目录：

[弹性盒布局](#4812-1554895835120)

[4个概念](#2141-1554894904214)

[1种布局](#8320-1554894904214)

[meta标签及属性](#1574-1554894904214)

[移动端1px解决方案](#2131-1554894904214)

[动画](#3325-1554894904214)

[2D转换](#5628-1554894904215)

[3D转换](#4163-1554894904215)

[过渡](#2786-1554894904215)

[事件](#3565-1554894904215)

[localStorage](#5152-1554894904215)

[localStorage基本语法](#7926-1554894904215)

[Drag&Drop拖放](#4026-1554894904216)

[DAY9-移动开发-day9-多媒体(视频和音频)](#7937-1554894904216)

[canvas](#2347-1554894904216)

[颜色、样式和阴影](#8310-1554894904219)

[路径](#9261-1554894904219)

[转换](#7900-1554894904219)

[文本](#3750-1554894904219)

[better-scroll 是什么](#1784-1554894904219)

**弹性盒布局**

注意：设为Flex布局以后，子元素的float、clear和vertical-align属性将失效。

7个属性设置在父元素上

·flex-direction   位置

·flex-wrap   换行

·flex-flow   （flex-direction+flex-wrap）

·justify-content   主轴

·align-items    反主轴

·align-content

·display:flex

6个属性设置在子元素上

·order         排序

·flex-grow      自动撑开剩余空间

·flex-shrink     收缩

·flex-basis      默认的

·flex

·align-self

align-items  //设置垂直（侧轴）方向的对齐方式

·flex-start      交叉轴起点对齐。

·flex-end         交叉轴终点对齐。

·center          交叉轴中点对齐。

·baseline         项目的第一行文字的基线对齐。

·stretch (默认值)：如果项目未设置高度或设为auto，将占满整个容器的高度

align-content  //设置多轴对齐的方式(如果只有一根轴线，一行，这个属性不起作用)

·flex-start

·flex-end

·center

·space-between

**4个概念**

设备检测

·判断 是苹果或安卓手机

·window.navigator.appVersion

移动app的开发方式

·nativeapp(原生app)

·webApp

·hybridapp（混合app）

手机像素

·逻辑像素 css

·物理像素 设备

·设备像素比（window.devicePixelRatio）devicePixelRatio=window.devicePixelRatio

在维度上dpr计算公式：1px=2\*dp

dpr=逻辑像素（640）/物理像素（320）

320px=640dp

手机视口-viewport

·理想视口

·布局视口

·视觉视口

**1种布局**

·自适应

·圣杯布局

·双飞翼布局

·绝对定位布局

·flex弹性盒布局（父/子元素、属性、使用场景）

**meta标签及属性**

**meta篇**

<metaname="viewport"content="width=device-width,initial-scale=1,minimum-scale=1,maximum-scale=1,user-scalable=no">

①viewport获取视口宽度

②device-width设备宽度

③initial-scale初始化缩放比例

④minimum-scale最小缩放比例

⑤maximum-scale最大缩放比例

⑥user-scalable=no禁止用户缩放

默认值是yes开启状态

**移动端1px解决方案**

ios设备

·transform:scale(0.5)

