暗示全局变量

预编译的四个过程：

1.创建AO（activation object）对象，全局中生成GO（global　object）对象

2.找形参和变量声明，将变量和形参名作为ＡＯ属性名，值为undefined

3.将实参值和形参统一

4.在函数体里面找函数声明，值赋予函数体

变量声明和函数声明优先级？？？

选择器的权重：256进制

！important infinite

行间样式 1000

id 100

class ||属性||伪类 10

标签||伪元素 1

通配符 0

防止外部div元素随着内部div元素的margin-top值改变而改变：在父级元素中加上overflow: hidden;(推荐)或者在子元素中加padding-top：1px;（不推荐）

选择器的种类：

Id选择器，class选择器，标签选择器，通配符，属性选择器，父子选择器，并列选择器，分组选择器

伪类选择器(hover):当鼠标进入标签才会触发css属性

css3选择器： div: nth-child(index+1), 选中是第index个子元素的div

div: nth-of-type(index+1)，选中第index个div元素，不分位置

Arguments只用与函数里，用于管理带入函数的参数，对象含有：

caller属性：如果函数在全局执行环境被调用，则值为null，若在其他函数被调用，则值为调用它的函数

length属性：表示输入参数的个数，从0到n-1。

对象里面元素的属性：

Value:属性的值

Writable(bool):是否为只读

Enumerable(bool):是否可用for in 循环枚举属性

Configurable(bool):无法用delete删除该属性

Get:设置一个getter访问器

Set:设置一个setter访问器

var ken=Object.defineProperties({},{

sex:{

value=”male”,

set function(){},

writable:false

},

Weight:{}

});

Border-style:

|  |  |
| --- | --- |
| **值** | **描述** |
| none | 定义无边框。 |
| hidden | 与 "none" 相同。不过应用于表时除外，对于表，hidden 用于解决边框冲突。 |
| dotted | 定义点状边框。在大多数浏览器中呈现为实线。 |
| dashed | 定义虚线。在大多数浏览器中呈现为实线。 |
| solid | 定义实线。 |
| double | 定义双线。双线的宽度等于 border-width 的值。 |
| groove | 定义 3D 凹槽边框。其效果取决于 border-color 的值。 |
| ridge | 定义 3D 垄状边框。其效果取决于 border-color 的值。 |
| inset | 定义 3D inset 边框。其效果取决于 border-color 的值。 |
| outset | 定义 3D outset 边框。其效果取决于 border-color 的值。 |
| inherit | 规定应该从父元素继承边框样式。 |

事件（信号）：

Click(点击),mouse move(鼠标移动),focus(获得焦点)

一些audio元素特有的事件：

Play(开始播放),pause(停止播放),volumn change(音量变化),ended(播放结束)

写在<audio>中为onplay,onpause,onended.

Js:console.log();console就是浏览器的控制台，console.log()就是在控制台输出log里面的内容。相较于document.write()方法，输出的内容更加的详细。

**css3:**

Css3兼容行写法：

-webkit-：兼容chrome和safari

-moz-：兼容Firefox

-ms-：兼容IE

-o-：兼容opera，但是如今的opera内核已经改变成webkit,所以这一行一般不需要写了

但是在开发的时候兼容性的代码，打包工具会帮助自动补全

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [@keyframes](http://www.w3school.com.cn/cssref/pr_keyframes.asp) | 规定动画。 | 3 |
| [animation](http://www.w3school.com.cn/cssref/pr_animation.asp) | 所有动画属性的简写属性，除了 animation-play-state 属性。 | 3 |
| [animation-name](http://www.w3school.com.cn/cssref/pr_animation-name.asp) | 规定 @keyframes 动画的名称。 | 3 |
| [animation-duration](http://www.w3school.com.cn/cssref/pr_animation-duration.asp) | 规定动画完成一个周期所花费的秒或毫秒。默认是 0。 | 3 |
| [animation-timing-function](http://www.w3school.com.cn/cssref/pr_animation-timing-function.asp) | 规定动画的速度曲线。默认是 "ease"。 | 3 |
| [animation-delay](http://www.w3school.com.cn/cssref/pr_animation-delay.asp) | 规定动画何时开始。默认是 0。s|ms | 3 |
| [animation-iteration-count](http://www.w3school.com.cn/cssref/pr_animation-iteration-count.asp) | 规定动画被播放的次数。默认是 1。n|infinite | 3 |
| [animation-direction](http://www.w3school.com.cn/cssref/pr_animation-direction.asp) | 规定动画是否在下一周期逆向地播放。normal|alternate | 3 |
| [animation-play-state](http://www.w3school.com.cn/cssref/pr_animation-play-state.asp) | 规定动画是否正在运行或暂停。running|paused | 3 |
| [animation-fill-mode](http://www.w3school.com.cn/cssref/pr_animation-fill-mode.asp) | 规定对象动画时间之外的状态。 | 3 |

