Icon sampling form LittleJohn's VectorIconCollection.as logo ;-)

a PureBasic-Tool

Sourcecode, Colorlists and Documentation by
Charly Bauer (Omi, Oma), 2016-09-06, V0.50

The sourcecode contains code for the Colorpicker from PureBasic forum members:
wilbert, PureLust, netmaestro, infratec, idle, ?

This Document is created using **papyrus Autor 7** on Linux with Wine

Please excuse my rusty, bad english.

Inhaltsverzeichnis

1. Introduction

1.1 Short presentation

PB-ColorScout is a tool for PureBasic as source code.
It is running under

- Linux

- Mac

- Windows

It's the first time that i've applied LittleJohn's idea of Vector-Icons in practice.
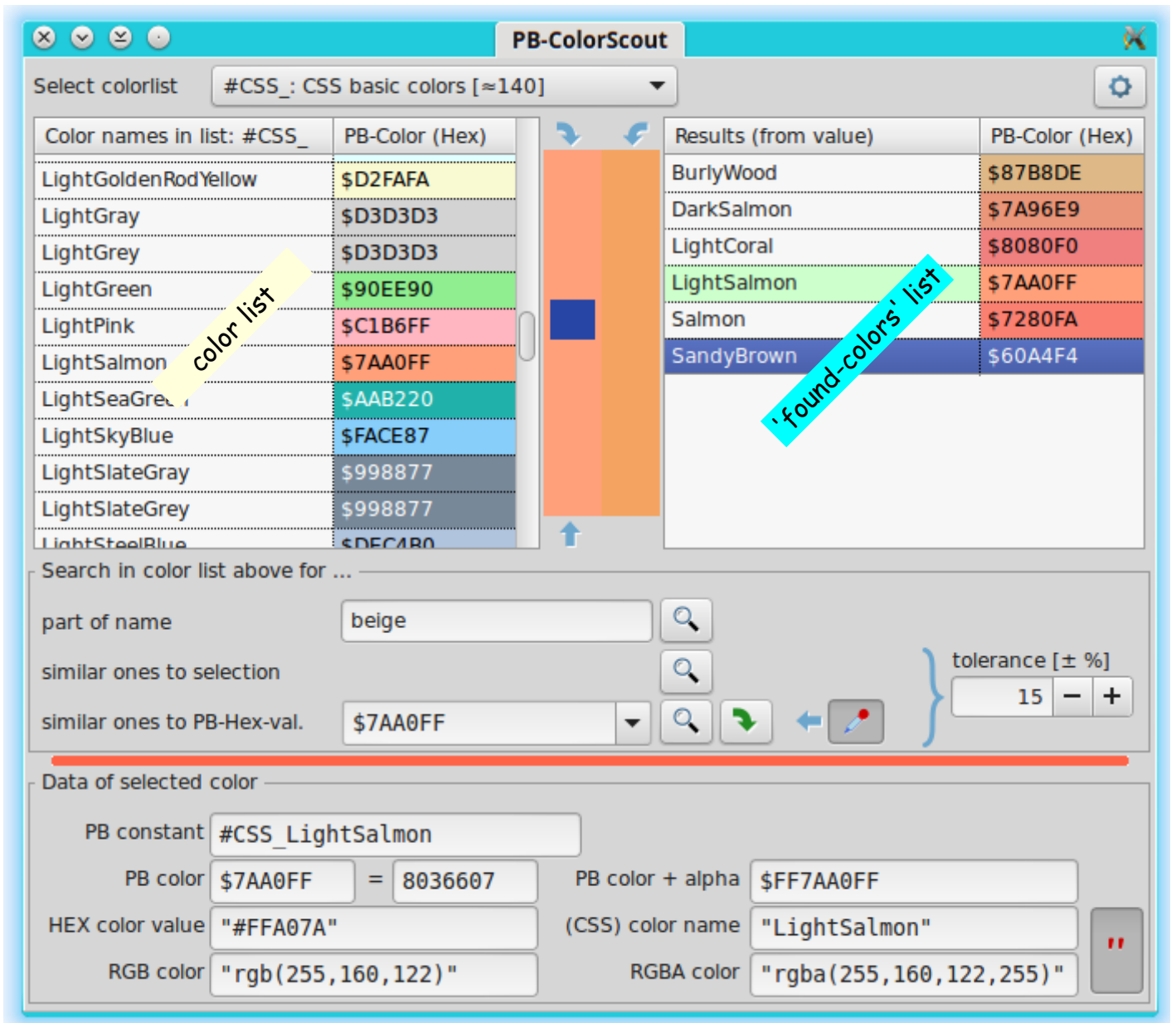
The main function is ...

