

# Random Terminal Colors



Also look into: [Random Terminal Images](#)

If You like randoms colors on the Mac Terminal, here a way to get it. The script itself is from the Internet, courtesy to [Daniel Jalkut](#).

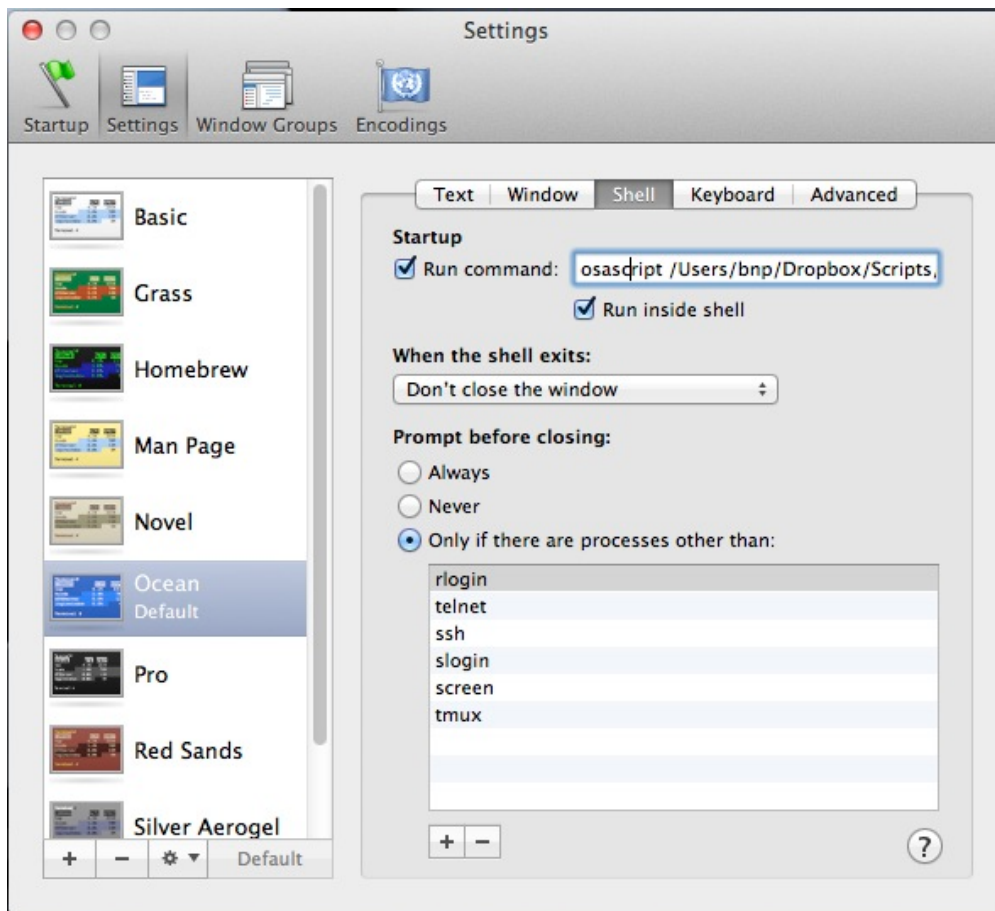
I have placed my script in `~/Dropbox/Scripts/osx/RandomColorTerminal.sct`

There are a few ways to get a new Terminal color for each started Terminal:

**A.** in `~/profile` the script can be called:

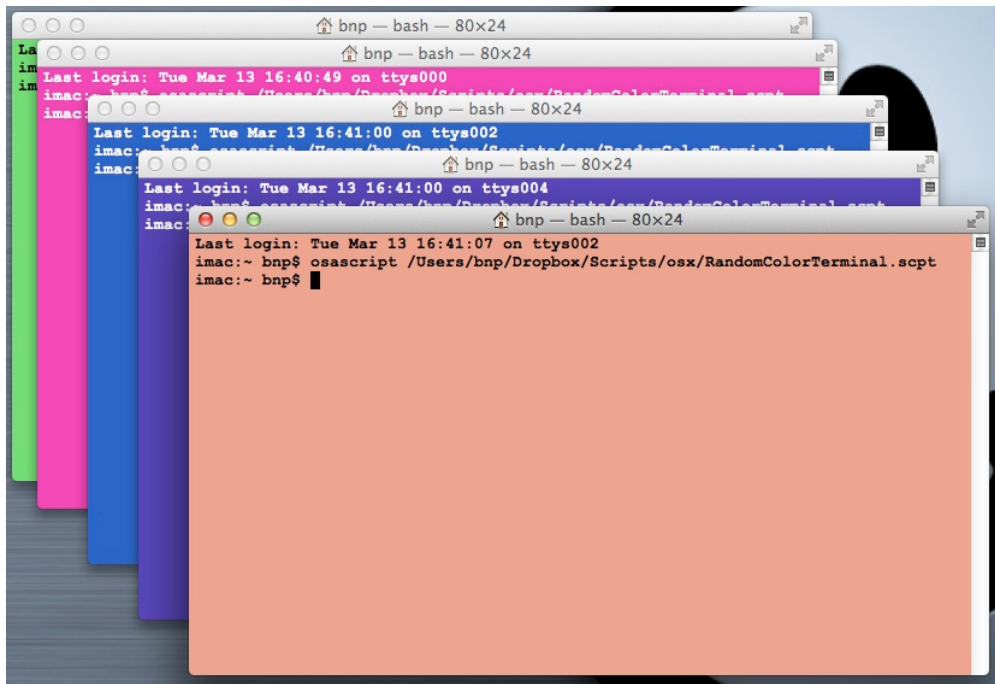
```
osascript ~/Dropbox/Scripts/osx/RandomColorTerminal.sct
```

**B.** Running it everytime a Terminal is launched via the Terminal preferences (use `osascript` to launch the script file) :



Screen Shot 2012-03-13 at 16.41.40.jpg

This gives a new color in the Terminal every time



Screen Shot 2012-03-13 at 16.41.20.jpg

Terminal.app

```

-- © Copyright 2006, Red Sweater Software. All Rights Reserved.
-- Permission to copy granted for personal use only. All copies of this script
-- must retain this copyright information and all lines of comments below, up to
-- and including the line indicating "End of Red Sweater Comments".
--
-- Any commercial distribution of this code must be licensed from the Copyright
-- owner, Red Sweater Software.
--
-- This script alters the color of the frontmost Terminal window to be something random.
--
-- End of Red Sweater Comments

-- This nasty constant might as well be a global
global kColorValueMaximum
set kColorValueMaximum to 65535

-- Choose a random color for the background
set randomRed to (random number) * kColorValueMaximum
set randomGreen to (random number) * kColorValueMaximum
set randomBlue to (random number) * kColorValueMaximum
set myBackgroundColor to {randomRed, randomGreen, randomBlue}

-- Select appropriate text colors based on that background
set {myTextColor, myBoldColor} to my ContrastingTextColors(myBackgroundColor)

-- Now inflict them on the frontmost window
tell application "Terminal"
    set targetWindow to window 1
    set background color of targetWindow to myBackgroundColor
    set cursor color of targetWindow to myTextColor
    set normal text color of targetWindow to myTextColor
    set bold text color of targetWindow to myBoldColor
end tell

on ContrastingTextColors(myColor)
    set whiteColor to {kColorValueMaximum, kColorValueMaximum, kColorValueMaximum, kColorValueMaximum}
    set lightGreyColor to {40000, 40000, 40000, kColorValueMaximum}
    set blackColor to {0, 0, 0, kColorValueMaximum}
    set darkGreyColor to {20000, 20000, 20000, kColorValueMaximum}

    -- From http://www.wilsonmar.com/lcolors.htm
    set myRed to (item 1 of myColor) / kColorValueMaximum
    set myGreen to (item 2 of myColor) / kColorValueMaximum
    set myBlue to (item 3 of myColor) / kColorValueMaximum
    set magicY to (0.3 * myRed) + (0.59 * myGreen) + (0.11 * myBlue)
    if (magicY < 0.5) then
        return {whiteColor, lightGreyColor}
    else
        return {blackColor, darkGreyColor}
    end if
end ContrastingTextColors