**css动画：**

transform：

旋转：rotateX(),rotateY(),rotateZ(),rotate3d(x, y, z, deg)，rotate3D与x,y,z分别旋转deg不相同，是沿着类似合力的方向旋转deg

缩放：scale(x,y)水平，竖直缩放，x,y可以是负值，scaleX(), scaleY(), scaleZ(),scale3D

倾斜：skewX()x方向上倾斜，高度不变，同skewY(),skew(x,y)若只填一个，第二个默认为0

平移：translateX(),translateY(),translateZ()平移相当于position：absolute，但是不脱离文档流，不会影响其他元素，可以实现垂直水品居中，translate(-50%, -50%)来代替margin-left和margin-top,在不知道元素宽高的情况下使用

Transform-origin: 改变旋转的中心参数可以填left, right, top, bottom, 百分数（从左上到右下）

transform-style: flat | preserve-3d，该属性要设置在父级当中，表示所有子元素都立体展示

preserve-origin：默认是50% 50%

perspective景深，可以设置在父级中perspective: px，也可以设置在每个子元素中, perspective(px)

backface-visibility: hidden，图形背面隐藏，默认为可见

transition: 过渡效果

transition-property: 指定过渡属性 width, height, …若想写多个属性，可以用逗号隔开

transition-duration：指定过渡时间 s, ms

transition-timing-function：指定过渡函数，linear匀速

transition-delay：指定开始出现的延迟时间 s, ms

animation: name time timing-function 动画

animation-name: 运动名（keyframes值）

animation-duration: 运动执行时间

animation-timing-function: 运动函数

animation-delay: 执行延迟

animation-direction: 方向，默认方向，normal默认值，动画按正常播放，reverse，动画反向播放，alternate奇数次正向，偶数次方向，相反的是alternate-reverse

animation-iteration-count: 运功次数，默认为一次，infinite为无限次

animation-play-state: running|paused

animation-fill-mode: 运动后停止的状态

@keyframs name{ (定义动画)  
 0%{}

100%{}

}

[***animation-timing-function***](http://www.w3school.com.cn/cssref/pr_animation-timing-function.asp)

|  |  |  |
| --- | --- | --- |
| **值** | **描述** | **测试** |
| linear | 动画从头到尾的速度是相同的。 | [测试](http://www.w3school.com.cn/tiy/c.asp?f=css_animation-timing-function) |
| ease | 默认。动画以低速开始，然后加快，在结束前变慢。 | [测试](http://www.w3school.com.cn/tiy/c.asp?f=css_animation-timing-function&p=2) |
| ease-in | 动画以低速开始。 | [测试](http://www.w3school.com.cn/tiy/c.asp?f=css_animation-timing-function&p=3) |
| ease-out | 动画以低速结束。 | [测试](http://www.w3school.com.cn/tiy/c.asp?f=css_animation-timing-function&p=4) |
| ease-in-out | 动画以低速开始和结束。 | [测试](http://www.w3school.com.cn/tiy/c.asp?f=css_animation-timing-function&p=5) |
| cubic-bezier(*n*,*n*,*n*,*n*) | 在 cubic-bezier 函数中自己的值。可能的值是从 0 到 1 的数值。 |  |