- to reduce the mass of hardcoded trial&error-color-values in the code

- to find existing named colors from color lists for color values with adjustable tolerance.

- to find similar colors in other color lists with adjustable tolerance.

- to find color list with specific colors.

- to get an autoconverted PureBasic-Hex-Value from other (usual) color formats - and back.

- to integrate the PureBasic Color-Picker for all 3 platforms developed in the PB-Forum by
  wilbert, **PureLust**, **netmaestro**, **infratec**, **idle, ... ?**

- using one big file with known color lists (at the moment) for:
  » `CSS` (= HTML),
  » `X11` (= expanded CSS / Unix/Linux-set),
  » `RAL`, (w. english-, german-names & RAL-numbers)
  » `XKCD`,
  » `VGA`,
  » `LibreOffice`,
  » `Crayola` colors (different sets),
  » `Resene` colors set,
  » `Pantone-C and TPX` color sets,
  » `WIKI` (big collection from Web),
  » ...
  » or a combination of all included lists in the package.

- adding a complement, but no replacement for the integrated PureBasic 'Color Picker'-Tool!

1.2 Terms of use

Published program as sourcecode, file with colorlists and documentation are public domain.

- The files may not be sold in sourcecode or executable form!

- Can be used by anyone in any way!

- Can be changed, extended and/or improved in any way (especially OS-specific) and can be reposted in the PureBasic-Forum!

- Every form of improvement (code, API or translation of documentation), debugging or hints is welcomed.

PB-ColorScout

Select colorlist   #CSS_: CSS basic colors [≈140]

Color names in list: #CSS_	PB-Color (Hex)
LightGoldenRodYellow	$D2FAFA
LightGray	$D3D3D3
LightGrey	$D3D3D3
LightGreen	$90EE90
LightPink	$C1B6FF
LightSalmon	$7AA0FF
LightSeaGreen	$AAB220
LightSkyBlue	$FACE87
LightSlateGray	$998877
LightSlateGrey	$998877
LightSteelBlue	$DEC4B0

color list

Results (from value)	PB-Color (Hex)
BurlyWood	$87B8DE
DarkSalmon	$7A96E9
LightCoral	$8080F0
LightSalmon	$7AA0FF
Salmon	$7280FA
SandyBrown	$60A4F4

'found-colors' list

Search in color list above for ...

part of name   beige

similar ones to selection

similar ones to PB-Hex-val. $7AA0FF

tolerance [± %]   15 − +

Data of selected color

PB constant   #CSS_LightSalmon

PB color $7AA0FF   =   8036607   PB color + alpha $FF7AA0FF

HEX color value   "#FFA07A"   (CSS) color name   "LightSalmon"

RGB color   "rgb(255,160,122)"   RGBA color   "rgba(255,160,122,255)"

2.   Color lists

2.1   Content

The file for the color data, '*PB_NamedColorLists.pb*', is useable as PB-File to copy the colors or constants and uses UTF-8 coding. It is used as sample color list for the single color sets like CSS, X11, ...

Within **PB-ColorScout** all comments are used as comments, except the following lines:

```
;<IncludedNamespaces>
;</IncludedNamespaces>
```

All included lines define colors lists with the used prefix, like

```
;#CSS_=  CSS basic colors [≈140]
```

and must be contained as list in the file.

You can use a **double comment** in front to deactivate (unuse) the whole corresponding color list, like

```
;;#LOff_= LibreOffice4 colors [174]
```

For that you don't need to delete the whole color list from the sample list!

Each found color with the same (active) prefix is allocated to this list. I think that's all.

