

颜色

特定颜色空间中的颜色。

Typst 支持：

sRGB 通过 `rgb` 函数

通过 `cmyk` 函数实现设备 CMYK

D65 Gray 通过亮度功能

Oklab 通过 `oklab` 函数

通过 `oklch` 函数的 `Oklch`

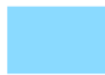
通过 `color.linear-rgb` 函数实现线性 RGB

HSL 通过 `color.hsl` 函数

HSV 通过 `color.hsv` 函数

例

```
#rect(fill: aqua)
```

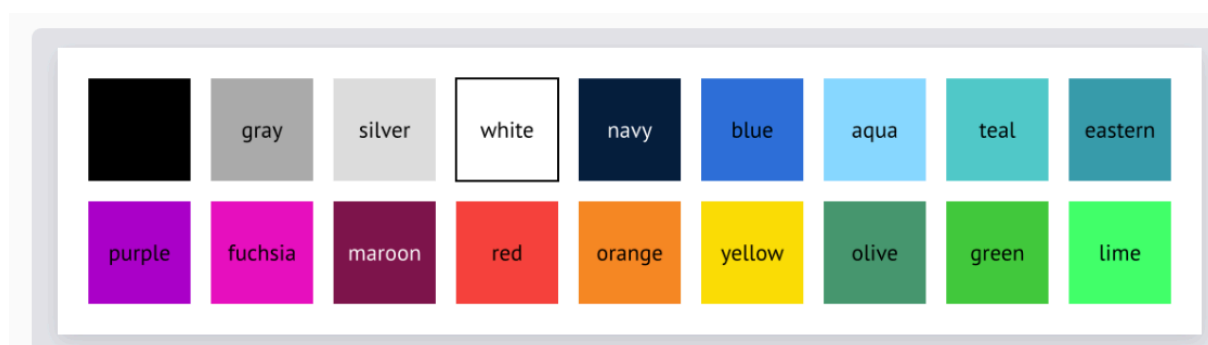


预定义颜色

Typst 定义了以下内置颜色：

颜色	定义
black	<code>luma(0)</code>
gray	<code>luma(170)</code>
silver	<code>luma(221)</code>
white	<code>luma(255)</code>
navy	<code>rgb("#001f3f")</code>
blue	<code>rgb("#0074d9")</code>
aqua	<code>rgb("#7fdbff")</code>
teal	<code>rgb("#39cccc")</code>
eastern	<code>rgb("#239dad")</code>
purple	<code>rgb("#b10dc9")</code>
fuchsia	<code>rgb("#f012be")</code>
maroon	<code>rgb("#85144b")</code>
red	<code>rgb("#ff4136")</code>
orange	<code>rgb("#ff851b")</code>
yellow	<code>rgb("#ffdc00")</code>
olive	<code>rgb("#3d9970")</code>
green	<code>rgb("#2ecc40")</code>
lime	<code>rgb("#01ff70")</code>

预定义的颜色和最重要的颜色构造函数在全局可用，也可以在颜色类型的范围内使用，因此您可以编写 `或只编写 .color.redred`



预定义的色彩映射表

Typst 还包括许多可用于渐变的预设颜色图。这些只是在模块中定义的颜色数组。`color.map`

```
#circle(fill: gradient.linear(..color.map.crest))
```



地图 详 turbo 感知上均匀的彩虹状彩色图。有关详细信息，请阅读这篇博文。

cividis 蓝色到灰色到黄色的颜色图。有关详细信息，请参阅此博客文章。

rainbow 在整个色谱中循环。通过将插值颜色空间设置为 HSL 来最好使用此颜色映射表。彩虹渐变不适合数据可视化，因为它在感知上并不均匀，因此读者不清楚值之间的差异。它只能用于装饰目的。