·meta:inital-scale=0.5

android设备

·border-image

·渐变

·阴影

·伪元素

**iconfont字体图标**

·阿里巴巴图标库

·font awesome字体图标库

·@font-face  定义一种字体

· font-family: "iconfont";  //起名为 iconfont

四种格式：.ttf/.woff/.eot/.svg

**rem**

·rem是相对于html/根元素的字体大小

·计算rem

·media

·calc

·vh

·vw

·js

·sass

**动画**



定义动画 @keyframes 动画名{

        form{}

to{}

}

animation-duration  动画花费时间

animation-timing-function:linear  匀速

      ease   慢-->快-->慢

animation-delay   延迟

animation-direction：alternate  逆向

animation-play-state：paused  暂停

animation-fill-mode：forwards  停在最后一帧

**2D转换**

translate() 平移

rotate() 旋转

scale() 放大

skew() 斜切

matrix() 把所有 2D 转换方法组合在一起。

matrix() 方法需要六个参数，包含数学函数，允许您：旋转、缩放、 移动以及倾斜元素。

transform-origin 被转换元素的位置

|  |  |
| --- | --- |
| **函数** | **描述** |
| matrix(*n*,*n*,*n*,*n*,*n*,*n*) | 定义 2D 转换，使用六个值的矩阵。 |
| translate(*x*,*y*) | 定义 2D 转换，沿着 X 和 Y 轴移动元素。 |
| translateX(*n*) | 定义 2D 转换，沿着 X 轴移动元素。 |
| translateY(*n*) | 定义 2D 转换，沿着 Y 轴移动元素。 |
| scale(*x*,*y*) | 定义 2D 缩放转换，改变元素的宽度和高度。 |
| scaleX(*n*) | 定义 2D 缩放转换，改变元素的宽度。 |
| scaleY(*n*) | 定义 2D 缩放转换，改变元素的高度。 |
| rotate(*angle*) | 定义 2D 旋转，在参数中规定角度。 |
| skew(*x-angle*,*y-angle*) | 定义 2D 倾斜转换，沿着 X 和 Y 轴。 |
| skewX(*angle*) | 定义 2D 倾斜转换，沿着 X 轴。 |
| skewY(*angle*) | 定义 2D 倾斜转换，沿着 Y 轴。 |

**3D转换**

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [transform](http://www.w3school.com.cn/cssref/pr_transform.asp) | 向元素应用 2D 或 3D 转换。 | 3 |
| [transform-origin](http://www.w3school.com.cn/cssref/pr_transform-origin.asp) | 允许你改变被转换元素的位置。 | 3 |
| [transform-style](http://www.w3school.com.cn/cssref/pr_transform-style.asp) | 规定被嵌套元素如何在 3D 空间中显示。 | 3 |
| [perspective](http://www.w3school.com.cn/cssref/pr_perspective.asp) | 规定 3D 元素的透视效果。 | 3 |
| [perspective-origin](http://www.w3school.com.cn/cssref/pr_perspective-origin.asp) | 规定 3D 元素的底部位置。 | 3 |
| [backface-visibility](http://www.w3school.com.cn/cssref/pr_backface-visibility.asp) | 定义元素在不面对屏幕时是否可见。 | 3 |

|  |  |
| --- | --- |
| **函数** | **描述** |
| matrix3d(*n*,*n*,*n*,*n*,*n*,*n*,  *n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*) | 定义 3D 转换，使用 16 个值的 4x4 矩阵。 |
| translate3d(*x*,*y*,*z*) | 定义 3D 转化。 |
| translateX(*x*) | 定义 3D 转化，仅使用用于 X 轴的值。 |
| translateY(*y*) | 定义 3D 转化，仅使用用于 Y 轴的值。 |
| translateZ(*z*) | 定义 3D 转化，仅使用用于 Z 轴的值。 |
| scale3d(*x*,*y*,*z*) | 定义 3D 缩放转换。 |
| scaleX(*x*) | 定义 3D 缩放转换，通过给定一个 X 轴的值。 |
| scaleY(*y*) | 定义 3D 缩放转换，通过给定一个 Y 轴的值。 |
| scaleZ(*z*) | 定义 3D 缩放转换，通过给定一个 Z 轴的值。 |
| rotate3d(*x*,*y*,*z*,*angle*) | 定义 3D 旋转。 |
| rotateX(*angle*) | 定义沿 X 轴的 3D 旋转。 |
| rotateY(*angle*) | 定义沿 Y 轴的 3D 旋转。 |
| rotateZ(*angle*) | 定义沿 Z 轴的 3D 旋转。 |
| perspective(*n*) | 定义 3D 转换元素的透视视图。 |

**过渡**

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [transition](http://www.w3school.com.cn/cssref/pr_transition.asp) | 简写属性，用于在一个属性中设置四个过渡属性。 | 3 |
| [transition-property](http://www.w3school.com.cn/cssref/pr_transition-property.asp) | 规定应用过渡的 CSS 属性的名称。 | 3 |
| [transition-duration](http://www.w3school.com.cn/cssref/pr_transition-duration.asp) | 定义过渡效果花费的时间。默认是 0。 | 3 |
| [transition-timing-function](http://www.w3school.com.cn/cssref/pr_transition-timing-function.asp) | 规定过渡效果的时间曲线。默认是 "ease"。 | 3 |
| [transition-delay](http://www.w3school.com.cn/cssref/pr_transition-delay.asp) | 规定过渡效果何时开始。默认是 0。 | 3 |

**事件**

TouchList {0: Touch, length: 1}

length:1

0:Touch

    clientX:65 // 触摸点在浏览器窗口中的横坐标

    clientY:18 // 触摸点在浏览器窗口中的纵坐标

    force:1 // 触摸点压力大小

    identifier:0 // 触摸点唯一标识（ID）

    pageX:65 // 触摸点在页面中的横坐标

    pageY:18 // 触摸点在页面中的纵坐标

    radiusX:11.5 // 触摸点椭圆的水平半径

    radiusY:11.5 // 触摸点椭圆的垂直半径

    rotationAngle:0 // 旋转角度

    screenX:560 // 触摸点在屏幕中的横坐标

    screenY:175 // 触摸点在屏幕中的纵坐标

target:div#touchLog 触摸目标

\_\_proto\_\_:Touch

\_\_proto\_\_:TouchList

1.Touch.identifier：此 Touch 对象的唯一标识符。 一次触摸动作(我们指的是手指的触摸)在平面上移动的整个过程中，该标识符不变。 可以根据它来判断跟踪的是否是同一次触摸过程，此值为只读属性。

