

# 梯度

颜色渐变。

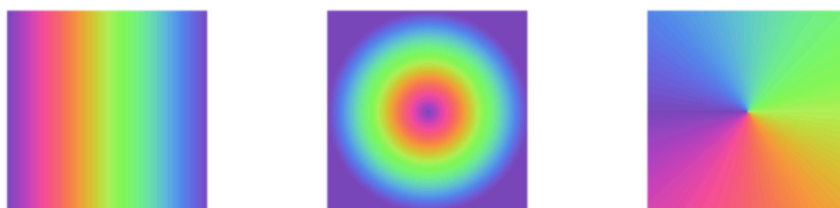
Typst 通过 [gradient.linear 函数](#) 支持线性渐变，通过 [gradient.radial 函数](#) 支持径向渐变，通过 [gradient.conic 函数](#) 支持圆锥渐变。

渐变可用于以下目的：

- 作为绘制形状内部的填充：`rect(fill: gradient.linear(...))`
- 作为绘制形状轮廓的笔触：`rect(stroke: 1pt + gradient.linear(...))`
- 作为文本的填充：`set text(fill: gradient.linear(...))`
- 作为颜色映射表，您可以从以下[位置进行采样](#)：`gradient.linear(...).sample(0.5)`

## 例子

```
#stack(  
  dir: ltr,  
  spacing: 1fr,  
  square(fill: gradient.linear(..color.map.rainbow)),  
  square(fill: gradient.radial(..color.map.rainbow)),  
  square(fill: gradient.conic(..color.map.rainbow)),  
)
```



渐变在文本上也受支持，但仅当将相对性设置为（默认值）或时。若要创建逐字或逐字形渐变，可以手动或通过显示规则将文本的字词或字符换行到框中。auto"parent"

```
#set text(fill: gradient.linear(red, blue))
#let rainbow(content) = {
  set text(fill: gradient.linear(..color.map.rainbow))
  box(content)
}
```

This is a gradient on text, but with a `#rainbow[twist]!`

This is a gradient on text, but with a `twist!`

## 停止

渐变由一系列停靠点组成。每个停靠点都有一个颜色和一个偏移量。偏移量是 `和` 之间的比率或 `和` 之间的角度。偏移量是一个相对位置，用于确定停止点沿梯度的距离。光标的颜色是该位置的渐变颜色。在定义渐变时，可以选择省略偏移量。在这种情况下，Typst 将均匀地间隔所有停靠点。`0%100%0deg360deg`

## 相对性

和停靠点的位置取决于容器的尺寸。这个容器可以是它被涂漆的形状，也可以是最近的周围容器。这由梯度构造函数的参数控制。默认情况下，渐变是相对于它们所绘制的形状的，除非渐变应用于文本，在这种情况下，渐变是相对于最接近的祖先容器的。`0%100%relative`

Typst 按如下方式确定祖先容器：

对于放置在文档根/顶层的形状，最接近的祖先是页面本身。

对于其他形状，祖先是包含该形状的最内层块或框。这包括由显示规则和元素隐式创建的框和块。例如，旋转不会影响渐变的父级，但栅格会影响。

## 色彩空间和插值

渐变可以在任何颜色空间中插值。默认情况下，渐变在 Oklab 色彩空间中插值，这是一个感知均匀的色彩空间。这意味着渐变将被视为具有平滑的颜色渐进。这对于数据可视化特别有用。

但是，您可以选择在所需的任何受支持的颜色空间中插入渐变，但请注意，某些颜色空间不适合在颜色之间进行感知插值。选择插值空间时，请参阅下表。

色彩空间	感知一致?
<a href="#">奥克拉布</a>	是的
<a href="#">奥克奇</a>	是的
<a href="#">sRGB格式</a>	不
<a href="#">线性RGB</a>	是的
<a href="#">CMYK系列</a>	不
<a href="#">灰度</a>	是的
<a href="#">HSL公司</a>	不
<a href="#">单纯疱疹病毒</a>	不



## 方向

某些渐变对方向敏感。例如，线性渐变具有确定其方向的角度。Typst 使用顺时针角度，0° 从左到右，90° 从上到下，180° 从右到左，270° 从下到上。

```
#stack(  
  dir: ltr,  
  spacing: 1fr,  
  square(fill: gradient.linear(red, blue, angle: 0deg)),  
  square(fill: gradient.linear(red, blue, angle: 90deg)),  
  square(fill: gradient.linear(red, blue, angle: 180deg)),  
  square(fill: gradient.linear(red, blue, angle: 270deg)),  
)
```