2.2 Format

The single color definition contain the following informations in the following order, separated by at least 2 spaces:

```
#CSS_AliceBlue = $FFF8F0;  $FFFFF8F0  "#F0F8FF"  "rgb(240,248,255)"  "rgba(240,248,255,255)"  "AliceBlue"
```

- #CSS_AliceBlue   = $FFF8F0;   `Colorconstant w. PB-Hex , w/o Alpha + comment sign`

- $FFFFF8F0   `Color w. Alpha as PB-Hex`

- "#F0F8FF"   `Color w/o Alpha as Hex-Value/triple, [DQuoted]`

- "rgb(240,248,255)"   `Color w/o Alpha as RGB()-Value, [DQuoted]`

- "rgba(240,248,255,255)"   `Color w. Alpha as RGBA()-Value, [DQuoted]`

- "AliceBlue"   `Pure (readable) name of the color [DQuoted]`
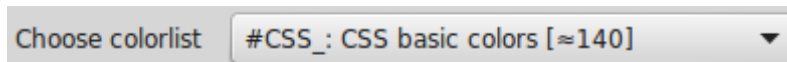
Adding a new color list to the file?

- Make a backup of your last well working color-file (*PB_NamedColorLists.pb*)

- Convert your new color list in a format as used in the file, see above.

- Choose a unique **Constant Prefix** for the new list and add the definition in the header between

  `;<IncludedNamespaces>` and `;</IncludedNamespaces>`

- Add the color list at an appropriate point.

2.3 Use

In **PB-ColorScout** each color-list found causes an entry in the ComboBoxGadget.
Info: The last entry is a auto-generated collection of all colors in all lists.
Each selection immediately loads the associated color list in the left **Color list (set)**-ListIconGadget.

Choose colorlist	#CSS_: CSS basic colors [≈140] ▼

#CSS_: CSS basic colors [≈140]
#X11_: X11 (ext. CSS colors) [≈550]
#VGA_: VGA basic colors [16]
#RALe_: RAL eng. names [213]
#RALg_: RAL ger. names [213]
#RALne_: RAL num. + eng. names [213]
#RALng_: RAL num. + ger. names [213]
#CryStd_: Crayola Standard [163]
#CryFlr_: Crayola Fluorescent [16]
#CrySSw_: Crayola Silver Swirls [24]
#CryMCt_: Crayola Multicultural [11]
#CrayMSc_: Crayola Magic Scent [30]
#CryGEM_: Crayola Gem Tones [16]
#CryPrl_: Crayola Pearl Brite [16]
#CryMtc_: Crayola Metallic FX [16]
#CryGel_: Crayola Gel FX [16]
#CrySlS_: Crayola Silly Scents [16]
#CryHnT_: Crayola Heads n Tails [16]
#Res_: Resene Colorlist [1383]
#LOff_: LibreOffice4 colors [174]
#XKCD_: xkcd color list [949]
#WIKI_: Wiki color collection [1298]
All: All [5717]

After a change of the color list, the program tries to find the previous selected color within the new list by name and value and selects this item.

3. Color list (set)

3.1 Colors in list

The **left** ListIconGadget, **color list** contains the names and PB-Hex-Values of the colors in the chosen set in the Color lists. The background of the 2. column shows the color of the item.

3.2 Selection

Changed selections are depict in

- Background of the left **Color compare panel** (TextGadget),
- Value-search-ComboBoxGadget as **PB-HEX-value**, see 5.3 by color value
- **Color data section** with all conversions, constant name and color name.

3.3 Aditional data

The whole data are available in the unfolded Color data section (see below) from a linked list.

3.4 Functions

- A **Doubleclick** on an item in the **color list** start the similarity search with current tolerance like the search **from selection in color list-**button below.
- A **Rightclick** (if a color is selected) opens the context menu (popup) with **export** functions for the current list or the selected item, see 3.5 Export.
- A **Rightclick** (if no color is selected) opens the context menu (popup) with **sort** functions for the current color list. An ascending sort by the names or a descending sort of values in the ListIconGadget is possible.
  → If that's difficult, e.g. on Windows, use the same function in the **Info & settings**-popup**.
  Info: While the program is running, no return to the original sorting is possible.
  Info: If the list was sorted previousely, the export is sorted too.
  Info: On Linux the system-search is activated (select one item, type in the colorname, scroll between the founds with cursor up/down)
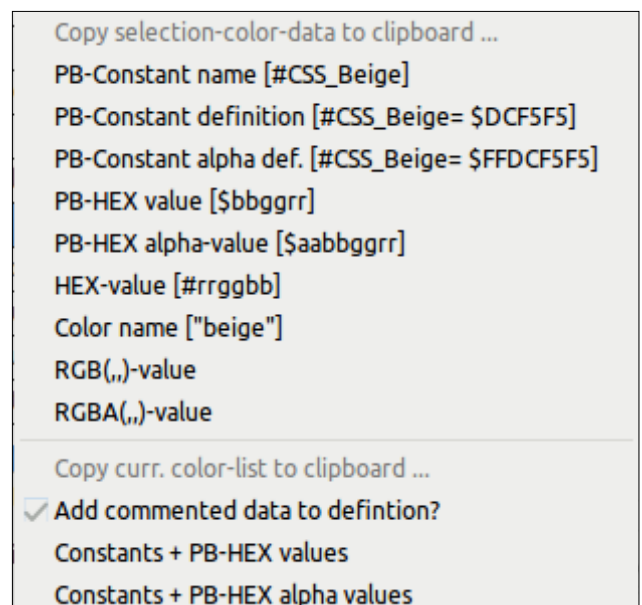 **Info:** It seems, that Windows selects items below the mousepointer on Rightclicks, whereas Linux maintains the previous selection.

