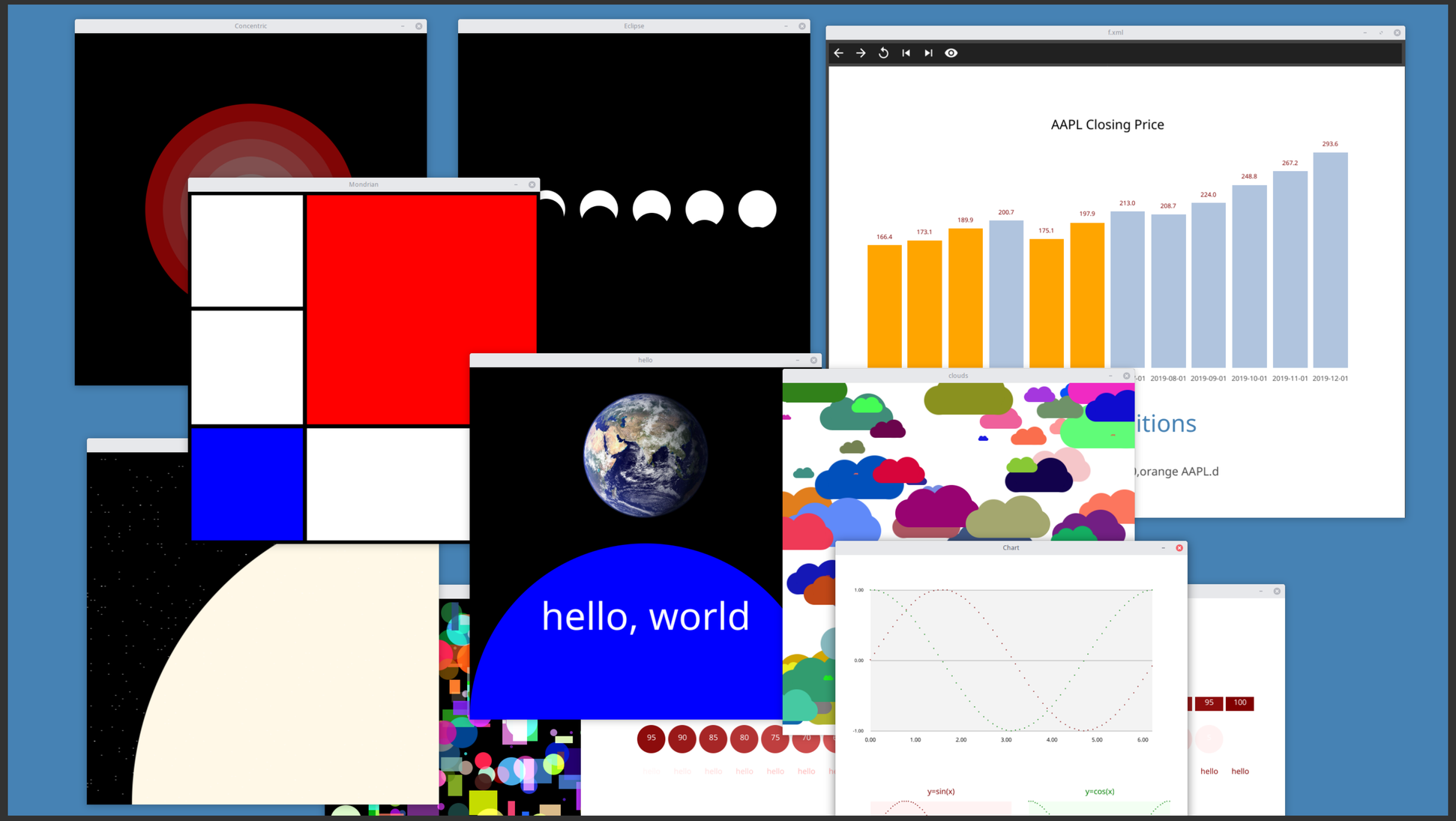
fc: a high-level canvas API for the fyne toolkit