spectral 红色到黄色到蓝色的颜色图。

viridis 紫色到蓝绿色到黄色的颜色图。

inferno 从黑色到红色到黄色的颜色图。

magma 黑色到紫色到黄色的颜色图。

plasma 紫色到粉红色到黄色的颜色图。

rocket 从黑到红到白的彩色地图。

mako 黑色到蓝绿色到黄色的颜色图。

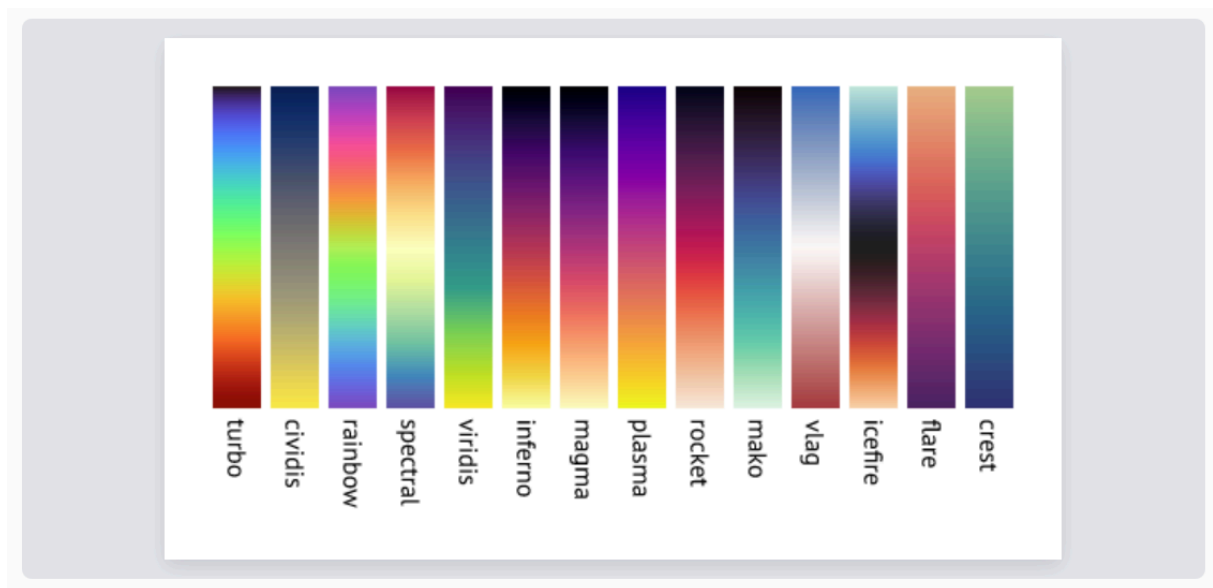
vlag 浅蓝色到白色到红色的颜色图。

icefire 浅蓝绿色到黑色到黄色的颜色图。

flare 感知上一致的橙色到紫色的颜色图。

crest 蓝色到白色到红色的颜色映射。

一些流行的预设不包括在内，因为它们可免费许可下不可用。其他的，如 Jet，不包括在内，因为它们对色盲不友好。随意使用或创建包含其他对您有用的预设的包！



定义

luma

创建灰度颜色。

灰度颜色在内部由单个组件表示。lightness

这些组件也可以使用 components 方法获得。

```
颜色。亮度(
    整数 率,
    率,
    颜色,
) -> 颜色
```

```
#for x in range(250, step: 50) {
    box(square(fill: luma(x)))
}
```



lightness 整数 或 率 必填 位置的 ?

轻盈部分。

alpha 率 必填 位置的 ?

alpha 组件。

color 颜色 必填 位置的 ?

或者：要转换为灰度的颜色。

如果给出，则不应给出。lightness

oklab

创建 Oklab 颜色。

此色彩空间非常适合以下用例：

色彩处理，例如饱和度，同时保持感知色调

创建具有均匀感知亮度的灰度图像

创建平滑均匀的颜色过渡和渐变

线性 Oklab 颜色在内部由四个分量组成的数组表示：

亮度（比率）

a（浮点数或比率。比率相对于；意义等于 0.450%0.2）

b（浮点数或比率。比率相对于；意义等于 0.450%0.2）alpha（比率）

这些组件也可以使用 components 方法获得。

颜色。OKLAB公司C

```
率 ,  
浮 率 ,  
浮 率 ,  
率 ,  
颜色 ,  
) -> 颜色
```

```
#square(  
  fill: oklab(27%, 20%, -3%, 50%)  
)
```



lightness 率 必填 位置的 ?