3.5 Export

This Popup appears on Rightclick on an entry for the selection.

Export functions to the **clipboard** are integrated for

- Single selected color
- Current color list

popup in color list

3.5.1 Selected Color

I.e. the export for a color **Beige** from list **CSS**, with the constant name **#CSS_Beige** and the Hex triple **#F5F5DC** delivers the following export examples ...

Copy selection-color-data to clipboard (works for **color list** and '**found-colors**' list) ...

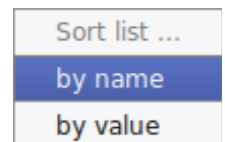|	
PB-Constant name [#CSS_Beige]	→ `#CSS_Beige`
PB-Constant definition [#CSS_Beige= $DCF5F5]	→ `#CSS_Beige= $DCF5F5`
PB-Constant alpha def. [#CSS_Beige= $FFDCF5F5]	→ `#CSS_Beige= $FFDCF5F5`
PB-HEX value [$bbggrr]	→ `$DCF5F5`
PB-HEX alpha-value [$aabbggrr]	→ `$FFDCF5F5`
HEX-value [#rrggbb]	→ `#F5F5DC	"#F5F5DC"`
Color name ["beige"]	→ `Beige	"Beige"`
RGB(,,)-value	→ `rgb(245,245,220)	"rgb(245,245,220)"`
RGBA(,,)-value	→ `rgba(245,245,220,255)	"rgba(245,245,220,255)"`

3.5.2 Current color list

Complete color list exports are possible from the **color list** popup ...

Add commented data to defintion?	→ `Attaches all color data as comment`
Constants + PB-HEX values	→ `Format: #CSS_Beige= $DCF5F5[; $FFDCF5F5 "...`
Constants + PB-HEX alpha values	→ `Format: #CSS_Beige= $FFDCF5F5[; $DCF5F5 "...`

3.6 Sorting

The **Color list (set)** can be temporarily sorted by a rightclick in the ListIconGadget, if no item is selected (this can be difficult with windows, alternative see Info & settings!)

Sort list ...
by name
by value

4.   Found colors (search result)

The **right** ListIconGadget **found colors (results)** is filled with the **'found-colors'** from the current **Color list (set)** by a search by name, search by similar value to selection a color-value. It also contains the names and PB-Hex-Values of the colors in the chosen color list.

4.1 Source & Identification

The headline contains a hint to the source of the data (name, selection, value) in the column 0.

4.2 Compare

A selection in the **Found colors (search result)**-list shows the color in the right **Color compare panel** to compare this color with the one in left panel (from selection in **Color list (set)** or the recent **value-search**) side by side.

4.3 Functions

The content in the **found colors-**list depends on the current **color list** and the setting of the **Tolerance**-SpinGadget. There's no merging function after changing the color list!

