

# Zpravodaj Československého sdružení uživatelů TeXu

---

Marek Pomp

Názvy barev v METAPOSTu

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 23 (2013), No. 3-4, 127–132

Persistent URL: <http://dml.cz/dmlcz/150199>

## Terms of use:

© Československé sdružení uživatelů TeXu, 2013

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ*:  
*The Czech Digital Mathematics Library* <http://dml.cz>

**Abstrakt:** V článku je rozebráno použití barev v METAPOSTu a popsáno použití makra pro hromadnou deklaraci názvů barev v METAPOSTu.

**Klíčová slova:** METAPOST, názvy barev.

## Úvod

METAPOST může pracovat ve třech barevných modelech. Základní barevný model je RGB a je možné použít barvy modelů CMYK a greyscale. Pro barvy těchto modelů používá METAPOST běžný vektorový zápis. Pokud jde o pojmenované barvy, je základní nastavení METAPOSTu skoupé. V souboru `plain.mp` jsou pojmenované jen základní barvy modelu RGB, `black` – černá,  $(0, 0, 0)$ ; `white` – bílá,  $(1, 1, 1)$ ; `red` – červená,  $(1, 0, 0)$ ; `green` – zelená,  $(0, 1, 0)$  a `blue` – modrá,  $(0, 0, 1)$  a dále barva `background`, která je nastavena na hodnotu  $(1, 1, 1)$ .

Stephan Hennig sepsal makro `mpcolornames`, viz [1], které řeší hromadné pojmenování barev v návaznosti na makra  $\text{\LaTeX}$ u `xcolor.sty`, viz [3]. Nevýhodou makra `mpcolornames` je poměrně významná „ruční“ práce při přípravě nové množiny pojmenovaných barev a její dokumentace. Makro také nešikovně řeší problém duplicity jmen pro různé barvy v různých barevných modelech. Toto makro dovoluje míchat barvy různých modelů, což může na výstupním zařízení působit potíže. Navíc toto makro neřeší problém barvy `background`, viz níže.

## Barvy v METAPOSTu

Barvy modelů RGB a CMYK jsou v METAPOSTu standardně tvořeny uspořádanou trojicí, resp. čtveřicí čísel z intervalu  $\langle 0, 1 \rangle$ , hodnoty mimo tento interval jsou převedeny, před uložením na výstup, na bližší z čísel 0, 1. Barvy modelu greyscale reprezentují numerické hodnoty z intervalu  $\langle 0, 1 \rangle$  – 0 odpovídá černé a 1 bílé barvě – a hodnoty mimo tento interval METAPOST převede na bližší z čísel 0, 1.

Kreslení barvou příslušného barevného modelu aktivujeme jedním z přepínačů `withrgbcolor`, `withcmypcolor`, resp. `withgreycolor`, které kontrolují, zda použitá barva odpovídá barevnému modelu. Barvu lze také použít přepínačem `withcolor`, který je v závislosti na typu použité barvy interpretován jako `withmodelcolor`.

Hodnota	Model
1	žádný model
3	stupně šedi
5	RGB (default)
7	CMYK

Tabulka 1: Hodnoty proměnné `defaultcolormodel`.

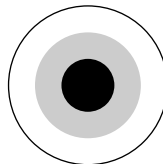
Příkazy `draw`, `fill` atd. bez přepínače `withcolor` použijí černou barvu v barevném modelu v závislosti na interní proměnné `defaultcolormodel`.

Použití METAPOSTu v příkazech typu `undraw`, `unfill` je jiné než v původním METAFONTu. V programu METAFONT příkazy typu `undraw`, `unfill` odebírají z navrstvených černých pixelů. Naproti tomu v METAPOSTu vybarví oblast barvou `background`. Tedy v `plain.mp` je `unfill` definováno jako:

```
def unfill expr p = fill p withcolor background enddef;
```

Pokud je `background` nastaveno jinak než na bílou barvu výstupního zařízení, zanechá příkaz `unfill` na papíru kresbu barvou `background`, zatímco skutečné pozadí, tj. místa, kde nebylo kresleno nic, je výstupním zařízením ponecháno bez tiskové stopy.

```
background:= (0,0,0); % černá
draw fullcircle scaled 2.1cm;
fill fullcircle scaled 1.4cm
    withcolor (0.8,0.8,0.8);
unfill fullcircle scaled .7cm;
```