2.Touch.screenX：触点相对于屏幕左边沿的X坐标。只读属性。

3.Touch.screenY：触点相对于屏幕上边沿的Y坐标。只读属性。

4.Touch.clientX：触点相对于可见视区(visual viewport)左边沿的X坐标。不包括任何滚动偏移。只读属性。

5.Touch.clientY：触点相对于可见视区(visual viewport)上边沿的Y坐标。不包括任何滚动偏移。只读属性。

6.Touch.pageX：触点相对于HTML文档左边沿的X坐标。当存在水平滚动的偏移时，这个值包含了水平滚动的偏移。只读属性。

7.Touch.pageY：触点相对于HTML文档上边沿的Y坐标。当存在水平滚动的偏移时，这个值包含了垂直滚动的偏移。只读属性。

8.Touch.radiusX：能够包围用户和触摸平面的接触面的最小椭圆的水平轴(X轴)半径。这个值的单位和 screenX 相同。只读属性。

9.Touch.radiusY：能够包围用户和触摸平面的接触面的最小椭圆的垂直轴(Y轴)半径。这个值的单位和 screenY 相同。只读属性。

10.Touch.rotationAngle：它是这样一个角度值：由radiusX 和 radiusY描述的正方向的椭圆，需要通过顺时针旋转这个角度值，才能最精确地覆盖住用户和触摸平面的接触面。只读属性。

11.Touch.force：手指挤压触摸平面的压力大小，从0.0(没有压力)到1.0(最大压力)的浮点数。只读属性。

12. Touch.target：当这个触点最开始被跟踪时(在 touchstart 事件中)，触点位于的HTML元素。哪怕在触点移动过程中，触点的位置已经离开了这个元素的有效交互区域，或者这个元素已经被从文档中移除。需要注意的是，如果这个元素在触摸过程中被移除，这个事件仍然会指向它，但是不会再冒泡这个事件到 window 或 document 对象。因此，如果有元素在触摸过程中可能被移除，最佳实践是将触摸事件的监听器绑定到这个元素本身，防止元素被移除后，无法再从它的上一级元素上侦测到从该元素冒泡的事件。只读属性。

**localStorage**

在客户端存储数据

·cookie

·localStorage

·sessionStorage

localStorage基本语法

·localStorage本地方法

·localStorage 的巧用

自己扩展：真实项目中使用的第三方插件

**在客户端存储数据**

cookie

·但是 cookie 不适合大量数据的存储，因为它们由每个对服务器的请求来传递，这使得 cookie 速度很慢而且效率也不高。

·HTML5 提供了两种在客户端存储数据的新机制

localStorage

·localStorage 存储的数据没有时间限制。只要浏览器不清楚数据，它储存的数据一直存在

sessionStorage

·sessionStorage 方法针对一个 会话窗口进行数据存储。当用户关闭浏览器窗口后，数据会被删除。

**localStorage基本语法**

HTML5使用 JavaScript 来存储和访问数据。可以通过调用Window.sessionStorage和Window.localStorage属性创建一个Storage对象的实例。

·Storage 对象以简单的键值得形式来储存数据，键值都是以字符串的形式 进行储存，如果一个值是数字，它将被转为字符串。

localStorage本地方法

|  |  |
| --- | --- |
| **header 1** | **header 2** |
| 添加键值对 | localStorage.setItem(key,value); |
| 获取键值对 | localStorage.getItem(key); |
| 删除键值对 | localStorage.removeItem(key);// 删除数据数据时使用单个参数； |
| 清除所有键值对 | localStorage.clear();// 清空数据数据时不需要参数； |
| 获取localStorage的键名 | localStorage.key; |
| 获取localStorage中保存的键值对的数量 | localStorage.length; |

localStorage 的巧用

·对用户访问页面的次数进行计数：

<script type="text/javascript">

if (localStorage.pagecount)

  {

  localStorage.pagecount=Number(localStorage.pagecount) +1;

  }

else

  {

  localStorage.pagecount=1;

  }

</script>

·用户每次刷新页面都会触发localStorage，触发localStorage后都会给 localStorage.pagecount的值增加增加一，以此来达到统计访问量的目的。

