- 取值：
  - 以px为单位的数字
  - 百分比 50% 设置圆形

边框阴影

- 属性：`box-shadow`
- 取值：`h-shadow` `v-shadow` `blur` `spread` `color` `inset`
  - `h-shadow`: 阴影在水平方向的偏移，必须值

    取值为正，阴影右偏移

    取值为负，阴影左偏移
  - `v-shadow`: 阴影在垂直方向的偏移，必须值

    取值为正，阴影下偏移

    取值为负，阴影上偏移
  - `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`: 将默认外阴影设置为内阴影，可选值

轮廓

- 概念：轮廓指的是边框的边框，绘制于边框外围的一条线
- 属性:

  `outline:width style color;`

`outline:none/0;` 取消轮廓

框模型

- 什么是框模型

  框：页面元素皆为框

  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：边框与内容区域之间的间距

  外边距：边框与边框之间的间距（围绕在边框外的空白间距）

  框模型默认计算方式：

  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

- 外边距
  - 什么是外边距

    边框与边框之间的间距
  - 语法

    属性：

    margin 定义某个元素四个方向的外边距

    margin-top/bottom/left/right 定义某一个方向的外边距

    取值：

    1. 以px为单位的数字
    2. 取值为负数

```
    margin-left: 取值为正，让元素右移动；取值为负，让元素左移动
margin-top: 取值为正，让元素下移动；取值为负，让元素上移动
```

    3. 百分比 %

    外边距的值，相对于父元素高宽的占比

    4. 取值为auto

    自动计算外边距（控制块级元素在水平方向居中显示）

> 外边距的特殊效果
>
> - **外边距合并**
>
>   当两个垂直外边距相遇时，它们将合并成一个，最终的距离取决于两个外边距中距离较大的那个。
>
> - **行内元素以及行内块垂直外边边距**
>
>   1.行内元素垂直外边距无效（img）,水平外边距会相加
>
>   2.行内块设置垂直外边距时，整行元素都跟着发生改变

> - **外边距溢出**
>
>   在某些特殊条件，为子元素设置上外边距时，有可能作用到父元素上
>
>   ① 父元素没有上边框；② 为第一个子元素设置上外边距
>
>   解决方案：
>
>   1.为父元素设置上边框
>
>   弊端：对父元素的高度有影响
>
>   2.使用父元素的上内边距来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，增加一个空 <table></table>

框模型

- 属性: `box-sizing`
- 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=左右内边距+左右外边距+左右边框+width

   实际占地高度=上下内边距+上下外边距+上下边框+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

背景

- 背景色
  - 属性：`background-color`
  - 取值：合法的颜色值（transparent 透明色）

  注意：背景颜色和背景图片默认都从边框位置处开始填充
- 背景图像
  - 属性：`background-image`
  - 取值：url(图像的路径);

  ex: background-image:url(a.jpg);
- 背景图像平铺

    - 属性：background-repeat
    - 取值：
        1. `repeat` 默认值，横向和纵向都平铺
        2. `repeat-x` 只在横向平铺
        3. `repeat-y` 只在纵向平铺
        4. `no-repeat` 不平铺，图像只显示一次
- 背景图像尺寸
    - 属性：`background-size`
    - 取值：
        1. `width height`
        2. `width% height%`
        3. `cover` 将背景图等比放大，直到背景图完全覆盖到元素所有区域为止
        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
- 背景图像的固定
    - 作用：将背景图固定在某个位置，一直在可视化区域中，不会随着滚动条发送位置的变化。
    - 属性：background-attachment
    - 取值：
        1. `fixed` 固定
        2. `scroll` 默认值，滚动
- 背景图像的定位
    - 作用：改变背景图在元素中位置
    - 属性：background-position
    - 取值：
        1. `x y`