**animate-fill-mode:**

|  |  |
| --- | --- |
| **值** | **描述** |
| none | 不改变默认行为。 |
| forwards | 当动画完成后，保持最后一个属性值（在最后一个关键帧中定义）。 |
| backwards | 在 animation-delay 所指定的一段时间内，在动画显示之前，应用开始属性值（在第一个关键帧中定义）。 |
| both | 向前和向后填充模式都被应用。 |

尾元素清除浮动：

Node:after {

content: “”;

clear: both;

display: block;

}

box-shadow: x轴偏移量 y轴偏移量 阴影模糊距离 阴影扩展半径 阴影颜色 投影方式(inset 默认：outset)

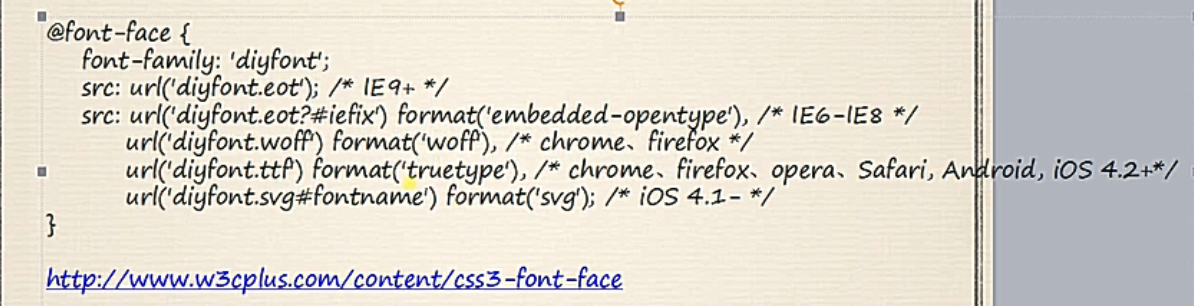
text-shadow: x轴偏移量 y轴偏移量 阴影模糊距离 阴影颜色

word-break: 强制换行，默认值是normal，当遇到空格是换行，break-word,当字体超出容器宽度强制换行。

自定义字体：

@font-face: font-family: ‘字体名’

src : 该字体文档的地址



设置背景图片：background-image:url() number repeat|round|stretch

number表示从图片四周截取该长度，铺在背景区域（包括border，padding，content，但是content区域的图片会被该区域挡住。但是该区域是有图片的）

strentch: 如果截取后还有剩余的区域，系统自动将剩余部分填充满，但是如果没有剩余的区域，那么剩余区域将不会被听填充

repeat：如果切除有剩余，填充不够的话保持剩下区域重复填充，即使不是正好填充满，多余部分切除

round：如果切割有剩余，填充不够的情况下，会把切割剩余部分适当调整与正好铺满剩下区域

背景图片填充起始位置：border-origin: border-box|padding-box|content-box

默认情况下值为padding-box，即从padding开始

背景图片裁剪：background-clip: border-box|padding-box|content-box|no-clip

默认值为border-box，表示全部显示，和no-clip是相同的效果，padding-box表示只留下padding以内的图片，content-box表示只留下content以内的图片

**css选择器：**

属性选择器：

ele[attr^=’val’]{ }类名以val开头的元素

ele[attr$=’val’]{ }类名以val结尾的元素

ele[attr\*=’val’]{ }类名中含有val的元素

伪类选择器：

: root{ }可以直接使用，选择根元素也就是html或xml

: not[]{ }除了选中的元素外，全部选中

: empty[]{ }元素内容为空的标签

: target目标元素选择器，被当作锚点目标的元素

: first-child{ }若前面不加元素，表示所有元素的第一个子元素，若有元素表示为第一个子元素的该元素(ele:first-child)，同:last-child

: nth-child(n){ }第n个元素，也可填写表达式，例如2n+1但是此时的n是从0开始的

: first-of-type(){}和first-child相似但first-of-type只在一中元素中查找，同last-of-type,同ntd-of-first

: only-child子元素下的唯一一个子元素将被选中，:only-of-type子元素中该类子元素只有一个将被选中

: enabled表示input中没有禁用的 : disabled表示被禁用的，相似的还有:checked，表示被点击的表单元素，例如：:checked+span表示选中checked的紧挨的下一个元素