轻盈部分。

a 浮 或 率 必填 位置的 ?

a（“绿色/红色”）组件。

b 浮 或 率 必填 位置的 ?

b（“蓝色/黄色”）组件。

alpha 率 必填 位置的 ?

alpha 组件。

color 颜色 必填 位置的 ?

或者：要转换为 Oklab 的颜色。

如果给出，则不应给出单个组件。

oklch

创建 Oklch 颜色。

此色彩空间非常适合以下用例：

涉及亮度、色度和色调的色彩处理

创建具有均匀感知亮度的灰度图像

创建平滑均匀的颜色过渡和渐变

线性 Oklch 颜色在内部由四个分量组成的数组表示：

亮度（比率）

色度（浮点数或比率。比率相对于 ; 意义等于 0.450%0.2）

色调（角度）

alpha（比率）

这些组件也可以使用 components 方法获得。

颜色。 `oklch`(

```
    率,  
    浮 率,  
    角度,  
    率,  
    颜色,  
  ) -> 颜色
```

```
#square(  
  fill: oklch(40%, 0.2, 160deg, 50%)  
)
```



lightness 率 必填 位置的 ?

轻盈部分。

chroma 浮 或 率 必填 位置的 ?

色度组件。

hue 角度 必填 位置的 ?

色调组件。

alpha 率 必填 位置的 ?

alpha 组件。

color 颜色 必填 位置的 ?

或者：要转换为 Oklch 的颜色。

如果给出，则不应给出单个组件。

linear-rgb

创建具有线性亮度的 RGB (A) 颜色。

此颜色空间类似于 sRGB，但区别在于颜色分量没有经过伽玛校正。这样可以更轻松地执行颜色操作，例如混合和插值。虽然，您应该更愿意使用 `oklab` 函数来执行这些操作。

线性 RGB (A) 颜色在内部由四个分量组成的数组表示：

红色（比率）

绿色（比率）

蓝色（比率）

alpha（比率）

这些组件也可以使用 `components` 方法获得。

颜色。线性RGB(

```
    整数 率 ,  
    整数 率 ,  
    整数 率 ,  
    整数 率 ,  
    颜色 ,  
    ) -> 颜色
```

```
#square(fill: color.linear-rgb(  
    30%, 50%, 10%,  
))
```



red 整数 或 率 必填 位置的 ?

红色组件。

green 整数 或 率 必填 位置的 ?

绿色组件。

blue 整数 或 率 必填 位置的 ?

蓝色组件。

alpha 整数 或 率 必填 位置的 ?

alpha 组件。

color 颜色 必填 位置的 ?

或者：转换为线性 RGB (A) 的颜色。

如果给出，则不应给出单个组件。

rgb

创建 RGB (A) 颜色。

颜色在 sRGB 色彩空间中指定。

RGB (A) 颜色在内部由四个分量组成的数组表示：

红色 (比率)

绿色 (比率)

蓝色 (比率)

alpha (比率)

这些组件也可以使用 components 方法获得。

颜色。RGB的C

```
    整数 率 ,  
    整数 率 ,  
    整数 率 ,  
    整数 率 ,  
    str ,  
    颜色 ,  
    ) -> 颜色
```

```
#square(fill: rgb("#b1f2eb"))  
#square(fill: rgb(87, 127, 230))  
#square(fill: rgb(25%, 13%, 65%))
```



red 整数 或 率 必填 位置的 ?

红色组件。

green 整数 或 率 必填 位置的 ?

绿色组件。

blue 整数 或 率 必填 位置的 ?

蓝色组件。

alpha 整数 或 率 必填 位置的 ?

alpha 组件。

hex str 必填 位置的 ?

或者：十六进制表示法中的颜色。

接受 3 位、4 位、6 位或 8 位十六进制数字，以及可选的前导哈希值。

如果给出，则不应给出单个组件。

color 颜色 必填 位置的 ?