Anthony Starks @ajstarks

Elements

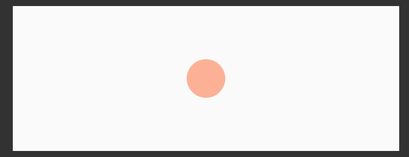
Text

CText

EText

circle

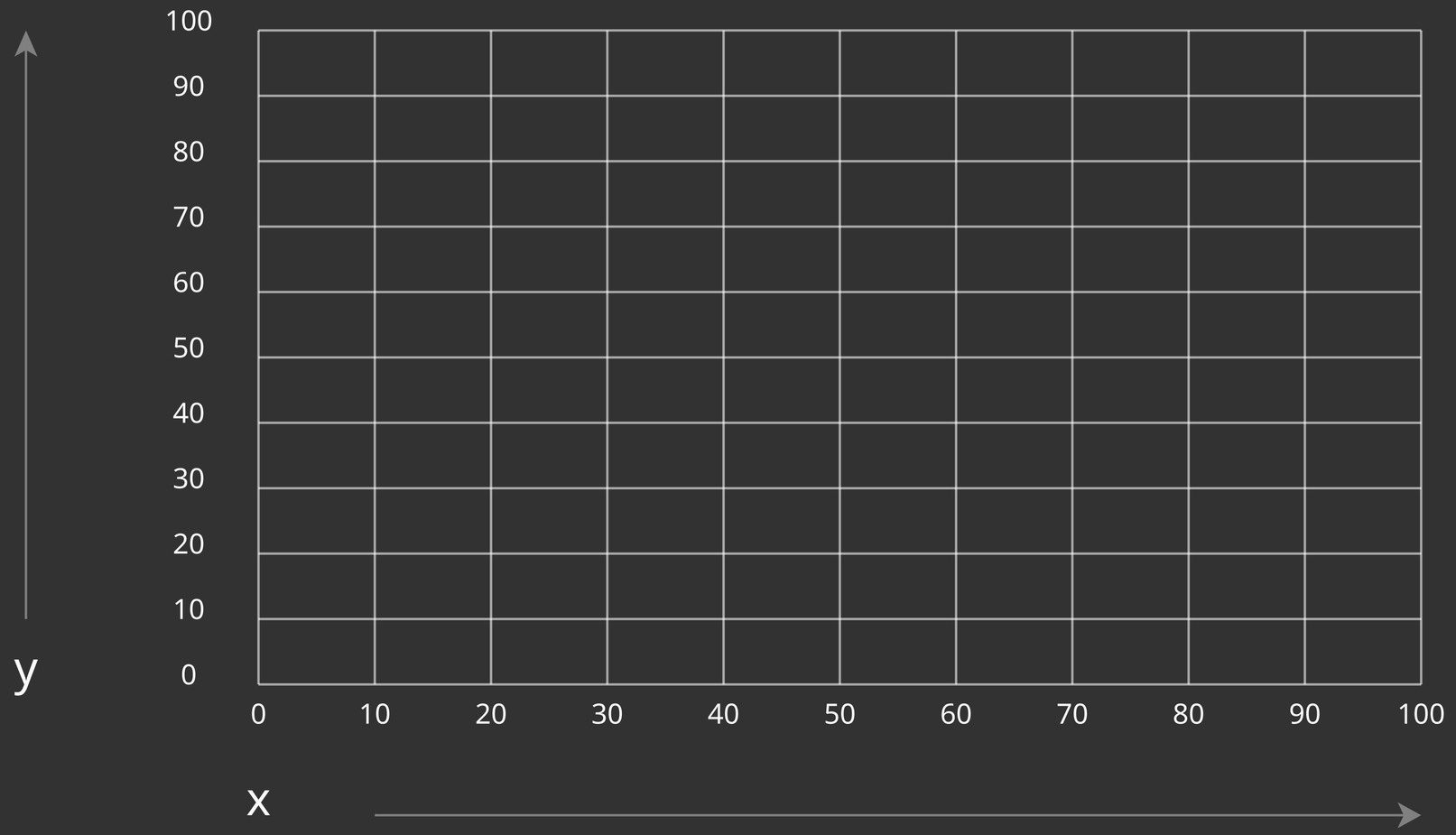