伪元素:

: after{ } : before{ }元素的after和before即使设置了样式也不会被当作元素内部Dom元素，也就是说这个两个伪类不是DOM元素

:fisrt-letter选中第一个字 :first-line 选中第一行

:selection选中的鼠标选中的字体

条件选择器:

a > b选中a下面的直接子元素

a + b 选中a紧挨着的b元素

a ~ b 选中a下的所有b元素

**columns：报纸布局**

column-count：number，表示字符呈现几列，每列之间会有空格，空格大小继承自字体大小，可以用column-gap手动控制

column-width：n px，表示每列的宽度，column-width和column-count最好不要一起使用， width = column-gap\*（column-count-1）+ column-width\* column-count

column-gap：控制空格大小

column-rule：参数值和border一样（column-rule-width column-rule-style column-rule-color），rule不占用DOM结构，所以不会影响布局

column-span：1/all 占一格或者占整个一行

**弹性盒子：display：flex**

主轴（x）侧轴（y），但是可以通过手动改动，默认DOM结构会按照主轴排列

**子元素设置的属性：**

flex:grow,shrink,basic的复合写法，默认值为0 1 auto

flex­-grow: 比例值，排版结束后剩下区域按比例分配

flex-shrink: 比例值，多出部分含比例缩小

flex-basic: 在主轴方向上的值是多少，可以用来设置宽高等属性，但是比width，height等属性的优先级高

order: 元素会按照order的大小顺序排列

align-self: 允许单个项目有与其他项目不一样的对齐方式，可覆盖align-items属性，默认值是auto，表示继承父元素的align-items属性，如果没有父级，就等同于stretch，参数有：baseline，center，flex-start，flex-end，stretch

**父元素设置的属性：**

flex-direction：自定义主轴方向

默认值为row，主轴为水平方向，起点为左端

row-revrese，主轴为水平从右向左，起点为右端

column，主轴为垂直方向，起点为上沿

column-reverse，主轴为垂直方向，起点为下沿

flex-wrap：设置超出换行方式

默认值为no-wrap，超出不换行

wrap：超出换行，wrap-reverse

flex-flow：是flex-direction和flex-wrap的简写

justify-content: 定义在主轴上的对其方式

参数有：flex-start左对齐，flex-end右对齐，center居中，space-between两端对齐中间均匀分开，space-around每个项目两侧间隔相等，所以比项目与边框间距大一倍

Align-iteams: 属性定义项目在侧轴上如何对齐

flex-start，flex-end，center，baseline，stretch

align-content：定义多根轴线的对齐方式

flex-start，flex-end，center，space-between，space-around

**响应式布局：**

meta中的content值就是为了防止设备的不同，而进行相应的处理

媒体查询：向不同的设备提供不同样式的一种方式，以提高用户的使用体验

**css2中只有media type对不同的设备指定特定的样式**

**css3中对media进行了增强，media query可以看成是media type+css属性判断**

响应式设计有两种写法：

1. link标签中通过meta属性来判定是否引入该css标签，例如meta=’screen and (max-width: 400px)’
2. 另一种方法可以在css文件中写@media screen and (max-width: 400px)，既满足条件是才会使用该属性

and可以合并多个属性

@media中的逻辑操作符，not, only在screen前面使用，and用于连接属性

device-width：设备宽度，电脑宽度，width：渲染区域，即document区域

**物理像素和设备独立像素：**

一个物理像素是显示器上最小的物理显示单位，在操作系统的调度下，每个物理像素都有自己的颜色值和亮度值

设备独立像素：又叫目睹无关像素，表示一个可以有程序使用的虚拟像素（css像素），然后由相关系统转换为物理像素

设备像素比dpr，定义了物理像素值和设别独立像素对应关系

渲染原理：

生成DOM树-> 生成CSS树-> 合成渲染树-> 对渲染树进行布局计算也称回流-> 绘制

重排：也叫回流，就是页面渲染的第四步

重绘：当一些元素需要更新属性，而这些属性会影响元素的外观风格，而不影响布局该过程叫做重绘

Canvas：

防止失真的方法：canvas设置宽高的两种方法，一个是在canvas标签中设置width和height，另一种是设置canvas的css样式，两者并不会相互覆盖，而是css样式会对canvas的宽高进行等比变化为css样式的大小