## 预设

Typst 预定义了可用于渐变的颜色映射表。有关详细信息，请参阅颜色文档。

## 关于文件大小的注意事项

梯度可能非常大，尤其是当它们有很多停靠点时。这是因为渐变存储为颜色和偏移量列表，这可能会占用大量空间。如果您担心文件大小，则应考虑以下几点：

SVG 梯度目前编码效率低下。这将在未来得到改进。

color.hsv、color.hsl 和 color.oklch 色彩空间中的 PDF 渐变存储为 color.oklab 颜色列表，中间有额外的停靠点。这样就避免了在 PDF 文件中对这些色彩空间进行编码的需要，但它确实会为渐变添加额外的停顿，从而增加文件大小。

## 定义

### linear

创建新的线性渐变，其中颜色沿直线过渡。

渐变。[线性](#)(  
 .. [颜色](#) 数组 ,  
 空间: [任何](#) ,  
 相对: [自动](#) [str](#) ,  
 方向 ,  
 角度 ,  
 ) -> [梯度](#)

```
#rect(  
  width: 100%,  
  height: 20pt,  
  fill: gradient.linear(  
    ..color.map.viridis,  
  ),  
)
```



**stops**    [颜色](#) 或 [数组](#)    *必填*    [位置的 ?](#)    [可变性 ?](#)

渐变的颜色[停止](#)。

**space**    [任何](#)

要在其中插入渐变的颜色空间。

默认为称为 [Oklab](#) 的感知统一色彩空间。

违约: oklab

**relative**    [自动](#) 或 [str](#)

梯度的[相对位置](#)。

对于放置在文档根/顶层的元素，父元素是页面本身。对于其他元素，父元素是包含该元素的最内层块、框、列、网格或堆栈。

变体	详
"self"	梯度是相对于自身的（它自己的边界框）。
"parent"	渐变相对于其父级（父级的边界框）。

违约: [auto](#)

**dir**    [方向](#)    [位置的 ?](#)

渐变的方向。

违约: ltr

**angle**    [角度](#)    *必填*    [位置的 ?](#)

渐变的角度。

## radial

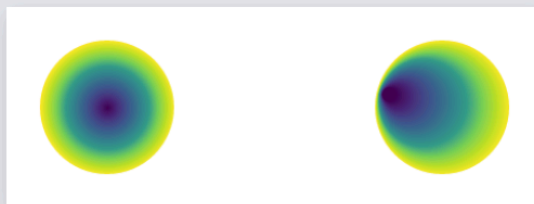
创建新的径向渐变，其中颜色从原点辐射出去。

渐变由两个圆圈定义：焦点圆圈和结束圆圈。焦点圆是一个具有中心和半径的圆，它定义了渐变开始的点，并具有第一个停止点的颜色。结束圆是具有中心和半径的圆，它定义了渐变结束的点，并具有最后一个停靠点的颜色。然后将梯度插值在这两个圆之间。focal-centerfocal-radiuscenterradius

使用这四个值，也称为起始圆的焦点和结束圆的中心和半径，我们可以定义一个比基本径向渐变具有更有趣属性的渐变。

```
渐变。径向(  
  .. 颜色 数组 ,  
  空间: 任何 ,  
  相对: 自动 str ,  
  中心: 数组 ,  
  半径: 率 ,  
  焦点中心: 自动 数组 ,  
  焦距半径: 率 ,  
) -> 梯度
```

```
#stack(  
  dir: ltr,  
  spacing: 1fr,  
  circle(fill: gradient.radial(  
    ..color.map.viridis,  
  )),  
  circle(fill: gradient.radial(  
    ..color.map.viridis,  
    focal-center: (10%, 40%),  
    focal-radius: 5%,  
  )),  
)
```



**stops** 颜色 或 数组 必填 位置的? 可变性?

渐变的颜色[停止](#)。

**space** 任何

要在其中插入渐变的颜色空间。

默认为称为 [Oklab](#) 的感知统一色彩空间。

违约: oklab

**relative** 自动 或 str

梯度的[相对位置](#)。

对于放置在文档根/顶层的元素，父元素是页面本身。对于其他元素，父元素是包含该元素的最内层块、框、列、网格或堆栈。

变体	详
<code>"self"</code> <code>"parent"</code>	梯度是相对于自身的（它自己的边界框）。 渐变相对于其父级（父级的边界框）。
违约: <code>auto</code>	
<b>center</b> <span>数组</span>	
渐变的端圆的中心。	
值 of 表示端圆在其容器内居中。(50%, 50%)	
违约: (50%, 50%)	
<b>radius</b> <span>率</span>	
渐变的端圆的半径。	
默认情况下，它设置为。结束半径必须大于焦距半径。50%	
违约: 50%	
<b>focal-center</b> <span>自动 或 数组</span>	
渐变的焦圆的中心。	
焦点中心必须位于端圆内。	
值 of 表示焦圆位于其容器内的中心。(50%, 50%)	
默认情况下，它设置为与最后一个圆的中心相同。	
违约: <code>auto</code>	
<b>focal-radius</b> <span>率</span>	
渐变的焦圆半径。	
焦点中心必须位于端圆内。	
默认情况下，它设置为。焦距半径必须小于结束半径。0%	
违约: 0%	