Jestliže požadujeme práci s pojmenovanými barvami, je třeba nejprve deklarovat nové jméno, v závislosti na zamýšleném barevném modelu, příkazem `rgbcolor` (synonymem je `color`), resp. `cmkcolor`. Pro stupně šedi použijeme k deklaraci příkaz `numeric`. Např. v `plain.mp` je definováno

```
color white;
white := (1,1,1);
```

V METAPOST bohužel není interpretace barvy `background` svázána s proměnnou `defaultcolormodel`. V souboru `plain.mp` je `background` nastaveno na RGB hodnotu `(1,1,1)`, a `defaultcolormodel` tedy neovlivňuje chování příkazů `undraw`, resp. `unfill` apod. Pokud uživatel chce obrázek z hlediska použitého barevného modelu konzistentní, musí zároveň se změnou `defaultcolormodel` sám změnit barvu `background`. Aby nenastaly potíže na výstupním zařízení, je tedy nutné dodržet soulad mezi typem použitých barev, defaultní černou, kterou volí `defaultcolormodel` a nastavením barvy `background`, viz příklady 1 a 2.

```

defaultcolormodel:=3;          0 0 0 0 setcmykcolor
cmykcolor white;              newpath 0 0 moveto
white := (0,0,0,0);           1 0 lineto stroke
draw origin--right            0 setgray
withcolor white;              newpath 0 0 moveto
draw origin--right;           1 0 lineto stroke
undraw origin--right;         1 1 1 setrgbcolor
                              newpath 0 0 moveto
                              1 0 lineto stroke

```

Příklad 1: Kód a jeho výstup, kde ve volbě barevných modelů vládne chaos.

```

defaultcolormodel:=3;          1 setgray
numeric white, background;    newpath 0 0 moveto
white := 1;                   1 0 lineto stroke
background := white;          0 setgray
draw origin--right            newpath 0 0 moveto
withcolor white;              1 0 lineto stroke
draw origin--right;           1 setgray
undraw origin--right;         newpath 0 0 moveto
                              1 0 lineto stroke

```

Příklad 2: Kód a jeho výstup, sjednocený pro barevný model greyscale.

## Obtíž s hromadnou deklarací barev

Ruční hromadná deklarace názvů barev může být zdlouhavá, především pokud názvy barev obsahují čísla. Například názvy barev podle X11 obsahují barvy `maroon`, `maroon1`, `maroon2` atp. Protože v METAPOSTu název proměnné neobsahuje čísla, s výjimkou položek polí, musíme deklarovat proměnnou `maroon` i pole `maroon`,

```
color maroon, maroon[];
```

Není tedy možné jen jednoduše do deklarace zkopírovat názvy barev ze zvoleného seznamu. Navíc nutnost barvu nejprve deklarovat a potom teprve zavádět její hodnotu nutí uživatele při úpravách jakoukoliv změnu provádět na dvou místech. Pokud bereme v úvahu ještě výrobu případné dokumentace, resp. vizualizaci barev pomocí  $\TeX$ u, je možnost zavlečení chyby z nepozornosti obrovská a automatizace se určitě vyplácí.

## Použití makra `colornames.mp`

Makro `colornames.mp` a s ním související další soubory lze najít na adrese

<http://homel.vsb.cz/~pom68/dalsi/colornames/>

Makro `colornames.mp` řeší zavedení pojmenovaných barev do METAPOSTu. Předem jsou již připravené některé množiny pojmenovaných barev. Typické použití tohoto makra pro načtení názvů barev z předdefinované množiny (např. VGA) je

```
input colornames.mp;
VGA_names;
draw origin--right withcolor maroon;
```

Příkazy zavádějící předdefinované množiny pojmenovaných barev lze zobrazit na terminál a do `*.log` příkazem `show_palettes`, viz tabulka 2. Jména barev z právě načtené množiny se zobrazí příkazem `show_names`.

Příkaz	Množina barev
<code>VGA_names</code>	VGA
<code>SVG_names</code>	SVG 1.0
<code>DVIPS_names</code>	DVIPS
<code>XELEVEN_names</code>	X11
<code>SAFE_names</code>	bezpečné barvy, VisiBone Colorlab
<code>GRAY_names</code> resp. <code>GREY_names</code>	stupně šedi

Tabulka 2: Předpřipravené příkazy pro zavedení množin pojmenovaných barev

Při zavedení vlastní množiny pojmenovaných barev uživateli stačí pouze seznam názvů těchto barev a jejich numerické reprezentace v příslušném barevném modelu. Deklarace a příprava dokumentace probíhá zcela automaticky.