line



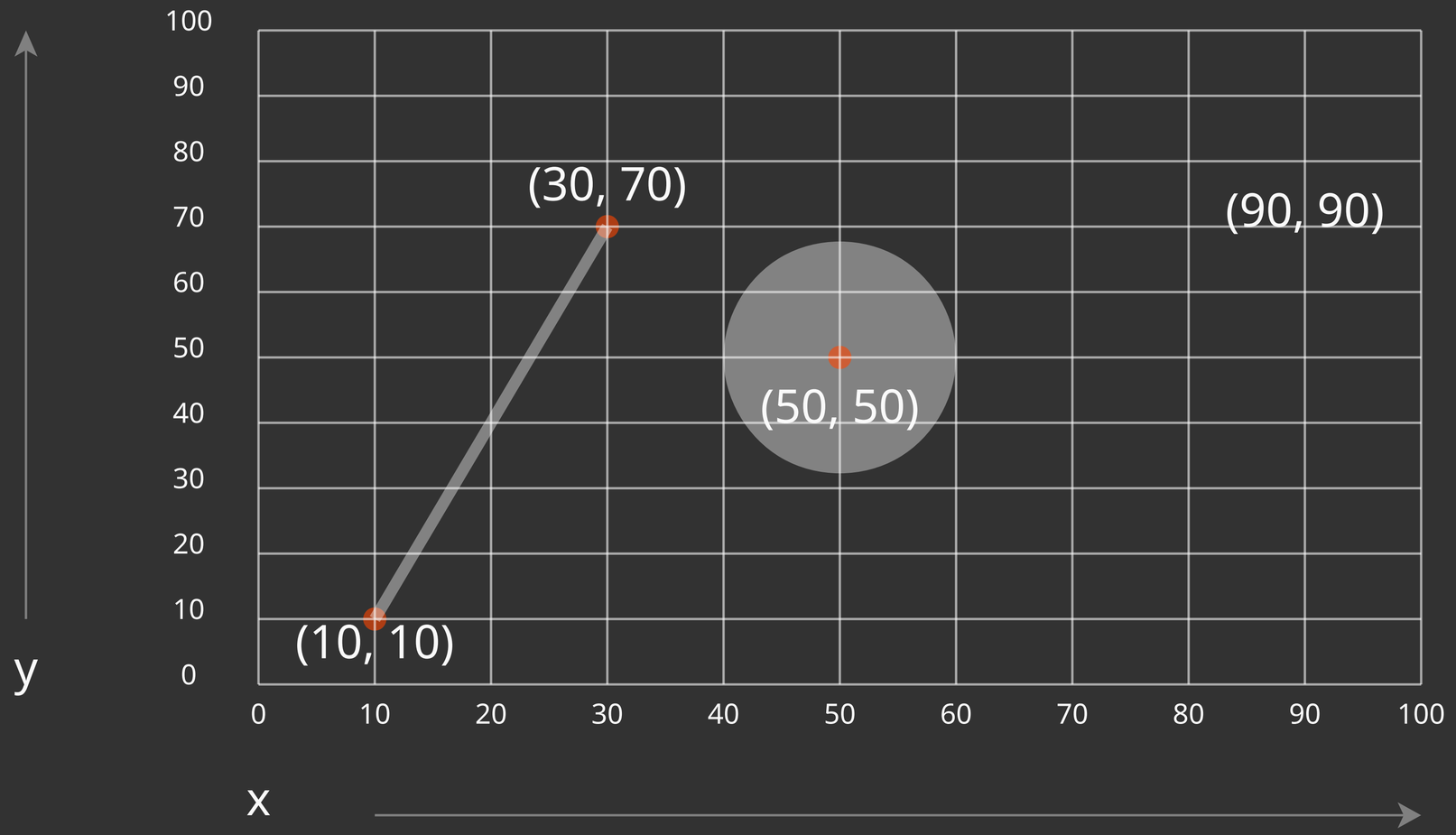
rectangle

image

Percentage-based Grid



Using the Percentage-based Grid



Rect(90, 80, ...)