```

## ITerm2.app

**Full Colors:**

```

-- This nasty constant might as well be a global
global kColorValueMaximum
set kColorValueMaximum to 65535

-- Choose a random color for the background
set randomRed to (random number) * kColorValueMaximum
set randomGreen to (random number) * kColorValueMaximum
set randomBlue to (random number) * kColorValueMaximum
set myBackgroundColor to {randomRed, randomGreen, randomBlue}

-- Select appropriate text colors based on that background
set {myTextColor, myBoldColor} to my ContrastingTextColors(myBackgroundColor)

-- Now inflict them on the frontmost window
tell application "iTerm"
    set targetWindow to window 1
    set background color of current session of current terminal to myBackgroundColor
    set foreground color of current session of current terminal to myTextColor
end tell

on ContrastingTextColors(myColor)
    set whiteColor to {kColorValueMaximum, kColorValueMaximum, kColorValueMaximum, kColorValueMaximum}
    set lightGreyColor to {40000, 40000, 40000, kColorValueMaximum}
    set blackColor to {0, 0, 0, kColorValueMaximum}
    set darkGreyColor to {20000, 20000, 20000, kColorValueMaximum}
    -- From http://www.wilsonmar.com/1colors.htm
    set myRed to (item 1 of myColor) / kColorValueMaximum
    set myGreen to (item 2 of myColor) / kColorValueMaximum
    set myBlue to (item 3 of myColor) / kColorValueMaximum
    set magicY to (0.3 * myRed) + (0.59 * myGreen) + (0.11 * myBlue)
    if (magicY < 0.5) then
        return {"white", lightGreyColor}
    else
        return {"black", darkGreyColor}
    end if
end ContrastingTextColors

```

**Dark colors:**

```

-- This nasty constant might as well be a global
global kColorValueMaximum

-- set kColorValueMaximum to 65535
set kColorValueMaximum to 10000

-- Choose a random color for the background
set randomRed to (random number) * kColorValueMaximum
set randomGreen to (random number) * kColorValueMaximum
set randomBlue to (random number) * kColorValueMaximum
set myBackgroundColor to {randomRed, randomGreen, randomBlue}

-- Select appropriate text colors based on that background
set {myTextColor, myBoldColor} to my ContrastingTextColors(myBackgroundColor)

-- Now inflict them on the frontmost window
tell application "iTerm"
    set targetWindow to window 1
    set background color of current session of current terminal to myBackgroundColor
    set foreground color of current session of current terminal to myTextColor
end tell
on ContrastingTextColors(myColor)
    set whiteColor to {kColorValueMaximum, kColorValueMaximum, kColorValueMaximum, kColorValueMaximum}
    set lightGreyColor to {40000, 40000, 40000, kColorValueMaximum}
    set blackColor to {0, 0, 0, kColorValueMaximum}
    set darkGreyColor to {20000, 20000, 20000, kColorValueMaximum}
    -- From http://www.wilsonmar.com/1colors.htm
    set myRed to (item 1 of myColor) / kColorValueMaximum
    set myGreen to (item 2 of myColor) / kColorValueMaximum
    set myBlue to (item 3 of myColor) / kColorValueMaximum
    set magicY to (0.3 * myRed) + (0.59 * myGreen) + (0.11 * myBlue)
    --if (magicY < 0.5) then
        return {"white", lightGreyColor}
    --else
        return {"black", darkGreyColor}
    --end if
end ContrastingTextColors

```

## ITerm2.app (Version 2.9 and onwards)

Ref: <https://iterm2.com/applescript.html>

```

-- This nasty constant might as well be a global
global kColorValueMaximum
set kColorValueMaximum to 65535

-- Choose a random color for the background
set randomRed to (random number) * kColorValueMaximum
set randomGreen to (random number) * kColorValueMaximum
set randomBlue to (random number) * kColorValueMaximum
set myBackgroundColor to {randomRed, randomGreen, randomBlue}

-- Select appropriate text colors based on that background
set {myTextColor, myBoldColor} to my ContrastingTextColors(myBackgroundColor)

-- Now inflict them on the frontmost window
tell application "iTerm"
    tell current session of first window
        set background color to myBackgroundColor
        set foreground color to myTextColor
    end tell
end tell

on ContrastingTextColors(myColor)
    set whiteColor to {kColorValueMaximum, kColorValueMaximum, kColorValueMaximum, kColorValueMaximum}
    set lightGreyColor to {40000, 40000, 40000, kColorValueMaximum}
    set blackColor to {0, 0, 0, kColorValueMaximum}
    set darkGreyColor to {20000, 20000, 20000, kColorValueMaximum}
    -- From http://www.wilsonmar.com/lcolors.htm
    set myRed to (item 1 of myColor) / kColorValueMaximum
    set myGreen to (item 2 of myColor) / kColorValueMaximum
    set myBlue to (item 3 of myColor) / kColorValueMaximum
    set magicY to (0.3 * myRed) + (0.59 * myGreen) + (0.11 * myBlue)
    if (magicY < 0.5) then
        return {{65535,65535,65535 },lightGreyColor}
    else
        return {{0,0,0},darkGreyColor}
    end if
end ContrastingTextColors

```

Additional colors? Check out <http://noiseandheat.com/blog/2011/12/os-x-lion-terminal-colours/>