·用户在当前 session 中访问页面的次数进行计数时，将上面代码中的localStorage换成sessionStorage即可。

自己扩展：真实项目中使用的第三方插件

对于数据多维的储存，使用Storage插件会更加方便，比较常用有Storage.js、store.js等插件。

store.js 插件：

该插件小巧，简介，兼容性很强，可实现大多数日常的应用

**Drag&Drop拖放**

HTML5 中提供了直接拖放的 API，极大的方便我们实现拖放效果，不需要去写一大堆的 js，只需要通过监听元素的拖放事件就能实现各种拖放功能。 想要拖放某个元素，必须设置该元素的 draggable 属性为 true，当该属性为 false 时，将不允许拖放。而 img 元素和 a 元素都默认设置了 draggable 属性为 true，可直接拖放，如果不想拖放这两个元素，把属性设为 false 即可。

**拖放事件**

·被拖放的元素称为源对象

·经过的元素称为过程对象

·到达的元素称为目标对象

源对象

·dragstart：源对象开始拖放。

·drag：源对象拖放过程中。

·dragend：源对象拖放结束。

过程对象

·dragenter：源对象开始进入过程对象范围内

·dragover：源对象在过程对象范围内移动。

·dragleave：源对象离开过程对象的范围。

目标对象

·drop：源对象被释放时触发。 凡是在页面内都认为是过程对象，所以要阻止document的dragover的默认行为

**dataTransfer 对象**

在所有拖放事件中提供了一个数据传递对象 dataTransfer，用于在源对象和目标对象间传递数据。 dataTransfer 对象不支持IE

setData()

该方法向 dataTransfer 对象中存入数据。接收两个参数，第一个表示要存入数据种类的字符串，现在支持有以下几种：

·text/plain：文本文字。

·text/html：HTML文字。

·text/xml：XML文字。

·text/uri-list：URL列表，每个URL为一行。

第二个参数为要存入的数据。

event.dataTransfer.setData('text/plain','Hello World');

getData()

该方法从 dataTransfer 对象中读取数据。参数为在 setData 中指定的数据种类。

event.dataTransfer.getData('text/plain');

clearData()

该方法清除 dataTransfer 对象中存放的数据。参数可选，为数据种类。若参数为空，则清空所有种类的数据

event.dataTransfer.clearData();

setDragImage()

该方法通过用img元素来设置拖放图标。接收三个参数，第一个为图标元素，第二个为图标元素离鼠标指针的X轴位移量，第三个为图标元素离鼠标指针的Y轴位移量。

effectAllowed 和 dropEffect 属性

这两个属性结合起来设置拖放的视觉效果。

**DAY9-移动开发-day9-多媒体(视频和音频)**

**audio**

该标签往页面插入音频标签

**语法**

<audio src="song.mp3" controls="controls" loop="loop" autoplay="autoplay">亲 您的浏览器不支持html5的audio标签</audio>

拥有两份源文件的音频播放器

<audio controls>

   <source src="horse.ogg" type="audio/ogg">

   <source src="horse.mp3" type="audio/mpeg">

 Your browser does not support the audio element.

 </audio>

**音频的兼容性**

|  |  |  |
| --- | --- | --- |
| **浏览器** | **版本** | **支持格式** |
| Internet Explorer | 9.0+ | MP3, AAC |
| Chrome | 6.0+ | Ogg Vorbis, MP3, WAV（9.0+） |
| Firefox | 3.6+ | Ogg Vorbis, WAV |
| Safari | 5.0+ | MP3, AAC, WAV |
| Opera | 10.0+ | Ogg Vorbis, WAV |

大约有80%的浏览器支持HTML5的audio标签，但是并没有一种统一的音频格式。从支持的格式来看，要让所有的浏览器可以播放audio元素上的音频，最佳的方式是提供MP3和Ogg两种格式，兼容代码如下：

<audio controls>

    <source src="elvis.mp3" type='audio/mpeg; codecs="mp3"'>

    <source src="elvis.oga" type='audio/ogg; codecs="vorbis"'>

    *<!-- 向后兼容代码：如，显示提示信息、提供下载链接使用flash播放器等 -->*

    浏览器不支持<code>audio</code>标签

</audio>

**视频的兼容性**

|  |  |  |
| --- | --- | --- |
| **浏览器** | **版本** | **支持格式** |
| Internet Explorer | 9.0+ | MP4 |
| Chrome | 6.0+ | MP4,WebM,Ogg |
| Firefox | 3.6+ | WebM,Ogg |
| Safari | 5.0+ | MP4 |
| Opera | 10.0+ | WebM,Ogg |

