

# Converting Colors

RGBPercent(100%, 65%, 60%)

Have a look what the booklet for  
RGBPercent(100%, 65%, 60%)  
contains.

<b>RGBPercent(100%, 65%, 60%)</b> .....	3
<i><b>Conversions</b></i> .....	4
<i><b>Details</b></i> .....	6
<i><b>Harmonies</b></i> .....	11
<i><b>Previews</b></i> .....	23
<i><b>Color Blindness Simulation</b></i> .....	27
<i><b>CSS Examples</b></i> .....	30

# Color

**RGBPercent(100%, 65%,  
60%)**

# Conversions

## Conversions Part 1

Format	Color
Hex	FFA699
RGB	255, 166, 153
RGB Percent	100%, 65%, 60%
CMY	0.0000, 0.3500, 0.4000
CMYK	0.00, 0.35, 0.40, 0.00
HSL	8°, 100%, 80%
HSV	8°, 40%, 100%
XYZ	60.5806, 50.7415, 36.7381
YIQ	191.1290, 57.2170, 14.8250

# Conversions

## Conversions Part 2

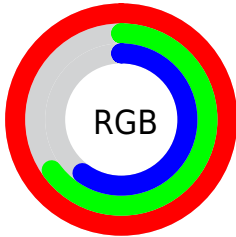
Format	Color
R <sub>Y</sub> B	255, 168, 153
Decimal	16754329
CIE Lab	76.52, 31.49, 20.29
CIE LCh	77, 37.462, 32.786
Yxy	50.7415, 0.4092, 0.3427
Android (android.graphics.Color)	4294944409 (0xFFFFA699)
YUV	191.1290, -18.7976, 56.0149
Hunter-Lab	71.2331, 27.1484, 19.2846

# Details

The RGBPercent color **100%, 65%, 60%** is a light color, and the websafe version is hex **FF9999**. A complement of this color would be **60%, 95%, 100%**, and the grayscale version is **75%, 75%, 75%**.

A 20% lighter version of the original color is **100%, 87%, 81%**, and **77%, 44%, 40%** is the 20% darker color. If you saturate the color by 10%, you get **100%, 56%, 50%**, and if you desaturate by 10%, it is **100%, 74%, 70%**.

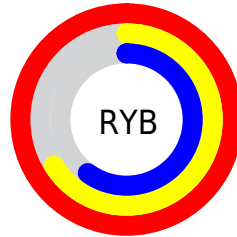
# Distribution



Red (100%)

Green (65%)

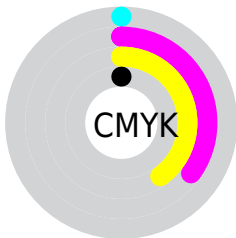
Blue (60%)



Red (100%)

Yellow (66%)

Blue (60%)

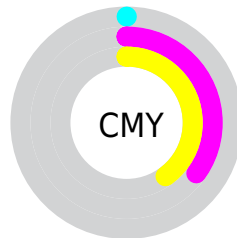


Cyan (0%)

Magenta (35%)

Yellow (40%)

Black (0%)



Cyan (0%)

Magenta (35%)

Yellow (40%)

# Brightness & Saturation Gradients

These gradients show how the RGBPercent color 100%, 65%, 60% changes by changing the brightness by 10 percent. The first figure shows a shift by +10% for each color and the second figure -10%.

Similar to the brightness gradients but the following saturation gradients show a change of the RGBPercent color 100%, 65%, 60% by changing the saturation by 10% instead.



100%, 65%, 60%

100%, 65%, 60%

100%, 100%,  
100%

88%, 55%, 50%

100%, 87%, 81%

77%, 44%, 40%

100%, 98%, 93%

65%, 35%, 31%

55%, 25%, 22%

44%, 15%, 13%

33%, 5%, 5%

22%, 0%, 0%

12%, 0%, 0%

0%, 0%, 0%

100%, 65%, 60%

100%, 65%, 60%

100%, 56%, 50%

100%, 74%, 70%

100%, 47%, 40%

100%, 82%, 80%

100%, 39%, 30%

100%, 91%, 90%

100%, 30%, 20%

100%, 100%,  
100%

100%, 21%, 10%

100%, 13%, 0%

# Harmonies

## Analogous

The Analogous color harmony consists of three colors that are next to each other on the color wheel.