Circle(50, 50, ...)

Line(10, 10, 30, 70, ...)

Percentage-based Methods

Make a new canvas	`NewCanvas(name string, w, h int) Canvas`
Place text, left-aligned	`Text(x, y, size float64, s string, fill color.RGBA)`
Place centered text	`CText(x, y, size float64, s string, fill color.RGBA)`
Place end-aligned text	`EText(x, y, size float64, s string, fill color.RGBA)`
Obtain the text width	`TextWidth(s string, size float64) float64`
Make a circle centered (x,y)	`Circle(x, y, r float64, fill color.RGBA)`
Rectangle with corner at (x,y)	`CornerRect(x, y, w, h float64, fill color.RGBA)`
Rectangle centered at (x,y)	`Rect(x, y, w, h float64, fill color.RGBA)`
Draw line from (x1,y) to (x2,y2)	`Line(x1, y1, x2, y2, size float64, stroke color.RGBA)`
Place an image centered at (x,y)	`Image(x, y float64, w, h int, name string)`
Display and run	`EndRun()`

Convenience methods

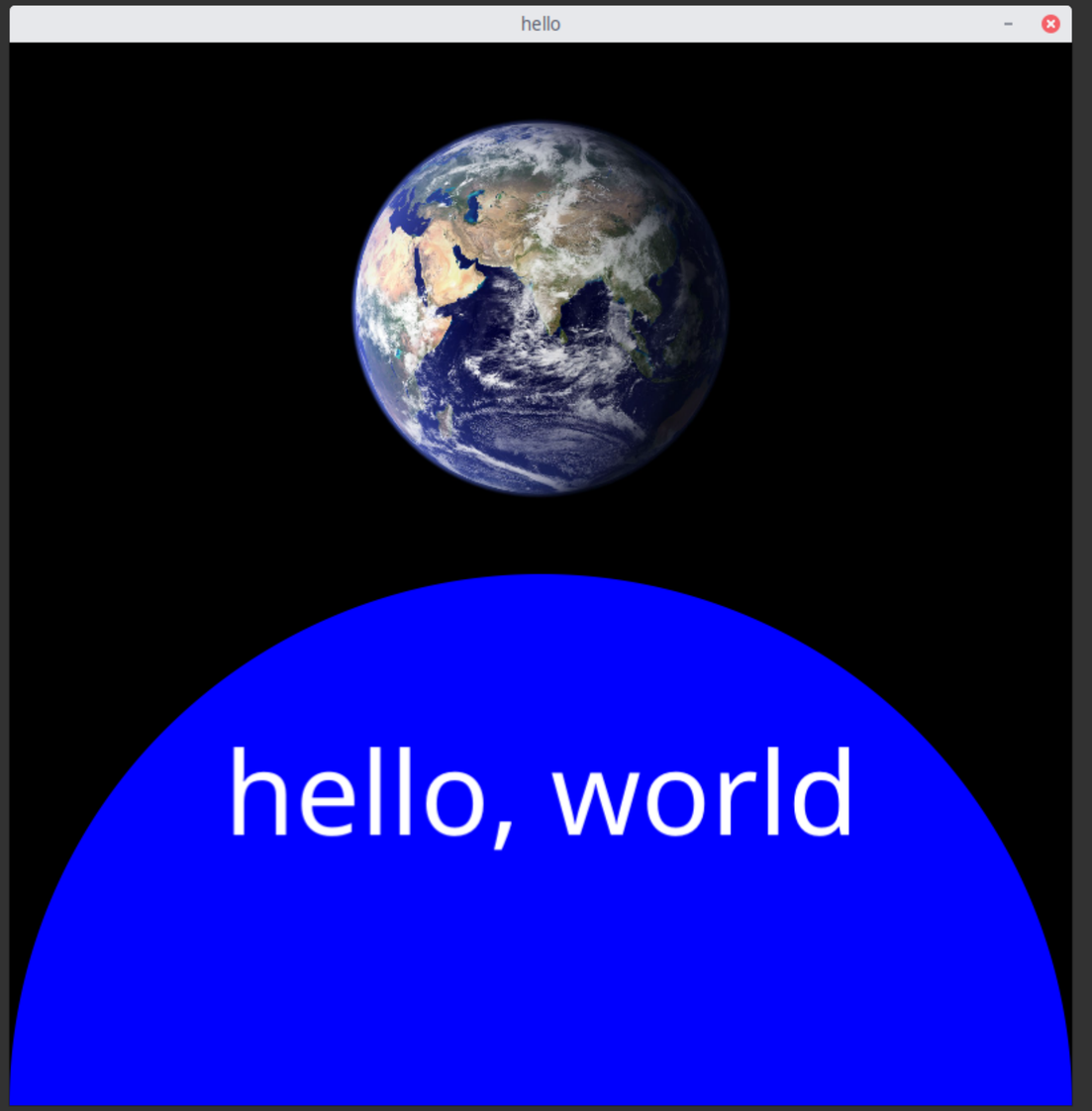
Lookup colors by name	`ColorLookup(s string) color.RGBA`
Map one range into another	`MapRange(value, low1, high1, low2, high2 float64) float64`
Polar to Cartesian coordinates	`Polar(x, y, r, angle float64) (float64, float64)`
Convert degrees to radians	`Radians(deg float64) float64`