从浏览器支持的视频格式来看，最佳的方式是提供WebM和MP4两种格式的视频。兼容代码如下：

<**video** controls>

    <source src=**video**.webm type=video/webm>

    <source src=**video**.mp4 type=video/mp4>

    <!—向后兼容代码: -->

    <**iframe** width="480" height="360" src="http://www.youtube.com/embed/xzMUyqmaqcw?rel=0" frameborder="0" allowfullscreen></iframe>

</video>

**属性**

|  |  |
| --- | --- |
| **名称** | **说明** |
| src | 歌曲的路径 |
| controls | 设置或返回音频/视频是否显示控件（比如播放/暂停等） |
| loop | 设置或返回音频/视频是否应在结束时重新播放 |
| autoplay | 当歌曲加载后自动播放，但是只有pc端可以实现,移动端不行 |
| currentTime | 指示音频/视频播放的当前位置（以秒计） |
| playbackRate | 设置或返回音频/视频播放的速度 |
| duration | 返回当前音频/视频的长度（以秒计） |
| ended | 返回音频/视频的播放是否已结束 |
| muted | 设置或返回音频/视频是否静音 |
| paused | 设置或返回音频/视频是否暂停 |
| volume | 设置或返回音频/视频的音量 |

**方法**

|  |  |
| --- | --- |
| **名称** | **说明** |
| load() | 重新加载音频/视频元素 |
| play() | 开始播放音频/视频 |
| pause() | 暂停当前播放的音频/视频 |

**事件**

|  |  |
| --- | --- |
| **名称** | **说明** |
| canplaythrough | 当浏览器可在不因缓冲而停顿的情况下进行播放时 |
| ended | 当目前的播放列表已结束时 |
| timeupdate | 当目前的播放位置已更改时 |

**canvas**

canvas标签用来绘制图像，本身就是一个画布，没有绘画能力，所有的绘制都是通过js脚本实现的

默认大小

Canvas 的默认大小为300像素×150像素（宽×高，像素的单位是px）。不能通过行内样式来修改宽高，它有自己的宽高属性。

渲染上下文

<canvas>元素创造了一个固定大小的画布，它公开了一个或多个渲染上下文，其可以用来绘制和处理要展示的内容。

canvas起初是空白的。为了展示，首先脚本需要找到渲染上下文，然后在它的上面绘制。

<canvas> 元素有一个叫做 getContext() 的方法，这个方法是用来获得渲染上下文和它的绘画功能

getContext()只有一个参数，上下文的格式

var canvas = document.getElementById('tutorial');

var ctx = canvas.getContext('2d');

使用路径绘制图形需要的步骤

·首先，你需要创建路径起始点

·然后你使用画图命令去画出路径

·之后你把路径封闭

·一旦路径生成，你就能通过描边或填充路径区域来渲染图形

beginPath()

新建一条路径，生成之后，图形绘制命令被指向到路径上生成路径。

closePath()

闭合路径之后图形绘制命令又重新指向到上下文中

stroke()

通过线条来绘制图形轮廓

fill()

通过填充路径的内容区域生成实心的图形

绘制路线

步骤

·开启路径

beginPath()

·设置起始点/移动点

moveTo(x,y);

·设置'结束点'

lineTo(x,y);

·描边/填充

stroke()/fill()

·关闭路径

closePath()

**绘制矩形的方法**

rect(x,y,width,height)

需要配合 stroke() 或 fill() 方法在画布上实际地绘制矩形

fillRect(x, y, width, height)

绘制一个填充的矩形

strokeRect(x, y, width, height)

绘制一个矩形的边框

clearRect(x, y, width, height)

清除指定矩形区域，让清除部分完全透明

**圆弧**

arc(x,y,r,sAngle,eAngle,counterclockwise);

x,y 是圆心点的坐标

r是圆的半径

sAngle、eAngle是圆的起始角度和结束角度。以弧度计

counterclockwise可选。规定应该逆时针还是顺时针绘图。False = 顺时针，true = 逆时针

**样式**

strokeStyle

描边的颜色

fillStyle

填充的颜

**阴影**

shadowColor

阴影的颜色

shadowBlur

阴影的模糊级别

shadowOffsetX

阴影的水平距离

shadowOffsetY

阴影的垂直距离

**线条的样式**

lineCap

设置或返回线条的结束端点样式

· butt 默认。向线条的每个末端添加平直的边缘

· round 向线条的每个末端添加圆形线帽。

· square 向线条的每个末端添加正方形线帽

lineWidth

当前线条的宽度，以像素计。

**旋转**

rotate(angle);

旋转角度，以弧度计。

角度转弧度公式

deg\*Math.PI/180

如需旋转 5 度，可规定下面的公式：5\*Math.PI/180。

**重新映射画布（0,0）**

translate(x,y)*;*

**绘制文本方法**

fillText(text,x,y,maxWidth)