定义画布的步骤，先定义一块有id的canvas，

var m=getElementById(“canvas的id”);

var m1=m.getContext(‘2d’);

m1.fillStyle=‘rgb()’; 设置当前画布的style

m1.fillRect(x,y,w,h);

制作渐变式效果：

var mycanvas=document.getElementById("canvas1");

var mycanvas2=mycanvas.getContext('2d');

var mygradient=mycanvas2.createLinearGradient(0,0,400,400); //定义变化方向以及位置

mygradient.addColorStop(0,'red');

mygradient.addColorStop(0.3,'green');

mygradient.addColorStop(0.6,'blue');

mygradient.addColorStop(1,'black');

mycanvas2.fillStyle=mygradient; //确定填充颜色，前四行就是定义渐变颜色

mycanvas2.fillRect(0,0,400,400);

二维画图，用笔画：

var mycanvas=document.getElementById("canvas1");

var t=mycanvas.getContext('2d');

t.beginPath(); //开始画画

t.lineWidth=””;

t.arc(x,y,radius,startangle,endangle,ture or faluse); //画圆

t.closePath(); //将图形封闭

t.strokeStyle=””; //画笔的颜色

t.fillStyle=””; //填充效果，自动补全不封闭图形

阴影：

t.globalAlpha = ‘’ 设置全局的透明度

t.shadowColor 阴影颜色

t.shadowOffsetX 阴影在X轴上的偏移量

t.shadowOffsetY 阴影在Y轴上的偏移量

t.shadowBlur 阴影模糊距离

文本：

t.fillText(str, x, y)

t.font = ‘font-size font-family’

t.textAlign = ‘’设置文字对齐方式，默认为左下方，水平方向有start，left，center，right，end，竖直方向：top，hanging，middle，alphabetic，ideographtic，bottom

t.textBaseline = ‘’设置文字与水平中线的对齐方式 top, bottom, middle, alphabetic, hanging

线段样式：

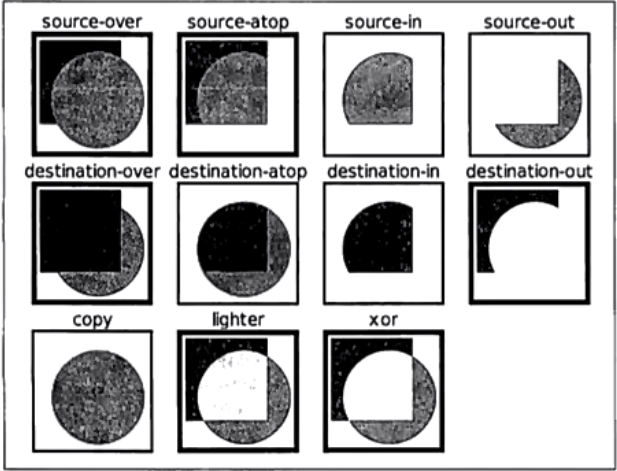
lineCap:默认为butt，表示线段两头不加东西，square表示两头加上方块，round表示两边加上半圆

lineJoin：相交处的处理，默认miter为尖角，bevel为折角，round为圆角，miter属性有一定的范围，超过范围就会变为bevel，同样可以通过miterLimit设置范围

裁剪：clip()，当前路径以外的区域不再绘制

裁剪后可以通过save() 和restore()来恢复裁剪之间的画布

合成：t.globalCompositeOperation =’’



绘制图片：

三个参数时：t.drawImage(img, x, y);