           x：背景图水平方向偏移距离

           取值为正，向右偏移

           取值为负，向左偏移

           y:背景图垂直方向偏移距离

           取值为正，向下偏移

           取值为负，向上偏移
        2. `x% y%`

           0% 0% 背景图在左上角

           50% 50% 在正中间

           100% 100% 右下角
        3. `关键词`

           x: left/center/right

           y: top/center/bottom
- 背景简写方式

- 属性：`background`
- 取值：`color url() repeat attachment position;`

  注意：如果不设置其中某个值的话，该位置将采用默认值

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

---

2. 渐变

1. 概念

渐变指的是多种颜色平缓变化的一种显示效果

2. 主要因素

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

3. 分类

1. 线性渐变

   以直线的方向来填充渐变色
2. 径向渐变

   以圆形的方式实现填充效果
3. 重复渐变

   将线性渐变和径向渐变重复多次实现填充效果

---

4. 线性渐变

属性：background-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　　从下到上

   to bottom　　从上到下

   to left　　从右到左

   to right　　从左到右
2. 角度值

color-point 色标

取值： 颜色以及位置的组合，中间用空格隔开

ex：

1. `red 0%` 在填充方向的开始位置处为红色
2. `green 50%` 到填充方向一半的位置处颜色变为绿色
3. `blue 200px` 到填充方向的200px位置处颜色变为蓝色

---

5. 径向渐变

属性：`background-image`

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

1. `size`：半径，以px为单位的数字
2. `position`：圆心所在位置

   取值：

   1. `x y`： 具体数字
   2. `x% y%` ： 元素宽高占比
   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);`

   注意：

   color-point 位置一定要给px，不用相对单位
2. 重复径向渐变

   `background-image: repeating-radial-gradient(size at position, color-point);`

---

7. 浏览器兼容性

各个浏览器的新版本支持渐变属性

对于不支持的浏览器，可以通过增加浏览器前缀的方式，让浏览器支持渐变

Chrome&Safari: -webkit-

Firefox: -moz-

Opera: -o-

IE: -ms-

文本格式化

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: "Microsoft YaHei", Arial, "宋体", "黑体";

2. 字体大小

属性：`font-size`

取值：一般以px或pt为单位的数字

3. 字体加粗

属性：`font-weight`

取值：

1. `bold/700` 加粗
2. `normal/400` 正常
3. `bolder/900` 更粗
4. `100 - 900`

4. 字体样式

属性：`font-style`

取值：

1. `normal` 正常
2. `italic` 斜体

5. 小型大写字母

将小写字母变为大写，但大小与小写字母一致

属性：`font-variant`

取值：

1. `normal` 正常
2. `small-caps` 小型大写字母

6. 字体属性简写

属性：`font`

取值：`style variant weight size family;`

注意：使用简写形式时，必须要设置 family 值，否则无效

---

2. 格式化属性

1. 文本颜色

属性：`color`

取值：合法颜色值

2. 文本对齐方式（水平）

作用：指定文本（行内，行内块）水平对齐方式

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` 无线条
2. `underline` 下划线
3. `overline` 上划线
4. `line-through` 中划线

4. 行高

作用：一行数据所占的高度

特点：如果行高大于文字本身大小，那么该行文本将在指定行高内呈现垂直居中的效果

属性：`line-height`

取值：以px为单位的数字

5. 首行文本缩进

属性：`text-indent`

取值：以px或em为单位的数字

6. 文本阴影

属性：`text-shadow`

取值：h-shadow v-shadow blur color;

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

单元格中文本的垂直对齐方式

取值：top / middle / bottom

---

2. 表格特有属性

1. 边框合并

属性：`border-collapse`

取值：

1. separate 默认值，即分离边框模式
2. collapse 边框合并

2. 边框边距

属性：`border-spacing`

作用：设置单元格之间或单元格与表哥之间的间距

取值：

1. x y;　水平和垂直边距
2. x;　水平=垂直边距

注意：只有在分离边框模式下（border-collapse: separate;），该属性有效

3.标题位置

属性：`caption-side`

取值：

1. top 默认值，标题在表格内容之上
2. bottom 在表格内容之下

4. 显示规则

属性：`table-layout`

作用：用来帮助浏览器指定如何布局一张表格，即指定td尺寸计算方式

取值：

1. fixed 固定表格布局，td尺寸由设定的为准
2. auto 默认值，自动进行表格布局，td尺寸实际由内容来决定

定位

1. 定位

指的是改变元素在页面中默认的位置

2. 分类

按照定位的效果，可以分为以下几类

1. 普通流定位（默认）

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”**，页面中元素的默认定位方式

1. 每个元素在页面中都有自己的空间
2. 每个元素默认都是从其父元素的左上角开始显示
3. 页面中的块级元素都是按照从上到下逐个排列，每个元素独占一行
4. 页面中的行内元素以及行内块都是按照从左到右来排列的