在画布上绘制“被填充的”文本

text 规定在画布上输出的文本。**x**：开始绘制文本的**x**坐标位置（相对于画布）

y:开始绘制文本的Y坐标位置（相对于画布。

maxWidth 可选。允许的最大文本宽度，以像素计。

strokeText()

在画布上绘制文本（无填充）

同上

**绘制文本的属性**

textAlign

**start** 默认。文本在指定的位置开始。**end** 文本在指定的位置结束。

center 文本的中心被放置在指定的位置。**left** 文本左对齐。**right** 文本右对齐。

textBaseline

alphabetic默认。文本基线是普通的字母基线top 文本基线是 em 方框的顶端。。

hanging 文本基线是悬挂基线。

middle 文本基线是 **em** 方框的正中。

ideographic 文本基线是表意基线。bottom 文本基线是 em 方框的底端。

font 属性设置或返回画布上文本内容的当前字体属性

|  |  |
| --- | --- |
| **值** | **说明** |
| font-style | 规定字体样式。可能的值：normal、italic、oblique |
| font-variant | 规定字体变体。可能的值：normal、small-caps |
| font-weight | 规定字体的粗细。可能的值normal、bold、600等 |
| font-size / line-heigh | 规定字号和行高，以像素计 |
| font-family | 规定字体系列 |

**渐变**

createLinearGradient(x0,y0,x1,y1); 线性渐变

**x**0渐变开始点的**x**坐标

y0 渐变开始点的y坐标**x**1 渐变结束点的 **x** 坐标

y1 渐变结束点的 y 坐标

createRadialGradient(x0,y0,r0,x1,y1,r1);放射性渐变

**x**0 渐变的开始圆的 **x** 坐标

y0 渐变的开始圆的 y 坐标

r0 开始圆的半径**x**1 渐变的结束圆的 **x** 坐标

y1 渐变的结束圆的 y 坐标

r1 结束圆的半径

addColorStop(stop,color);

**stop** 介于 0.0 与 1.0 之间的值，表示渐变中开始与结束之间的位置。**color** 在结束位置显示的 **CSS** 颜色值

var grd=ctx.createLinearGradient(0,0,170,0);

grd.addColorStop(0,"black");

grd.addColorStop("0.3","magenta");

grd.addColorStop("0.5","blue");

grd.addColorStop("0.6","green");

grd.addColorStop("0.8","yellow");

grd.addColorStop(1,"red");

ctx.fillStyle=grd;

ctx.fillRect(20,20,150,100);

**save()**

保存当前环境的状态

**restore()**

返回之前保存过的路径状态和属性

**绘制图像**

drawImage(img,sx,sy,swidth,sheight,x,y,width,height);

|  |  |
| --- | --- |
| **参数** | **描述** |
| img | 规定要使用的图像、画布或视频 |
| sx | 可选。开始剪切的 x 坐标位置 |
| sy | 可选。开始剪切的 y 坐标位置 |
| swidth | 可选。被剪切图像的宽度 |
| sheight | 可选。被剪切图像的高度 |
| x | 在画布上放置图像的 x 坐标位置 |
| y | 在画布上放置图像的 y 坐标位置 |
| width | 可选。要使用的图像的宽度。（伸展或缩小图像） |
| height | 可选。要使用的图像的高度。（伸展或缩小图像） |

**颜色、样式和阴影**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [fillStyle](http://www.w3school.com.cn/tags/canvas_fillstyle.asp) | 设置或返回用于填充绘画的颜色、渐变或模式 |
| [strokeStyle](http://www.w3school.com.cn/tags/canvas_strokestyle.asp) | 设置或返回用于笔触的颜色、渐变或模式 |
| [shadowColor](http://www.w3school.com.cn/tags/canvas_shadowcolor.asp) | 设置或返回用于阴影的颜色 |
| [shadowBlur](http://www.w3school.com.cn/tags/canvas_shadowblur.asp) | 设置或返回用于阴影的模糊级别 |
| [shadowOffsetX](http://www.w3school.com.cn/tags/canvas_shadowoffsetx.asp) | 设置或返回阴影距形状的水平距离 |
| [shadowOffsetY](http://www.w3school.com.cn/tags/canvas_shadowoffsety.asp) | 设置或返回阴影距形状的垂直距离 |

|  |  |
| --- | --- |
| **方法** | **描述** |
| [createLinearGradient()](http://www.w3school.com.cn/tags/canvas_createlineargradient.asp) | 创建线性渐变（用在画布内容上） |
| [createPattern()](http://www.w3school.com.cn/tags/canvas_createpattern.asp) | 在指定的方向上重复指定的元素 |
| [createRadialGradient()](http://www.w3school.com.cn/tags/canvas_createradialgradient.asp) | 创建放射状/环形的渐变（用在画布内容上） |
| [addColorStop()](http://www.w3school.com.cn/tags/canvas_addcolorstop.asp) | 规定渐变对象中的颜色和停止位置 |