- A **Doubleclick** on a color causes here the **Found colors (search result)**-list to select the same color, if possible. **Info:** If the current list contains more than one matching color, the first one will be selected.
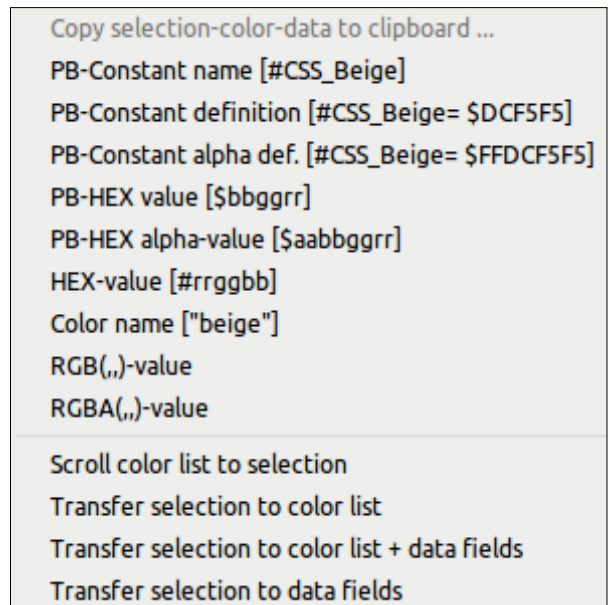
- A **Rightclick** opens the **context menu** (popup) with **export**, transfer to **data fields** and other funktions for the selected item.

4.4 Export

This Popup appears on Rightclick on an entry for the selection.

4.4.1 Selected Color

The Clipboard-Export for the selected color is the same as in the **3.5.1 Selected Color** for the current **Color list (set).**

Copy selection-color-data to clipboard ...
PB-Constant name [#CSS_Beige]
PB-Constant definition [#CSS_Beige= $DCF5F5]
PB-Constant alpha def. [#CSS_Beige= $FFDCF5F5]
PB-HEX value [$bbggrr]
PB-HEX alpha-value [$aabbggrr]
HEX-value [#rrggbb]
Color name ["beige"]
RGB(,,)-value
RGBA(,,)-value

Scroll color list to selection
Transfer selection to color list
Transfer selection to color list + data fields
Transfer selection to data fields

popup in 'found-colors'

4.4.2 Transfer functions

Transfer selection data from 'found-colors' list ...

```
Scroll color list to selection           → colorlist scrolls to same item as the selected 'found-color'
Transfer selection to color list         → 'found-color'-selection is transfered to 'color list'.
Transfer selection to col. list + datasection → " + data fields are filled from the selection (= doubleclick)
Transfer selection to Color data section → only datasection is filled from selection in 'found-colors'
```

5. Color search

Colors **within a tolerance** or **found color names** are listed in the right Found colors (search result)-ListIconGadget.
Info: A 'empty' search will cause a short blink of the ListIconGadget-Background!
Info: If a color matches exactly, the item has a light-green background.

5.1 by (part of) color name

part of name	wood	🔍

Even parts within the color names can be found.
The search can be started with the find icon at the right, but also with [Return] if the cursor is inside the entry field.

5.1.1 Import conversion

Pasted usual PB-constant-names in the entry field, like *#CSS_Beige*, are autoconverted to the colorname (like *Beige*).

5.2 from selection in color list

The search can be started with the find icon at the right, for similar values to the selection in the **Color list (set)** depending on the current tolerance to each RGB value (primary color) of the selected color.
The search can also be started by a Doubleclick on a color in the **color list**.

similar ones to selection	🔍

5.3 by color value

A **PB-Hex-value** can be entered (or be picked from the color picker).
The search can be started with the find icon at the right for values within the tolerance to each RGB value of the Hex-Value as set in the tolerance-SpinGadget.
The search can also be started with [Return] if the cursor is inside the entry field of the ComboBox.

similar ones to PB-Hex-val.	$7AA0FF	▼	🔍	↳	←	🖊

The entry field must be filled with a PB-HEX-value with **$ & 6 signs** (w/o alpha) like '*$FF00FF*' as '*$bbggrr*' (at the moment).

• The value is always set from the last selection in the **Color list (set)**. **(You can used it for a search from the last selected color in the list, including the original color!)**

• The last 10 entries are stored temporarily.

5.3.1 Import conversion

Pasted values to the ComboBox (like *natural numbers*,'*#rrggbb*', '*$aabbggrr*', *rgb(r,g,b)*, *rgba(r,g,b,a)*) are **auto-converted** to the PB-Hex-format (*$bbggrr*) w/o alpha when the assigned search-button is pressed.

5.4 Tolerance

The **Tolerance** is adjustable for similarity search from the

tolerance [± %]
15 − +

• **selected color** in the list, or the

• **PB-Hex-value**,

in percent from full value (=255). Colors with deviations of each of the **3 primary colors** up to the set level are listed.
The adjustable range goes from 1% to 50%.

A change of the tolerance value causes an **immediate update** of the '**found-colors**' list.
The program remembers the last searchmode (name, selection, value) and it will be applied at every change of tolerance again, see the "Results (from ...)" in the header in the '**found-colors**' list.

