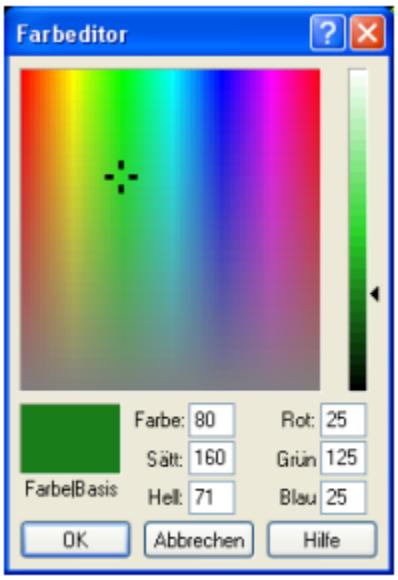
Den Farbwert (COLOR) erhält man wenn man mit dem Rekorder das ändern einer Farbe aufzeichnet!



Die rgb -Werte für OSDM-Makros rechnen

$$rgb\,Wert\,für\,Makro = \frac{Wert\,aus\,Farbeditor}{255}$$

Der max. Wert „255“ im Farbeditor entspricht folglich 1
Der min. Wert „0“ im Farbeditor entspricht folglich 0

Umrechnen von HEX-FArbcode:

http://www.drpeterjones.com/colorcalc/

http://www.farb-tabelle.de/de/rgb2hex.htm?q=178%2C229%2C255

Hier mal ein Beispiel, dass schematisch das setzen von Farben zeigt. Wenn auf den Knopf gedreuckt wird, wird das Eingabefled auf 'gruen' geupdated.

Code:

```
(sd-defdialog 'TESTE_FARBE
   :variables '(
     (FARBE
        ;weist der Variablen "FARBE" den Typ ":rgb-color" zu (bringt Farbeditor mit sich!)
        :value-type :rgb-color
        ;nach dem setzen der Variable wird deren Wert im Ausgabefenster angezeigt
        :after-input (display FARBE)

     );; Farbe

     (SETZE_GRUEN
        ;Aktion beim drücken auf "SETZE_GRUEN"
        :push-action (sd-set-variable-status 'FARBE :value (sd-rgb-to-color 0.09803,0.49019,0.09803))
     );; SETZE_GRUEN

    );; variables
);; sd-defdialog
```

Und hier noch eine kleine Erklaerung. In der Doku fuer :value-type findest du folgende Tabelle:
:value-type LISP Data Type
---------- ----------
:rgb-color rgb integer
Das bedeutet das der Wert eine Variable vom Typ :rgb-color eine Integer-Zahl ist. Im Funktionsindex findest du dann eine Funktion 'sd-rgb-to-color', die dir aus einem 3D-Vektor (oder GPNT3D, was genau deinen Daten entspricht) genau solch eine Integer-Zahl macht.
Ich hoffe, das hilft dir jetzt ein bisschen weiter. Ich denke mal, wenn du aus dem Programm-Code all die Konvertierungen aus einem String zu einem 3D Vektor erstmal raus nimmst, dann wird der Code schon ein wenig uebersichtlicher.

Abfragen (prüfen) von Flächenfarbe:

(== (sd-vec-equal-p (sd-inq-face-color face) 1.0,0.63,0.6) :resolution 1e-3)

Utilities zum Umwandeln der Farben:

SD-RGB-TO-HSL [function]

```
(sd-rgb-to-hsl rgb-color)
```

Description:
Returns a vector containing the hsl (hue,saturation,luminance) equivalent of the rgb (red,green,blue) input vector.

Parameters:
rgb-color {GPNT3D} - the rgb color vector to be converted

Return Value:
hsl-vector {GPNT3D} - the hsl equivalent color vector

SD-HSL-TO-RGB [function]

(sd-hsl-to-rgb *hsl-color*)

Description:
Returns a vector containing the rgb equivalent of the hsl input vector.

Parameters:
hsl-color {GPNT3D} - the hsl color vector to be converted

Return Value:
rgb-vector {GPNT3D} - the rgb equivalent color vector

SD-COLOR-TO-RGB [function]

(sd-color-to-rgb *color*)

Description:
Returns a vector containing the rgb (red, green, blue) equivalent of the color input (Starbase).

Parameters:
color {FIXNUM} - the color value (Starbase) to be converted

Return Value:
rgb-vector {GPNT3D} - the rgb equivalent color

SD-RGB-TO-COLOR [function]

(sd-rgb-to-color *rgb-value*)

Description:
Returns a color value (Starbase) equivalent to the passed rgb (red, green, blue) vector.

Parameters:
rgb-value {GPNT3D} - the rgb vector to be converted

Return Value:
color {FIXNUM} - the color value (Starbase)

Hier gibts weiter Infos zu RGB-Farben:

http://www.wackerart.de/rgbfarben.html