**线条样式**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [lineCap](http://www.w3school.com.cn/tags/canvas_linecap.asp) | 设置或返回线条的结束端点样式 |
| [lineJoin](http://www.w3school.com.cn/tags/canvas_linejoin.asp) | 设置或返回两条线相交时，所创建的拐角类型 |
| [lineWidth](http://www.w3school.com.cn/tags/canvas_linewidth.asp) | 设置或返回当前的线条宽度 |
| [miterLimit](http://www.w3school.com.cn/tags/canvas_miterlimit.asp) | 设置或返回最大斜接长度 |

**矩形**

|  |  |
| --- | --- |
| **方法** | **描述** |
| [rect()](http://www.w3school.com.cn/tags/canvas_rect.asp) | 创建矩形 |
| [fillRect()](http://www.w3school.com.cn/tags/canvas_fillrect.asp) | 绘制“被填充”的矩形 |
| [strokeRect()](http://www.w3school.com.cn/tags/canvas_strokerect.asp) | 绘制矩形（无填充） |
| [clearRect()](http://www.w3school.com.cn/tags/canvas_clearrect.asp) | 在给定的矩形内清除指定的像素 |

**路径**

|  |  |
| --- | --- |
| **方法** | **描述** |
| [fill()](http://www.w3school.com.cn/tags/canvas_fill.asp) | 填充当前绘图（路径） |
| [stroke()](http://www.w3school.com.cn/tags/canvas_stroke.asp) | 绘制已定义的路径 |
| [beginPath()](http://www.w3school.com.cn/tags/canvas_beginpath.asp) | 起始一条路径，或重置当前路径 |
| [moveTo()](http://www.w3school.com.cn/tags/canvas_moveto.asp) | 把路径移动到画布中的指定点，不创建线条 |
| [closePath()](http://www.w3school.com.cn/tags/canvas_closepath.asp) | 创建从当前点回到起始点的路径 |
| [lineTo()](http://www.w3school.com.cn/tags/canvas_lineto.asp) | 添加一个新点，然后在画布中创建从该点到最后指定点的线条 |
| [clip()](http://www.w3school.com.cn/tags/canvas_clip.asp) | 从原始画布剪切任意形状和尺寸的区域 |
| [quadraticCurveTo()](http://www.w3school.com.cn/tags/canvas_quadraticcurveto.asp) | 创建二次贝塞尔曲线 |
| [bezierCurveTo()](http://www.w3school.com.cn/tags/canvas_beziercurveto.asp) | 创建三次方贝塞尔曲线 |
| [arc()](http://www.w3school.com.cn/tags/canvas_arc.asp) | 创建弧/曲线（用于创建圆形或部分圆） |
| [arcTo()](http://www.w3school.com.cn/tags/canvas_arcto.asp) | 创建两切线之间的弧/曲线 |
| [isPointInPath()](http://www.w3school.com.cn/tags/canvas_ispointinpath.asp) | 如果指定的点位于当前路径中，则返回 true，否则返回 false |

**转换**

|  |  |
| --- | --- |
| **方法** | **描述** |
| [scale()](http://www.w3school.com.cn/tags/canvas_scale.asp) | 缩放当前绘图至更大或更小 |
| [rotate()](http://www.w3school.com.cn/tags/canvas_rotate.asp) | 旋转当前绘图 |
| [translate()](http://www.w3school.com.cn/tags/canvas_translate.asp) | 重新映射画布上的 (0,0) 位置 |
| [transform()](http://www.w3school.com.cn/tags/canvas_transform.asp) | 替换绘图的当前转换矩阵 |
| [setTransform()](http://www.w3school.com.cn/tags/canvas_settransform.asp) | 将当前转换重置为单位矩阵。然后运行 transform() |

**文本**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [font](http://www.w3school.com.cn/tags/canvas_font.asp) | 设置或返回文本内容的当前字体属性 |
| [textAlign](http://www.w3school.com.cn/tags/canvas_textalign.asp) | 设置或返回文本内容的当前对齐方式 |
| [textBaseline](http://www.w3school.com.cn/tags/canvas_textbaseline.asp) | 设置或返回在绘制文本时使用的当前文本基线 |

|  |  |
| --- | --- |
| **方法** | **描述** |
| [fillText()](http://www.w3school.com.cn/tags/canvas_filltext.asp) | 在画布上绘制“被填充的”文本 |
| [strokeText()](http://www.w3school.com.cn/tags/canvas_stroketext.asp) | 在画布上绘制文本（无填充） |
| [measureText()](http://www.w3school.com.cn/tags/canvas_measuretext.asp) | 返回包含指定文本宽度的对象 |