(On Linux, the SpinGadget is converted to the gtk3-API version to get a comfortable handling (mousewheel etc.) on each OS.)

6. Colorpicker

Since they exists and are very useful, i have built in the **colorpicker**-routines from PB-Forum by **wilbert**, **PureLust**, **netmaestro**, **infratec**, **idle**).
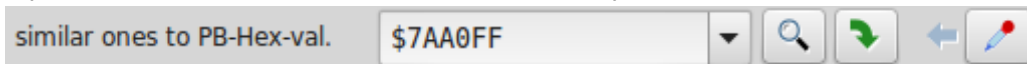
6.1 Use

To use it, the Icon ⬅️🖊️ has to be selected.

To pick a color within the current desktop, the **PB-ColorScout**-Window must stay focused. Pick the color under the mouse pointer with '**SPACEBAR**'. To get a better preview from this single pixel, use the Colorspy.

However the own window is taciturn 😜.

The picked color is converted to PB-HEX-Format und placed in the value search ...

similar ones to PB-Hex-val. $7AA0FF
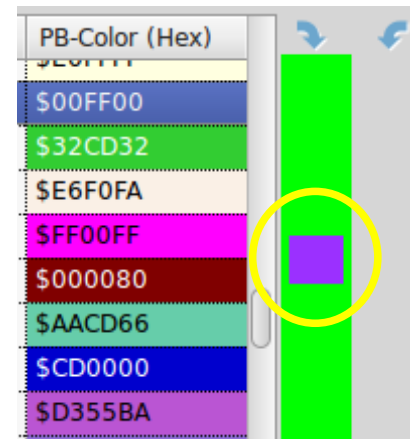
6.2 Options

If the popup-entry: Reset colorpicker button after each action is selected, the icon (and 'ON'-state) is reset after each pick-action!

6.3 Colorspy

There's a colorspy (zoom), which steadily shows the pixel-color under the mouse-hotspot within a box if the colorpicker is activated. The colorspy is placed within the **left** Color compare panel.

It's active if the colorpicker is active and the entry *'Color-Spy on' with colorpicker* in Info & settings is activated.

The color is updated in 100 ms-Timer-Intervals (defined in **#PeriodCSpy**-constant).

7.   Color compare panel

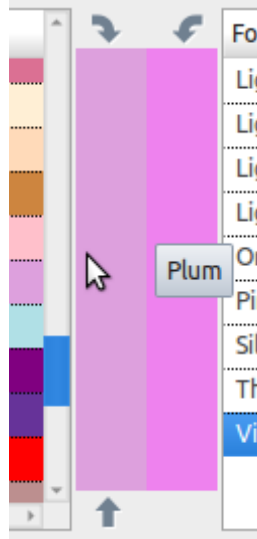To get a good impression of the difference and appearance of the compared colors,

- the selected color from the **Color list (set)** or from the **PB-Hex-Value** is faced against

- the selected color in the Found colors (search result)-list

in 2 bigger blocks. Therefore TextGadgets are used at the moment. (Since it maked problem an Linux-KDE (gradient) and PB5.50, it maybe will be converted to Image- or CanvasGadgets in the future.)

The **left** color panel is filled from ...

- a selection or doubleclick in left color list

- a search from selection in left color list

- a value search

The color name can be seen in the tooltip of the left color block (not after a value search)

The **right** color panel is filled from ...

- a selection or doubleclick in the right 'found-colors' list

The color name can be seen in the tooltip of the right color block.

8. Clipboard exports and transfers

Export and transfer functions are available on context menu (popup) within the ListIconGadgets for **color list** or **found colors**.