100%, 64%, 73%



100%, 65%, 60%



93%, 69%, 50%

# Triad

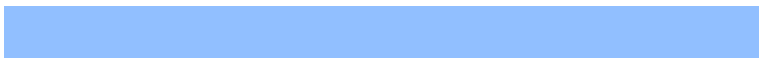
The Triadic color harmony groups three colors that are evenly spaced from another and form a triangle on the color wheel.



100%, 65%, 60%



51%, 80%, 61%



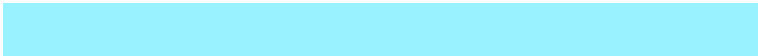
57%, 75%, 100%

# Complementary

The Complementary color scheme is a pair of colors which are on the opposite of each other on the color wheel.



100%, 65%, 60%



60%, 95%, 100%

# Split Complementary

Split-complementary colors differ from the complementary color scheme. The scheme consists of three colors, the original color and two neighbors of the complement color.



35%, 79%, 98%



100%, 65%, 60%



35%, 82%, 75%

# Square

The Square scheme is like the rectangle color scheme, but the four colors are evenly spaced on the color wheel.



100%, 65%, 60%



67%, 78%, 51%



24%, 81%, 88%



77%, 70%, 97%

# Rectangle

The Rectangle color scheme consists of four colors that form a rectangle on the color wheel.



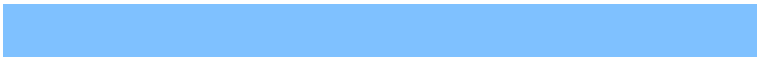
100%, 65%, 60%



86%, 72%, 47%



24%, 81%, 88%



50%, 76%, 100%

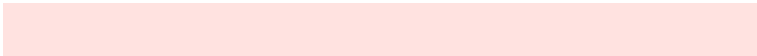


# Sweetspot

The Sweet Spot groups the original color and five complimentary colors.



100%, 65%, 60%



100%, 89%, 88%



100%, 60%, 95%



50%, 44%, 43%



0%, 0%, 0%



50%, 50%, 50%



# Same Dimension

The Same Dimension uses a secret algorithm to generate beautiful new colors.



100%, 65%, 60%



100%, 58%, 52%



100%, 85%, 60%



50%, 45%, 45%



75%, 9%, 0%



25%, 3%, 0%



# Inverse Universe

The Inverse Universe completely reimagines the original color for something new.



60%, 95%, 100%



52%, 94%, 100%



60%, 75%, 100%



45%, 49%, 50%



0%, 65%, 75%



0%, 22%, 25%



# Previews

## White Background



This preview shows how the RGBPercent color 100%, 65%, 60% looks on a white background.

## Color Contrast Check

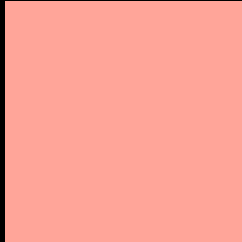
Large Text (above 18pt) WCAG AA × Fail

Any Text WCAG AA × Fail

Large Text (above 18pt) WCAG AAA × Fail

Any Text WCAG AAA × Fail

# Black Background



This preview shows how the RGBPercent color 100%, 65%, 60% looks on a black background.

## Color Contrast Check

Large Text (above 18pt) WCAG AA ✓ Pass

Any Text WCAG AA ✓ Pass

Large Text (above 18pt) WCAG AAA ✓ Pass

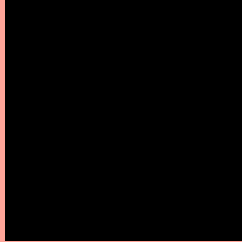
Any Text WCAG AAA ✓ Pass

If you want to check with other color combinations, try the [Color Contrast Checker](#).



# RGBPercent 100%, 65%, 60%

## Background



This preview shows how black text looks on a background with the RGBPercent color 100%, 65%, 60%.



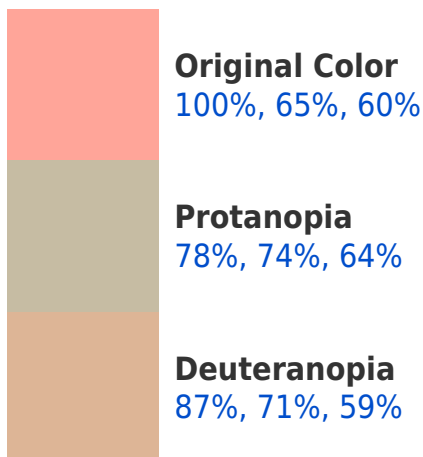
This preview shows how white text looks on a background with the RGBPercent color 100%, 65%, 60%.