或者：要转换为 RGB (a) 的颜色。

如果给出，则不应给出单个组件。

cmyk

创建 CMYK 颜色。

如果要面向特定打印机，这将非常有用。转换为 RGB 进行显示预览可能与打印机再现颜色的方式不同。

CMYK 颜色在内部由四个组件组成的数组表示：

青色（比率）

洋红色（比率）

黄色（比率）

键（比率）

这些组件也可以使用 components 方法获得。

颜色。CMYK公司C

```
率 ,  
率 ,  
率 ,  
率 ,  
颜色 ,  
) -> 颜色
```

```
#square(  
  fill: cmyk(27%, 0%, 3%, 5%)  
)
```



cyan 率 必填 位置的 ?

青色分量。

magenta 率 必填 位置的 ?

洋红色组件。

yellow 率 必填 位置的 ?

黄色组件。

key 率 必填 位置的 ?

关键组件。

color 颜色 必填 位置的 ?

或者：要转换为 CMYK 的颜色。

如果给出，则不应给出单个组件。

hsl

创建 HSL 颜色。

此色彩空间可用于按色调、饱和度和亮度指定颜色。它对于色彩处理也很有用，例如在保持感知色调的同时保持饱和度。

HSL 颜色在内部由四个分量组成的数组表示：

色调（角度）

饱和度（比率）

亮度（比率）

alpha（比率）

这些组件也可以使用 components 方法获得。

颜色。HSL公司(

```
    角度 ,  
    整数 率 ,  
    整数 率 ,  
    整数 率 ,  
    颜色 ,  
  ) -> 颜色
```

```
#square(  
  fill: color.hsl(30deg, 50%, 60%)  
)
```



hue 角度 必填 位置的 ?

色调角度。

saturation 整数 或 率 必填 位置的 ?

饱和度分量。

lightness 整数 或 率 必填 位置的 ?

轻盈部分。

alpha 整数 或 率 必填 位置的 ?

alpha 组件。

color 颜色 必填 位置的 ?

或者：要转换为 HSL 的颜色。

如果给出，则不应给出单个组件。

hsv

创建 HSV 颜色。

此颜色空间可用于按色调、饱和度和值指定颜色。它对于色彩处理也很有用，例如在保持感知色调的同时保持饱和度。

HSV 颜色在内部由四个分量组成的数组表示：

色调（角度）

饱和度（比率）

值（比率）

alpha（比率）

这些组件也可以使用 components 方法获得。

颜色。单纯疱疹病毒（

```
    角度 ,  
    整数 率 ,  
    整数 率 ,  
    整数 率 ,  
    颜色 ,  
  ) -> 颜色
```

```
#square(  
  fill: color.hsv(30deg, 50%, 60%)  
)
```



hue 角度 必填 位置的 ?

色调角度。

saturation 整数 或 率 必填 位置的 ?

饱和度分量。

value 整数 或 率 必填 位置的 ?

值组件。

alpha 整数 或 率 必填 位置的 ?

alpha 组件。

color 颜色 必填 位置的 ?

或者：要转换为 HSL 的颜色。

如果给出，则不应给出单个组件。

components

提取此颜色的成分。

此数组的大小和值取决于颜色空间。您可以使用空间来获取色彩空间。下面是色彩空间及其组件的表格：

色彩空间	C1级	C2级	C3级	C4级
luma	亮度			
oklab	亮度	a	b	阿尔法
oklch	亮度	色度	色调	阿尔法
linear-rgb	红	绿	蓝	阿尔法
rgb	红	绿	蓝	阿尔法
cmyk	青色	品红	黄色	钥匙
hsl	色调	饱和	亮度	阿尔法
hsv	色调	饱和	价值	阿尔法

有关每个单独值的含义和类型，请参阅相应颜色空间的文档。alpha 组件是可选的，仅当参数为 时才包含。返回数组的长度取决于分量的数量以及是否包含 alpha 分量。alphatru

自我。[组件](#)(阿尔法: [布尔](#)) -> [数组](#)

```
// note that the alpha component is included by default
#(rgb(40%, 60%, 80%).components() == (40%, 60%, 80%, 100%))
```

false

alpha [布尔](#)

是否包含 alpha 组件。

违约: [true](#)

space

返回此颜色空间的构造函数:

