



```
[[ stack("image:{PAC}", [pos], "transparent", "line", 1)]]
```

Transparency: I don't know how to add transparency to the [R G B] matrix, nor in the image nor the XYPlot components. But for a single character, you can add transparency color in the native plot component as an hexadecimal, or with 4 decimals in the XYPlot component:

$$\alpha := \frac{120}{255} = 0.4706$$

Color with transparency as one hexa and decimal number:

```
d2h2(x) := | h(n) := substr("0123456789ABCDEF", trunc(n)+1, 1)
            | concat(h(x-mod(x,16)/16), h(mod(x,16)))
```

```
CLR := concat("#", d2h2(255.alpha), d2h2(r), d2h2(g), d2h2(b)) = "#78FF8C00"
```

```
clr := | d := b + 16^2.g + 16^4.r + 16^6.trunc(255.alpha) = 2030013440
        | d - 2^32.(2.d >= 2^32 - 1)
```

That's what rgb function do:

```
rgb(C, a) := | A := [ C_3 C_2 C_1 255.a ] Clear(k)
              | d := sum_{k=1}^4 max([ 0 min([ 255 trunc(A_k) ])]).16^{2.(k-1)}
              | eval(d - 2^32.(2.d >= 2^32 - 1))
```

```
rgb(r, g, b, a) := | rgb([ r g b ], a)
```

```
rgb(r, g, b, alpha) = 2030013440
```

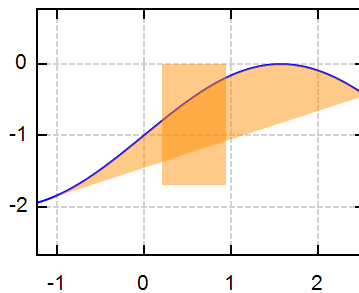
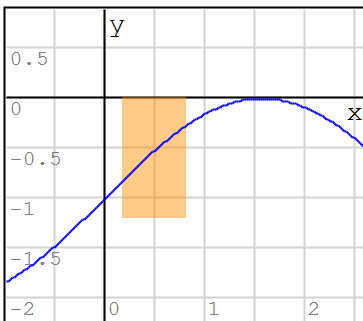
See also: [https://en.wikipedia.org/wiki/RGBA\\_color\\_model](https://en.wikipedia.org/wiki/RGBA_color_model)

Native SMath plot and XYPlot

```
XYPlot'Traces#0'FillStyle'FillColor := num2str(clr)
```

```
XYPlot'Traces#0'FillStyle'Filled := "True"
```

$$[255.alpha \ r \ g \ b] = [120 \ 255 \ 140 \ 0]$$



```
{ sin(x) - 1
  augment(0, 0, "#", 40, CLR)
```

```
{ sin(x) - 1
  augment(0, 0, "#", 40, clr)
```

Propiedades de Trace:

- ▼ Apariencia
  - ▼ FillStyle (...)
    - FillColor 120, 255, 140, 0
    - Filled True**
    - Hatched False
    - HatchStyle Percent20
  - IsY2Data False
  - > LineStyle (...)
  - Name
  - > SymbolStyle (...)

How can we use that for show text in XYPlot? I don't know. Maybe we can ask to Viacheslav to show a matrix with values between  $-2^{32}$  and  $+2^{32}$  as a colored one, instead a gray scale.

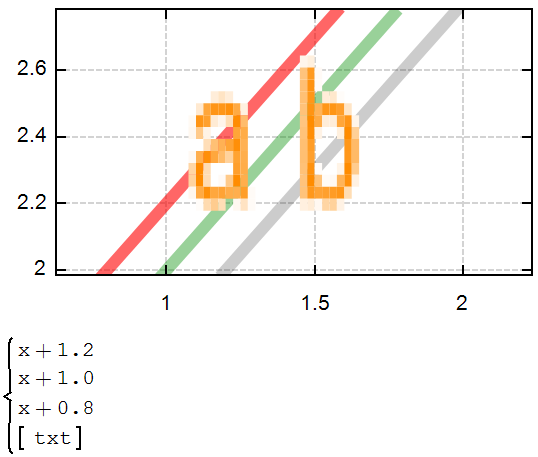
But we can draw the text by squares, avoiding the whitest colors

```
pos(X, r, c, k) := stack(X_1 + k/20.c, X_2 + k/14.r, k/20, k/14)
```

```

for r ∈ [1..rows(PA)]
  for c ∈ [1..cols(PA)]
    if PAr c < 250
      txtk := k + 1 := [
        "rect"
        pos(A, rows(PA) - r, c, k)
        ""
        ""
        1
        clr(PAr c)
      ]
    end if
  end for
end for

```



where ["rect", [a; b; w; h], ... ] draws a rectangle with diagonal from (a,b) to (a+w,b+h)

Alvaro `appVersion(4) = "1.73.9126.0"`