---

4. 浮动

1. 什么是浮动和特点

   如果将元素浮动起来以后，将具有以下特征：

   1. 元素一旦浮动起来将不占页面空间，脱离文档流定位，其他元素将上前部位
   2. 浮动元素会停靠在父元素的左边或右边，或者是其他已经浮动元素的边缘上

2. 语法：

属性：`float`

取值：

`left`：左浮动，让元素停靠在父元素左边或挨着左侧已浮动元素

`right`：右浮动，让元素停靠在父元素右边或挨着右侧已浮动元素

`none` ：默认值，无浮动效果

3. 浮动引发的特殊效果
4. 消除浮动带来的影响

属性：`clear`

取值：

`left / right / both`

> 浮动元素对父元素高度带来的影响
>
> 元素的高度，都是以未浮动元素的高度为准，浮动元素是不占高度的
>
> 解决父元素高度的问题：
>
> 1. 直接给父元素设置高度
>
> 缺点：不是每次都知道父元素的高度
> 2. 给父元素也设置浮动
>
> 缺点：会对其他元素造成影响
> 3. 为父元素设置 `overflow: hidden/auto`
>
> 缺点：会对容器内内容可能造成不必要的影响，溢出隐藏
> 4. 在父元素中，追加一个空标记，并设置为 `clear: both;`

显示

1. 显示方式

1. 什么是显示方式？
2. 语法

属性：`display`

取值：

1. `none` 不显示元素-隐藏
2. `block` 转换为块级元素
3. `inline` 转换为行内元素
4. `inline-block` 转换为行内块元素
5. `table` 转换为表格

特点：尺寸以内容为主，每个元素独占一行，允许修改尺寸

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` 元素不可见，隐藏

   Q：display: none 和 visibility: hidden 的区别？

   前者不保留位置，不占空间，后者保留位置，占据空间，只是视觉上看不到

3. 透明度

属性：`opacity`

取值：`0.0(完全透明) -- 1.0(完全不透明)`

注意：rgba() 与 opacity 的区别

opacity 作用于元素以及元素内容的透明度

rgba() 只作用于所指定的元素

---

4. 垂直对齐方式

属性：`vertical-align`

场合：

1. 表格

   取值：`top / middle / bottom`

2. 图片中使用

   作用：控制图片两边文本的垂直对齐方式

   取值：

   1. `top`：上
   2. `middle`：中
   3. `bottom`：下
   4. `baseline`：默认值，基线对齐

5. 光标

作用：改变鼠标悬停在元素上时，鼠标的状态

属性：`cursor`

取值：

1. `default` 默认值

2. `pointer` 小手
3. `crosshair` 十字准心
4. `text` 可输入文本状态
5. `wait` 等待状态
6. `help` 帮助

列表

1. 列表项标识

属性：`list-style-type`

取值：none/ disc / circle / square

2. 列表项图片

属性：`list-style-image`

取值：url();

3. 列表项位置

属性：`list-style-position`

取值：outside 默认值，将标识放置于内容区域之外

inside 将标识放置于内容区域之内

4. 列表属性简写

属性：`list-style`

取值：type url() position;

定位

1. 定位属性

属性：`position`

取值：

relative 相对定位

absolute 绝对定位

fixed 固定定位

static 静态，默认值

2. 偏移属性

属性：top / bottom / left / right

2. 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（相对于元素本身）

2. 使用场合

   在做元素位置的微调时使用

3. 语法

   position: relative;

---

3. 绝对定位

1. 语法

   `position:absolute`

   配合偏移属性使用来实现位置的修改

2. 使用场合

   1.有堆叠效果的元素

   2.弹出菜单

3. 特点

   绝对定位元素相对于离他最近的 已定位的 祖先元素去实现位置初始化，

   ```
   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。

   绝对定位的元素会脱离文档流，不占页面空间。
   ```