- [luma](#)
- [oklab](#)
- [oklch](#)
- [linear-rgb](#)
- [rgb](#)
- [cmyk](#)
- [hsl](#)
- [hsv](#)

自我。空间() -> 任何

```
#let color = cmyk(1%, 2%, 3%, 4%)  
#(color.space() == cmyk)
```

true

to-hex

返回颜色的 RGB (A) 十六进制表示形式 (如 或)。如果 alpha 分量 (中的最后两位数字) 等于 (255 / 100%)，则省略它。#ffaa32#020304fe#020304feff

自我。到十六进制() -> str

lighten

按给定因子使颜色变亮。

自我。减轻(率) -> 颜色

factor 率 必填 位置的?

使颜色变亮的因素。

darken

按给定因子使颜色变暗。

自我。变暗(率) -> 颜色

factor 率 必填 位置的?

使颜色变暗的因素。

saturate

按给定因子增加颜色的饱和度。

自我。饱和(率) -> 颜色

factor 率 必填 位置的?

使颜色饱和的因子。

desaturate

将颜色的饱和度降低给定因子。

自我。去饱和(率) -> 颜色

factor 率 必填 位置的 🔗

降低颜色饱和度的系数。

negate

使用提供的色彩空间生成互补色。你可以把它想象成色轮上的另一面。

自我。否定(空间: 任何) -> 颜色

```
#square(fill: yellow)
#square(fill: yellow.negate())
#square(fill: yellow.negate(space: rgb))
```



space 任何

用于转换的色彩空间。默认情况下，使用感知色彩空间。

违约: oklab

rotate

将颜色的色调旋转给定角度。

```
自我。旋转(  
  角度 ,  
  空间: 任何 ,  
) -> 颜色
```

angle 角度 必填 位置的 ?

旋转色调的角度。

space 任何

用于旋转的色彩空间。默认情况下，这发生在感知色彩空间 ([oklch](#)) 中。

违约: oklch

mix

通过混合两种或多种颜色来创建颜色。

在具有色调分量 (hsl、hsv、oklch) 的色彩空间中，一次只能混合两种颜色。在这样的空间中混合两种以上的颜色会导致错误！

```
颜色。混合(  
  .. 颜色 数组 ,  
  空间: 任何 ,  
) -> 颜色
```

```
#set block(height: 20pt, width: 100%)  
#block(fill: red.mix(blue))  
#block(fill: red.mix(blue, space: rgb))  
#block(fill: color.mix(red, blue, white))  
#block(fill: color.mix((red, 70%), (blue, 30%)))
```



colors 颜色 或 数组 必填 位置的? 可变性?

颜色（可选）带有权重，指定为颜色和权重（浮点数或比率）的一对（长度为 2 的数组）。
权重不需要相加，它们相对于所有权重的总和。100%

space 任何

要混合的色彩空间。默认情况下，这发生在感知色彩空间（[oklab](#)）中。
违约: oklab

transparentize

通过给定因素使颜色更透明。

此方法相对于现有的 alpha 值。如果刻度为正，则计算。负刻度的行为类似于
`.alpha - alpha * scalecolor.opacify(-scale)`

自我。透明化(率) -> 颜色

```
#block(fill: red)[opaque]
#block(fill: red.transparentize(50%))[half red]
#block(fill: red.transparentize(75%))[quarter red]
```

opaque
half red
quarter red

scale 率 必填 位置的?

要更改 alpha 值的因子。

opacity

在给定比例下使颜色更加不透明。

此方法相对于现有的 alpha 值。如果刻度为正，则计算。负刻度的行为类似于 $\text{alpha} + \text{scale} - \text{alpha} * \text{scale}$ `color.transparentize(-scale)`

自我。不透明(率) -> 颜色

```
#let half-red = red.transparentize(50%)
#block(fill: half-red.opacity(100%))[opaque]
#block(fill: half-red.opacity(50%))[three quarters red]
#block(fill: half-red.opacity(-50%))[one quarter red]
```

opaque

three quarters red

one quarter red

scale 率 必填 位置的?

更改 alpha 值的刻度。

< 圈
上一页

椭圆 >
下一页