Stačí mít množinu pojmenovaných barev a jejich definice zapsanou v souboru `seznam_barev.mp`, jehož řádky jsou ve formátu `název_barvy := definice_barvy;`. Po načtení makra `colornames.mp` se vlastní seznam barev deklaruje a aktivuje jediným příkazem

```
makeColorNames(název_seznamu)(barevný_model)(soubor);
```

Například pro definici názvů barev z množiny VGA je v souboru

```
vga_declare_colornames.mp
```

zapsán seznam

```
maroon := (128/255,0,0);
red := (1,0,0);
...
gray := (128/255,128/255,128/255);
```

a uživatel jej může použít

```
input colornames.mp;
makeColorNames(VGA)(rgb)(vga_declare_colornames.mp);
draw origin--right withcolor maroon;
```

Makra se samozřejmě postarají o proměnnou `defaultcolormodel` a barvu `background`, tedy o soulad barevného modelu použitých barev, implicitní černé a bílé barvy pozadí.

## Vizualizace barev ze seznamu

Pro zobrazení barev z definovaného seznamu je zapotřebí vyrobit paletu, například sadu obdélníků vyplněných danou barvou a připravit jejich načtení pro  $\text{T}_{\text{E}}\text{X}$ . Příkaz

```
make_palette(název_seznamu)(adresář);
```

vygeneruje do zvoleného (existujícího) adresáře takovouto sadu obdélníků pro množinu barev deklarovanou příkazem `makeColorNames`. Každý obdélník je uložen jako jeden obrázek s názvem ve tvaru

```
rectangle-název_seznamu_ColorNames-název_barvy.mps
```

a seznam takto vygenerovaných obrázků je uložen do stejného adresáře pod jménem

```
název_seznamu_ColorNames.txt
```

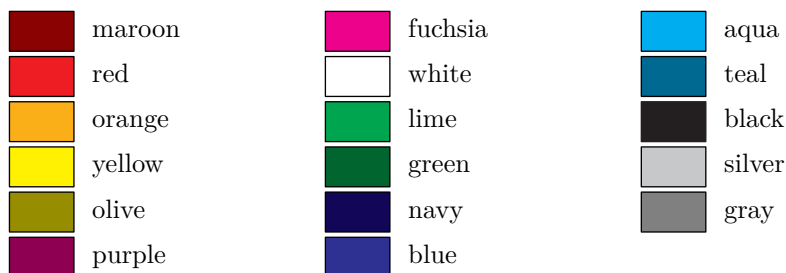
Pro  $\text{T}_{\text{E}}\text{X}$  jsou připravená makra `palette_colornames.tex`, která načítají vytvořené obrázky a vytisknou je ve třísloupcové sazbě, např.

```
\input palette_colornames.tex
\tableColor{VGA}{color_names_dir}
```

vytiskne vizualizaci seznamu barev (viz závěrečný obrázek), kterou jsme v  $\text{META-POSTu}$  připravili příkazy (`color_names_dir` zde znamená existující adresář):

```
input colornames.mps;
make_palette(VGA)(color_names_dir)
```

$\text{T}_{\text{E}}\text{X}$ ová makra `palette_colornames.tex` jsou připravena jak pro  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  (vyžaduje balík `multicol`), tak pro použití v `plainTEXu`, s podporou makrobalíčku `eplain.tex`.



Obrázek: Příklad vizualizace barev ze seznamu VGA

## Reference

- [1] HENNIG, S. The `mpcolornames` package. <http://www.ctan.org/tex-archive/graphics/metapost/contrib/macros/mpcolornames>, 2011.
- [2] HOBBY, J. D. METAPOST a user's manual. <http://www.tug.org/docs/metapost/mpman.pdf>, 2009.
- [3] KERN, U. Extending L<sup>A</sup>T<sub>E</sub>X's color facilities: the `xcolor` package. <http://www.ctan.org/tex-archive/macros/latex/contrib/xcolor>, 2007.
- [4] KNUTH, D. E. *The METAFontbook*. Addison Wesley, Reading, 1985.

## Summary: Named colors in METAPOST

This paper analyzes using color names in METAPOST and there is described how to use macro `colornames.mp` for mass declaration of color names in METAPOST.

**Key words:** METAPOST, color names.