---

4. 固定定位

1. 什么是固定定位

   将元素固定在页面某个位置处，位置不会随着滚动条变化 而发生位置的变化，固定在可视区域中

2. 语法：

   `position: fixed;`

   配合偏移属性

3. 注意

   1. 固定定位的元素会变为块级元素，可设置宽高

   2. 固定定位的元素永远都是相对于 body 进行位置初始化

   3. 固定定位的元素会脱离文档流，不占页面空间

堆叠顺序

概念：一旦元素变为 **已定位元素**，元素则可能出现堆叠效果，如何改变堆叠顺序？

属性：`z-index`

取值：数字，数字越大越靠上

注意：

1. 只有已定位元素才能设置 z-index
2. 元素在结构上是父子关系，z-index 无效，永远是子元素在父元素之上

弹性布局

概念

是一种布局方式，主要是解决某元素中子元素的布局方式，可以为布局提供最大的灵活性

相关概念

1. 容器

要布局子元素的父元素，称为 **容器**

2. 项目

要实现布局效果的元素，称为 **项目**

3. 主轴

项目排列方向的一根轴，称为 **主轴**

如果项目按 x 轴排列，那么 x 轴就是 主轴

如果项目按 y 轴排列，那么 y 轴就是 主轴

4. 交叉轴

与 主轴 垂直交叉的一根轴称为 **交叉轴**

语法（相关属性）

---

1. 容器

属性：display

取值：

flex 将块级元素变为容器

inline-flex 将行内元素变为容器

2. 注意

1. 容器的 text-align 属性 失效元素变为容器之后，
2. 子元素的 float, clear, vertical-align 属性 失效

---

3. 容器属性

1. flex-direction

作用：指定容器的主轴及其排列方向

取值：

1. row 默认值，主轴为 x 轴，起点在左端
2. row-reverse 主轴为 x 轴，起点在右端
3. column 主轴为 y 轴，起点在顶端
4. column 主轴为 y 轴，起点为底部

2. flex-wrap

作用：当一个主轴排列不下所有项目时，如何换行

取值：

1. nowrap 默认值，即空间不够时也不换行，自动缩小
2. wrap 换行
3. wrap-reverse 换行反转

3. flex-flow

作用：flex-direction + flex-wrap 的缩写形式

取值：

direction wrap ;

row nowrap 默认值

4. justify-content

作用：定义项目在主轴上的对齐方式

取值：

1. flex-start 默认值，主轴起点对齐
2. flex-end 主轴终点对齐
3. center 主轴居中对齐

4. space-between 两端对齐

5. space-around 每个项目两端间距相同

5. align-items

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 交叉轴上基线对齐

5. stretch 如果项目未设置宽高，在交叉轴上将占满父元素所有空间

---

4. 项目属性

该组属性只控制某一个项目，是不会影响别的项目和容器

1. order

作用：定义项目的排列顺序，值越小，越靠近起点

取值：整数数字，无单位，默认值 0

2. flex-grow

作用：定义项目的放大比例，如果容器有足够的剩余空间，项目将按指定比例进行放大

取值：整数数字，无单位，默认值 0，取值越大，占据空间越大

3. flex-shrink

作用：定义项目的缩小比例，当容器空间不足时，项目按指定比例进行缩小

取值：整数数字，无单位，

默认值 1，空间不足时，等比缩小

值为 0，不进行缩放

值越大缩小比例越大

4. align-self

作用：定义当前某一个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align-items)

转换

---

1. 概念

使用 CSS 属性值在一段时间内平缓变化的效果

2. 语法

1. 指定过渡属性

属性：`transition-property`

作用：指定哪个属性值在变化的时候需要使用过渡效果来体现

取值：

1. 具体属性名

   EX: transition-property: background;

   允许设置过渡效果的属性：

   1. 颜色相关属性（背景颜色，字体颜色，边框颜色 ...）
   2. 取值为数字的属性
   3. 转换属性
   4. 渐变属性
   5. visibility属性
   6. 阴影属性

2. all 能使用过渡的属性，一律使用过渡体现

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

3. 指定过渡速度

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，快速结束
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，再减速

4. 指定过渡的延迟时间

属性：`transition-delay`

取值：以 s 或 ms 为单位的数字

3. 注意

过渡属性的编写位置的影响

1. 将过渡属性放在元素声明的样式中

   既管去，又管回

   1. 将过渡放在触发的操作中（:hover）

   只管去，不管回

4. 简写方式

```
transition: property duration [timing-fucntion] [delay];
```

EX:

transition: all 2s;