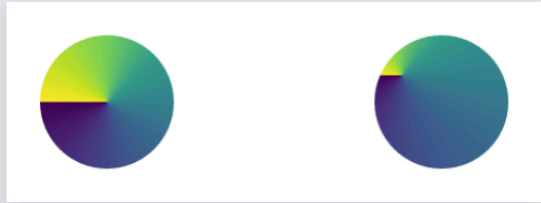
## conic

创建新的圆锥渐变，其中颜色围绕中心点径向变化。

您可以使用参数来控制渐变的中心点。默认情况下，中心点是形状的中心。`center`

渐变。圆 锥形(  
 .. 颜色 数组 ,  
 角度: 角度 ,  
 空间: 任何 ,  
 相对: 自动 str ,  
 中心: 数组 ,  
) -> 梯度

```
#stack(  
  dir: ltr,  
  spacing: 1fr,  
  circle(fill: gradient.conic(  
    ..color.map.viridis,  
  )),  
  circle(fill: gradient.conic(  
    ..color.map.viridis,  
    center: (20%, 30%),  
  )),  
)
```



**stops** 颜色 或 数组 必填 位置的? 可变性?

渐变的颜色[停止](#)。

**angle** 角度

渐变的角度。

违约: 0deg

**space** 任何

要在其中插入渐变的颜色空间。

默认为称为 [Oklab](#) 的感知统一色彩空间。

违约: oklab

**relative** 自动 或 str

梯度的[相对位置](#)。

对于放置在文档根/顶层的元素，父元素是页面本身。对于其他元素，父元素是包含该元素的最内层块、框、列、网格或堆栈。



变体	详
"self"	梯度是相对于自身的（它自己的边界框）。
"parent"	渐变相对于其父级（父级的边界框）。
违约: <b>auto</b>	
<b>center</b>	<b>数组</b>
渐变的最后一个圆的中心。	
值 of 表示端圆在其容器内居中。 <b>(50%, 50%)</b>	
违约: <b>(50%, 50%)</b>	

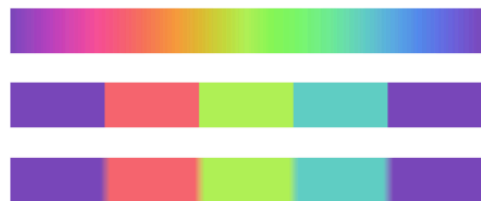
## sharp

创建此渐变的锐化版本。

锐利的渐变在颜色之间具有离散的跳跃，而不是平滑的过渡。它们对于创建预设渐变的颜色列表特别有用。

自我。锋利(  
**整数** ,  
 平滑: **率** ,  
 ) -> **梯度**

```
#set rect(width: 100%, height: 20pt)
#let grad = gradient.linear(..color.map
#rect(fill: grad)
#rect(fill: grad.sharp(5))
#rect(fill: grad.sharp(5, smoothness: 20))
```



**steps** **整数** **必填** **位置的?**

梯度中的停靠点数。

**smoothness** **率**

平滑梯度的程度。

违约: **0%**

## repeat

重复此渐变给定的次数，并可选择在每次重复时镜像它。

自我。重复(  
 整数,  
 镜子: 布尔,  
) -> 梯度

```
#circle(  
  radius: 40pt,  
  fill: gradient  
    .radial(aqua, white)  
    .repeat(4),  
)
```



**repetitions** 整数 必填 位置的?

重复渐变的次数。

**mirror** 布尔

是否在每次重复时镜像渐变。

违约: false

## kind

返回此渐变的种类。

自我。`类()` -> `功能`

## stops

返回此渐变的停靠点。

自我。`停止()` -> `数组`

## space

返回此渐变的混合空间。

自我。`空间()` -> `任何`

## relative

返回此梯度的相对位置。

自我。`相对()` -> `自动`

## angle

返回此渐变的角度。

自我。`角度()` -> `没有` `角度`

## sample

对给定位置的梯度进行采样。

该位置可以是沿梯度的位置（和 之间的**比率**）或**角度**。超出此范围的任何值都将被钳制。0%100%

```
自我。样本( 角度 率 ) -> 颜色
```

**t**    角度 或 率    必填    位置的 ?

对梯度进行采样的位置。

## samples

一次对多个位置的梯度进行采样，并以数组形式返回结果。

```
自我。样品( .. 角度 率 ) -> 数组
```

**ts**    角度 或 率    必填    位置的 ?    可变性 ?

对梯度进行采样的位置。