**图像绘制**

|  |  |
| --- | --- |
| **方法** | **描述** |
| [drawImage()](http://www.w3school.com.cn/tags/canvas_drawimage.asp) | 向画布上绘制图像、画布或视频 |

**像素操作**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [width](http://www.w3school.com.cn/tags/canvas_imagedata_width.asp) | 返回 ImageData 对象的宽度 |
| [height](http://www.w3school.com.cn/tags/canvas_imagedata_height.asp) | 返回 ImageData 对象的高度 |
| [data](http://www.w3school.com.cn/tags/canvas_imagedata_data.asp) | 返回一个对象，其包含指定的 ImageData 对象的图像数据 |

|  |  |
| --- | --- |
| **方法** | **描述** |
| [createImageData()](http://www.w3school.com.cn/tags/canvas_createimagedata.asp) | 创建新的、空白的 ImageData 对象 |
| [getImageData()](http://www.w3school.com.cn/tags/canvas_getimagedata.asp) | 返回 ImageData 对象，该对象为画布上指定的矩形复制像素数据 |
| [putImageData()](http://www.w3school.com.cn/tags/canvas_putimagedata.asp) | 把图像数据（从指定的 ImageData 对象）放回画布上 |

**合成**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [globalAlpha](http://www.w3school.com.cn/tags/canvas_globalalpha.asp) | 设置或返回绘图的当前 alpha 或透明值 |
| [globalCompositeOperation](http://www.w3school.com.cn/tags/canvas_globalcompositeoperation.asp) | 设置或返回新图像如何绘制到已有的图像上 |

**其他**

|  |  |
| --- | --- |
| **方法** | **描述** |
| save() | 保存当前环境的状态 |
| restore() | 返回之前保存过的路径状态和属性 |
| createEvent() |  |
| getContext() |  |
| toDataURL() |  |

globalCompositeOperation 属性设置或返回如何将一个源（新的）图像绘制到目标（已有）的图像上。

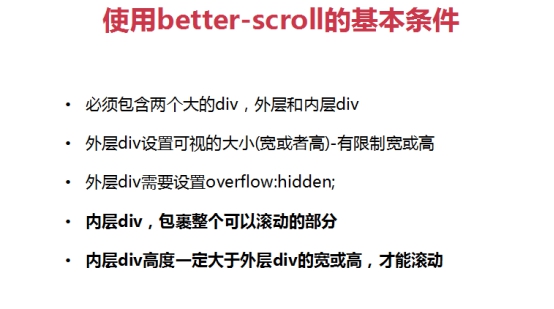
属性值

|  |  |
| --- | --- |
| **值** | **描述** |
| source-over | 默认。在目标图像上显示源图像。 |
| source-atop | 在目标图像顶部显示源图像。源图像位于目标图像之外的部分是不可见的。 |
| source-in | 在目标图像中显示源图像。只有目标图像内的源图像部分会显示，目标图像是透明的。 |
| source-out | 在目标图像之外显示源图像。只会显示目标图像之外源图像部分，目标图像是透明的。 |
| destination-over | 在源图像上方显示目标图像。 |
| destination-atop | 在源图像顶部显示目标图像。源图像之外的目标图像部分不会被显示。 |
| destination-in | 在源图像中显示目标图像。只有源图像内的目标图像部分会被显示，源图像是透明的。 |
| destination-out | 在源图像外显示目标图像。只有源图像外的目标图像部分会被显示，源图像是透明的。 |
| lighter | 显示源图像 + 目标图像。 |
| copy | 显示源图像。忽略目标图像。 |
| xor | 使用异或操作对源图像与目标图像进行组合。 |

**better-scroll 是什么**

better-scroll 是一款重点解决移动端（已支持 PC）各种滚动场景需求的插件。它的核心是借鉴的 iscroll 的实现，它的 API 设计基本兼容 iscroll，在 iscroll 的基础上又扩展了一些 feature 以及做了一些性能优化。

better-scroll 是基于原生 JS 实现的，不依赖任何框架。它编译后的代码大小是 63kb，压缩后是 35kb，gzip 后仅有 9kb，是一款非常轻量的 JS lib。



**Swiper**

effect:”cube”,  //3D,默认silde

effect:”coverflow”,

spaceBetween: 10,  //间距

slidesPerView: 4,

slidesPerGroup: 4