60%.

# Color Blindness Simulation

Color vision deficiency is a very complex topic, and I could not describe the different causes any better than Wikipedia does, so if you want to learn more, you should check out their [article about color blindness](#).

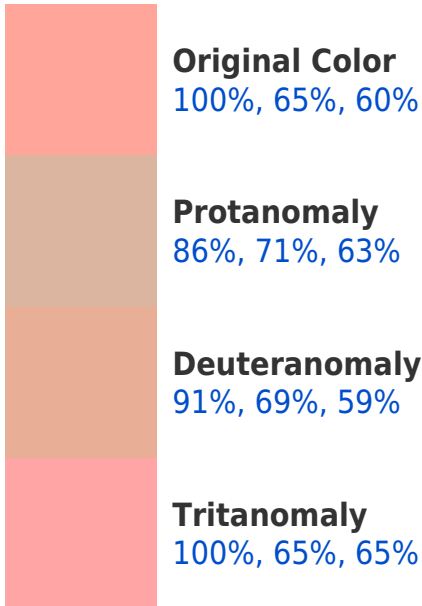
## Dichromacy



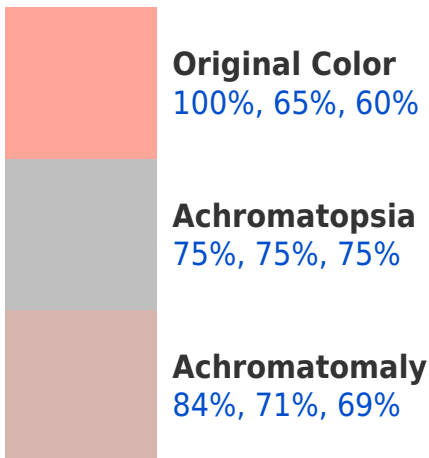


**Tritanopia**  
100%, 64%, 69%

# Trichromacy



# Monochromacy



# CSS Examples

## Text

The CSS property to change the color of the text to RGBPercent 100%, 65%, 60% is called "color". The color property can be set on classes, ids or directly on the HTML element.

This example shows how text in the color `rgb(100%, 65%, 60%)` looks like.

```
.text, #text, p{  
    color:rgb(100%, 65%, 60%)  
}
```

If you want to add a text shadow in that color use the text-shadow property, you can generate a text shadow directly with our [CSS Text Shadow Generator](#).

Here you see how black text with a 4 pixel rgb(100%, 65%, 60%) colored shadow looks like.

```
.shadow{ text-shadow: 4px 4px 2px  
rgb(100%, 65%, 60%) }
```

## Border

The CSS property to change the border of an element to RGBPercent 100%, 65%, 60% is called "border". The border property can be set on classes, ids or directly on the HTML element.

This example shows the color as border, it can be applied via the CSS property "border" or "border-color".

```
.border, #border, table{ border:4px solid  
rgb(100%, 65%, 60%) }
```

If only the border color should be changed use the property border-color.

```
.border{ border-color:rgb(100%, 65%, 60%)
}
```

If you want to add a box shadow in that color use:

Here you see how a box with a 4 pixel rgb(100%, 65%, 60%) colored shadow looks like.

```
.boxshadow{ -moz-box-shadow:4px 4px 4px
4px rgb(100%, 65%, 60%); -webkit-box-
shadow:4px 4px 4px 4px rgb(100%, 65%,
60%); box-shadow:4px 4px 4px 4px rgb(100%,
65%, 60%) }
```



# Background

The CSS property to change the background color of an element to RGBPercent 100%, 65%, 60% is called "background". The background property can be set on classes, ids or directly on the HTML element.

```
.background, #background, body{  
background: rgb(100%, 65%, 60%) }
```

If only the background color should be changed can be used:

```
.background{ background-color: rgb(100%,  
65%, 60%) }
```

This example shows the color as background, it is applied via the CSS property "background".

To optimize and compress your CSS code, you can use our [online CSS compressor and optimizer](#) based on csstidy. If you want to create a linear or radial gradient as background or border, check our [CSS Gradient Generator](#).

Hey! You found this booklet interesting? Support Converting Colors with the new Membership Option!

The pro membership hides all ads, plus gives you double the colors in the color bucket, and more awesome pro features!

**[Learn more, Memberships starting at \\$2.50/m!](#)**

**Follow me  
on Twitter!**

@ConvertingColor