五个参数时：t.drawImage(img, x, y, wid, hei)；

九个参数时：t.drawImage(img, x1, y1, wid, hei, x2, y2, wid2, hei2)，表示从img上x1, y1,除截取wid,hei的区域放置在canvas

x2, y2,宽为wid,高为hei

将canvas内容导出：mycanvas.toDataURL()，

1.是canvas自身的方法不是上下文对象，是canvas上的方法

2.在满足同源策略的条件下，把canvas内容抽成一张图片，base64编码格式

3.将canvas的内容放入img元素里

获取像素值信息：

t.getImageData(x, y, wid, hei) 信息表示每个像素的rgba值，所以信息个数为wid\*hei\*4

t.createImageData(wid, hei)创建新的空白image对象

t.putImageData(imgData, x, y)将图像信息放回到画布上

命中检测：

t.isPointInPath(x, y) 检测是否在区域内

t.isPointInStroke(x, y) 检测是否在线上

还可以通过检测当前点的像素值，如果为透明，则该点不在路径上

SVG：和canvas一样用来画图形的，但是不同的是svg是矢量图，放大后不会失真，且svg是由标签组成的，当内容特别复杂需要用到很多标签是建议使用canvas来代替svg

<svg width= height= viewBox=”” preserveAspectRatio=></svg>

<line x1 y1 x2 y2></line> line只能定义直线需要添加stroke=color, 来画出来

<rect x y width height rx ry></rect> rx, ry分别表示x,y轴上的圆角,且rect默认填充为黑色

<circle r cx cy></circle> cx, cy表示中心点

<ellipse rx ry cx cy></ellipse> rx, ry 表示x,y轴上的半径

<polyline points=”x1 y1, x2 y2, x3 y3,..”></polyline>折线，每两个值表示一个点，中间的逗号可以不写但是为了区分开，建议写上

<polygon point = “x1 y1, x2 y2, x3 y3,..”></polygon>多边形，最后会收尾闭合

<text x y></text>表示写入的位置

<g></g>用来包裹元素，用来设置包裹元素里面的颜色

<path d=’M x y L x2 y2 H h1 V v1’></path>可以定义一条路径，

M指令相当于canvas中的moveTo，当指令为大写其后的点表示绝对坐标，小写表示相对上一个点的相对坐标，

L指令表示移动到某一位置，相当于moveTo

H表示水平移动，

V表示垂直移动，

Z指令表示首位相连，

A指令有7个参数，分别表示x,y两轴半径，旋转角度，1大圆弧还是0小圆弧，1顺时针还是0逆时针，结束位置的标，

Q指令是表示二次贝塞尔曲线，参数是4个，表示两个点，另外T指令是Q的延伸，参数为一个点，延伸两点与与T构成另外一条线

C指令表示三次贝塞尔曲线，指令S为C的延伸，

<defs>

<linearGradient id x1 y1 x2 y2>线性渐变，id值用于其他元素的引用，x,y值是百分数，表示起始位置,需要<defs>包裹

<stop offset=’’ style=’stop-color: ’ > stop定义颜色渐变位置及颜色

<stop offset=’’ style=’stop-color: ’ >

</linearGradient>

</defs>

在填充是需要用到时，使用style=” fill:url(#id)”来引用，同样也有径向渐变，<radialGradient id cx cy r fx fy>

高斯滤镜：可以实现模糊效果,filter标签也需要用defs标签来包裹，引用方式也和渐变一样

<filter id >

<feGaussianBlur in=’SourceGraphic’ stdDeviation=”20”/>

</filter>

属性：在css中写

stroke：描边效果

fill：填充效果

stroke-width：描边宽度

fill-opacity/stroke-opacity: 填充/描边透明度

stroke-linecap：同canvas的lineCap

stroke-linejoin: 同canvas的lineJoin

stroke-dasharray: 可以填写无数参数，直线会根据参数值成虚线分布

stroke-dashoffset：正值向左缩进，负值向右缩进

view Box=”0, 0, 40 , 40”设计svg的实际大小

preserveAspectRatio: 设置图形对齐方式，默认时居中对齐，

xmin xmid xMax 轴的左中右对齐

Ymin Ymid YMax 轴的左中右对齐

meet 将图片撑开 clice撑开直到全部撑满 none形变来时整个区域撑开

方法：

getTotalLength() 获取路径总长度

getPointAtLength(x) 获取路径上距离起始点长度为x的点的坐标

!一般来说，这两个方法只能用于path元素，但是不同浏览器实现起来会有一点差别，谷歌也能获取到line元素的路径长度

Less:less是一种css预处理语言

less使用规则：

1. @变量名：定义变量，在该文件内都可以使用，使用时也是用@变量名
2. Less中可以进行运算
3. 可以进行嵌套式写法，例如：

.wrapper{

Width: 400px;