8.1   Color list: Single Color data: exports

see 3.5.1 Selected Color

8.2   Color list: List export

see 3.5.2 Current color list

8.3   Found color list: Single Color data: exports

see 4.4.1 Selected Color

8.4   Found color list: Transfer & convert to data block

see 4.4.2 Transfer functions

9. Color data section

The data section contains all other color informations in the color lists file.
The common color data formats e.g. for CSS are integrated.
The data in are copied in the data-fields (some fields with or w/o Doublequotes) and converted to other formats like the Hex-triple *#F5F5DC* to extract data from it for the use in PureBasic like API calls, CSS/HTML, ...
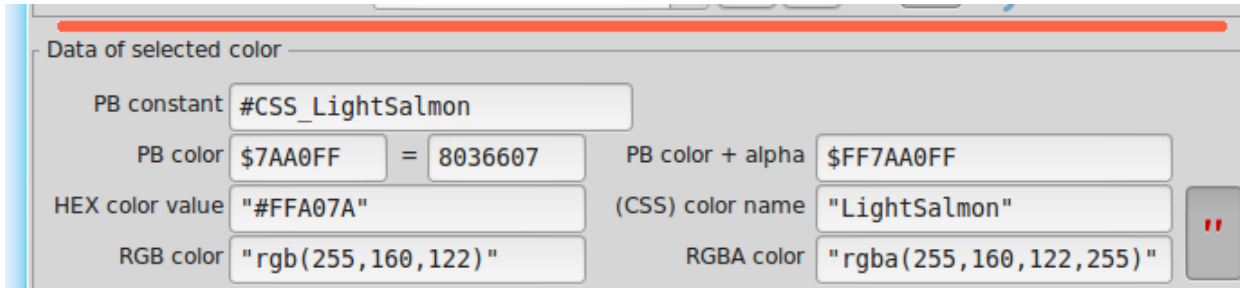
9.1 Visibility

You can fold / unfold the data section with a click on the horizontal bar above the container ...

9.2 Source

The data are transfered by ...

» color list : automatically set by a selection

» found-colors : set by a context menu entry on Rightclick

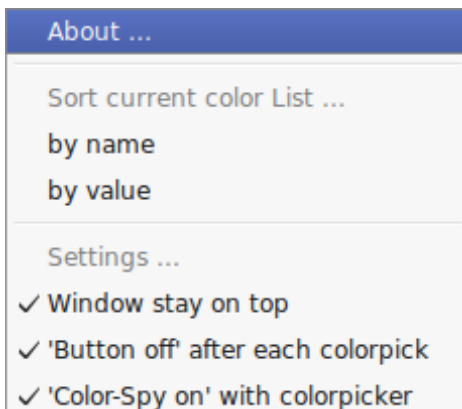» PB-Hex-value search : Data section is filled from this icon for the value in the search entry .

The last **data source** changes the headline of this section:

» "Data of selected color ..."

» "Data of selected found-color ..."

» "Data of PB Hex value ..." → in this case no *PB constant* name or *color names* is given and assigned!

The **Doublequote-**Button causes some fields being covered in Doubleqoutes as needed e.g. in CSS.

Maybe later this section will be extended or duplicated to a convert-tool for the different color formats.

10. Info & settings

contains a popup menu with

• the 'About' dialog.

• sorting for the current **Color list (set)** by name or value

• a checkbox for "**Window stay on top**" property.

• a checkbox for "**'Button off' after each colorpick**", which causes the ColorPicker-Button being reset to 'Off' after each action.

11. Preference

11.1 Information

The ini-File is loaded on Programstart and automatically save at programm end.
It's place in the program folder. If you wanna reset it: delete it!

11.2 Data

The following informations are included in the ini-file (setting at program end) ...

• Window coordinates

• "Stay on top" state

• State (fold/unfold) of Color data section

• Current Color list (set)

• Tolerance setting

• Colorspy ON with Colorpicker

• Doublequotes for transfer in Color data section

• Reset Colorpicker-Button after each action

• (Similarity search with Doubleclick on an item in Color list (set)) = Currently always ON.

12. Installation & operation

• Create a folder with any name (like "PBColorScout/") in your programming directory.

• Copy this 3 files in the folder:
  » PB-ColorScout.pb
  » PB_NamedColorLists.pb
  » PB-ColorScout.pdf

• Set in PureBasic 'Compiler options' / Compile/Run /'
  'Create temporary executable in the source directory'.

• Compile it with PureBasic 5.4x+ for your OS on your machine (Linux: default gtk3-subsystem), save it in your created folder (or your **PB-Tools**-folder and copy the colors-file: PB_NamedColorLists.pb into it) and use it.

• Available as **PureBasic-Sourcecode** for **V5.4x** to
  » modify it for your requirements.
  » compile it for your OS, 32/64 Bit (in Linux use PB5.4x with gtk3)

• There's a general ErrorHandler() integrated. If an error occurs repeatedly, please send a **pm** to user Omi / Oma in the PureBasic-Forum with the content of the error message. Thank you!