Absolute Coordinate Methods

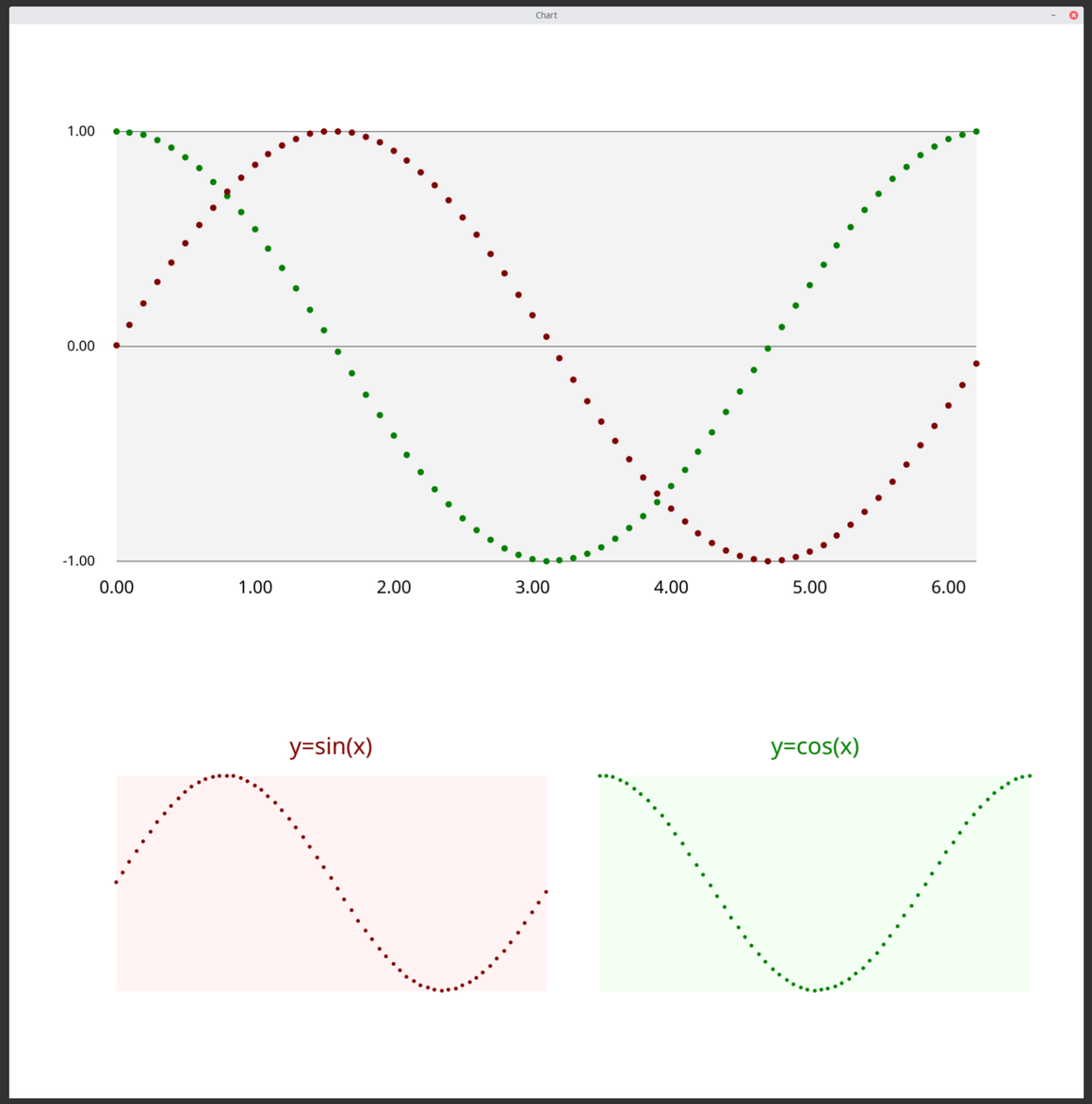
Place text, left-aligned	`AbsText(c *fyne.Container,x,y int,s string,size int,fill color.RGBA)`
Place centered text	`AbsTextMid(c *fyne.Container,x,y int,s string,size int,fill color.RGBA)`
Place end-aligned text	`AbsTextEnd(c *fyne.Container,x,y int,s string,size int,fill color.RGBA)`
Make a circle centered (x,y)	`AbsCircle(c *fyne.Container,x,y,r int,fill color.RGBA)`
Rectangle with corner at (x,y)	`AbsCornerRect(c *fyne.Container,x,y,w,h int,fill color.RGBA)`
Rectangle centered at (x,y)	`AbsRect(c *fyne.Container,x,y,w,h int,fill color.RGBA)`
Draw line from (x1,y) to (x2,y2)	`AbsLine(c *fyne.Container,x1,y1,x2,y2 int,size float32,stroke color.RGBA)`
Image upper-left corner at (x,y)	`AbsCornerImage(c *fyne.Container,x,y,w,h int,name string)`
Image centered at (x,y)	`AbsImage(c *fyne.Container,x,y,w,h int,name string)`
New app context	`AbsStart(name string,w,h int) (fyne.Window,*fyne.Container)`
Display and run	`AbsEndRun(window fyne.Window,c *fyne.Container,w,h int)`

hello, (fc) world

```
package main

import (
    "image/color"

    "github.com/ajstarks/fc"
)

func main() {
    width, height := 500, 500
    blue := color.RGBA{0, 0, 255, 255}
    white := color.RGBA{255, 255, 255, 255}

    canvas := fc.NewCanvas("hello", width, height)

    canvas.Circle(50, 0, 100, blue)
    canvas.CText(50, 25, 10, "hello, world", white)
    canvas.Image(50, 75, 200, 200, "earth.jpg")

    canvas.EndRun()
}
```



fc/chart:



fc/chart: data structures

```
// NameValue is a name,value pair
type NameValue struct {
    label string
    note  string
    value float64
}

// ChartBox holds the essential data for making a chart
type ChartBox struct {
    Title                   string
    Data                    []NameValue
    Color                   color.RGBA
    Top, Bottom, Left, Right float64
    Minvalue, Maxvalue      float64
    Zerobased               bool
}
```