Height: 400px;

.box{

Background: skyblue;

}

}

1. &用于一些伪元素，表示当前父元素
2. Less中设置的属性会遵循冒泡规则，@media，@keyframes，@font-size等同样遵循这个规则
3. 在运算时，每个运算单元都要单位，如果结果是无效的less会忽略这次设置，乘除是会忽略后面操作数的单位，以前面的操作数单位为准
4. 颜色值再相加时，会将颜色值4个值分别相加，red green blue alpha，如果颜色值大于ff或者小于00，就会默认为ff或0，alpha同样

**栅格布局：display: grid**

在父级中设置display:grid，子元素呈现栅格布局，基本规则(父级设置网格模型，子级设置在网格中的位置)

parent {

display: grid;

grid-template-column: … x轴上每个元素宽度

grid-template-rows: … y轴上每个元素的高度

}

child {

//grid-column-start, grid-column-end, grid-rows-start, grid-rows-end

//gird-column: start/end , gird-rows: start/end

//gird-area: => grid-column-start / grid-rows-start / grid-column-end / grid-rows-end

//以上是栅格定位的三种方式，栅格布局column，rows上的线都是从1开始的

}

Background：

Background-image: css3设置多张背景图片

Background-image: url(), url()

Background-size: size1, size2

Background-position: pos1, pos2

Background-repeat: no-repeat

//默认情况下，两张背景图片是重叠在一起的，需要使用background-position来定位

Background-origin: berder-box, padding-box(默认值), content-box //背景图片起始位置

Background-clip: border-box(默认), padding-box, content-box //背景图片结束位置

Background-clip: text: 用文字剪切背景图片

-webkit-background-clip: text;

Background-clip: text

-webkit-text-fill-color: transparent;

text-fill-color: transparent;

background-image:

linear-gredient(to dir, color1 pos1, color2 pos2) //pos为终止位置

！！！渐变色只能放置在background-image中才能有效，而不是background-color

Radial-gredient(pos, color1 pos1, color2 pos2, …)

radial-gradient(closest-side circle at 100px 100px, red, blue)

//pos: circle at pos1 pos2 | ellipse at pos1 pos2 | closest-side(到最近边为止) | closest-corner(到最近的角为止) | farthest-side(到最远的边为止) | farthest-corner(到最远的角为止)

background-repeat: no-repeat | repeat-x | repeat-y | around | space

space: 图片尽可能平铺，多余的空格会均匀分布在图片之间

around：图片尽可能平铺，多余的空格让图片平均伸展来填充

background-attachment:

scroll: 默认值,background-position相对于容器进行定位, 即不随滚动条移动

local: 相对于容器整个区域进行定位，随容器滚动条移动

fixed:相对于页面进行定位，即容器移动时，背景图片也不会移动

background-size:

cover: 背景图片长宽等比缩放，直到盖住整个容器

contain: 背景图片长宽等比缩放，直到图片长或宽等于容器，多余部分会repeat

@font-face { //引入字体包

Font-family: 自定义字体名

Src: 字体地址

}

Truetype 微软 苹果 .ttf

Opentype 微软 abode .opt

Woff .woff

.eat

.svg

text-align: justify //不处理最后一行文本，包括只有一行文本（第一行即最后一行）

overflow: 默认值是visible, overflow-x和overflow-y可以分开设置，但是凡两者中一个设置为非visible外的值，另一个都会自动设置为auto

resize: 调整元素的宽高，值有none|both|horizontal|vertical，使用时必须设置overflow值才能使用

align-items align-self属性值相同 center, flex-start, flex-end, baseline, stretch

justify-content align-content属性值相同 center, space-between, space-around

align-self用于设置元素自生的flex相关属性，优先级小于align-content，大于align-items

order 用于子元素排列，默认值为0，值越大处于页面层面越高；

flex: flex-grow flex-shrink flex-basis/width

flex-basis: 子元素基础width，一般情况下会覆盖width属性

flex-grow:

flex-shrink: content \* flex-shrink / ( content \* flex-shrink + … ) \* 多出部分

flex-basis ⬄ width

当元素中文字（英文）超出元素宽度，flex-basis会撑开直到包括所有内容，直到撑开到width的宽度。中文会自动换行

即flex-basis提供最小宽度，width提供最大宽度，若flex-basis > width，则元素宽度就一直为flex-basis，不变化，

当flex-basis撑开到width时不参加flex-shrink计算，若想参与换行可填入中文，或者设置word-break: break-word使文字换行

word-break ⬄ word-wrap ⬄ white-space

white-space：主要用来处理元素内文本的空白符、换行符

normal(默认)：连续空白合并，换行符当作空白符来处理

pre：空白符保留，不自动换行，原样输出

nowrap：和normal一样合并空白符，但不换行

pre-wrap：保留空白符序列，但是正常进行换行

pre-line：合并空白符序列，但是保留换行符

word-break:

normal(默认值),

break-word文档并未显示该属性(尽量保证单词完整的情况下换行，如果单词长度大于容器宽度，则单词会被截断)， 非中日韩文的语言与normal相同,

break-all(无视单词的完整性，到容器边界就截断单词换行),

keep-all

word-wrap:

normal(默认值), break-word(和word-break: break-word效果一样)

animation-fill-mode:

none: 不改变默认行为

forwards: 当动画完成后，保持动画最后一个属性值

backwards: 在animation-delay所指定的一段时间内，显示为动画开始的属性

both: 兼顾forwards和backwards

animation-timing-function: steps(step-count, start|end)

steps(1, start) === step-start steps(1, end) === step-end

将animation动画每一帧用跳转形式表示，step-count表示每一帧变为多少步展示出来，总时长不变

start一段时间里显示下一帧的开始状态，所有start看不到第一帧状态，很难弥补

end一段事件显示当前帧开始状态，看不到最后一帧可以甚至forwards来保留最后一帧

transform-origin: 设置元素变化中心

transform-style: preserve-3d； 将容器设置为3d渲染

perspective: \_\_px 设置容器的景深

perspective-origin: 设置人视角位置

transform：scale() 变化的不是元素本身，而是坐标轴的刻度，元素长宽并不发生变化

scale缩放后，旋转时缩放轴会随之旋转，但是原方向上的缩放效果仍然存在？？

will-change: 属性 //为该属性启动GPU渲染，浏览器会准备一个新的层来渲染，以节约性能

1. 处理 HTML 标记并构建 DOM 树。

2. 处理 CSS 标记并构建 CSSOM 树。

3. 将 DOM 与 CSSOM 合并成一个渲染树。

4. 根据渲染树来布局，以计算每个节点的几何信息。

5. 将各个节点绘制到屏幕上。

width == device-width : iphone 或者 ipad上横屏宽度 = 竖屏宽度

initial-scale=1.0 : windows系统的 phone上的 ie浏览器上横屏宽度 = 竖屏宽度

两者一起写为了兼容两者

meta中content其他值：

user-scalable = no 不允许用户缩放（双击或其他操作放大或缩小页面）

响应式网页开发：

1. 流体网格：可伸缩的网格，即尽量不要将元素大小定死，用百分比来表示
2. 弹性图片：图片高度不固定（min-width, max-width）
3. 媒体查询：让网页在不同终端上展示相同的效果
4. 主要断点：设备宽度的临界点

媒体查询作用方式：

1. 在link标签引入css文件时

<link rel=’stylesheet’ media=” screen and (max-width:375px)” href=’.css’/>

1. 在style标签结尾引入css文件时

@import ‘.css’ screen and (max-width:175px)

1. 在css文件中直接使用媒体查询

@media screen and (max-width: 300px) and (min-width: 100px) {

…

}

媒体查询条件用and表示逻辑与，用逗号表示逻辑或