# Go
# For Information Displays

Anthony Starks
@ajstarks

# Go is great in the back end

But sometimes it's about the picture

Information Displays

Libraries and Tools

Lots of pictures

Why Go?

Edward Tufte

Display data for precise, effective, quick analysis, Small multiples, Optimizing the data-ink ratio, Visual and beautiful evidence.

Nigel Holmes

Designing for the context: The data visualizer asks: “What’s the data?”, the infographics designer: “What’s the Story?”

An interesting arrangement of text and graphics designed to convey a message.

# I hate pushing pixels

programs are so much
better at it

Photo by deborahschillbach

**Kelsey Hightower** @kelseyhightower · Apr 10

Gophers will do anything to avoid writing Javascript.

> **Ben Johnson** @benbjohnson
> I wrote a full PhantomJS client in Go over the weekend. Still needs a higher level testing library though. github.com/benbjohnson/ph…

7 25 108

# SVGo

```
go get github.com/ajstarks/svgo/...
```

# OpenVG

```
go get github.com/ajstarks/openvg
```

# Deck

```
go get github.com/ajstarks/deck/
go get github.com/ajstarks/deck/generate
go get github.com/ajstarks/deck/cmd/pdfdeck/
go get github.com/ajstarks/deck/cmd/dchart/
```

# SVG

Element Arguments

Rect (100,200,250,125)

```
<rect x="100" y="200" width="250" height="125"/>
```

Element Arguments CSS Style

Rect (100,200,250,125,
"fill:gray;stroke:blue")

```
<rect x="100" y="200" width="250" height="125"
      style="fill:gray;stroke:blue"/>
```

Element Arguments Attributes

```
Rect (100,200,250,125,
      `id="box"`, `fill="gray"`, `stroke="blue"`)
```

```
<rect x="100" y="200" width="250" height="125"
     id="box" fill="gray" stroke="blue"/>
```

```go
package main

import (
    "github.com/ajstarks/svgo"
    "os"
)

func main() {
    canvas := svg.New(os.Stdout)
    width := 900
    height := 560
    style := "font-size:72pt;fill:white;text-anchor:middle"

    canvas.Start(width, height)
    canvas.Rect(0, 0, width, height)
    canvas.Circle(width/2, height, width/2, "fill:rgb(77, 117, 232)")
    canvas.Text(width/2, height*3/4, "hello, world", style)
    canvas.End()
}
```

Scale
fill:rgb(164,198,57)
Line
Roundrect
Circle
Arc
Line
Rect

```go
const defaultstyle = "fill:rgb(127,0,0)"

func circle(w http.ResponseWriter, req *http.Request) {
    w.Header().Set("Content-Type", "image/svg+xml")
    s := svg.New(w)
    s.Start(500, 500)
    s.Title("Circle")
    s.Circle(250, 250, 125, shapestyle(req.URL.Path))
    s.End()
}

func shapestyle(path string) string {
    i := strings.LastIndex(path, "/") + 1
    if i > 0 && len(path[i:]) > 0 {
        return "fill:" + path[i:]
    }
    return defaultstyle
}
```

# https://ajstarks.org:1958/{thing}/

clock

funnel

rotext

flower

rshape

cube

mondrian

lewitt

face

pacman

tux

concentric

Data

```
<thing top="100" left="100" sep="100">
    <item width="50"  height="50"  name="Little" color="blue">This is small</item>
    <item width="75"  height="100" name="Med"    color="green">This is medium</item>
    <item width="100" height="200" name="Big"    color="red">This is large</item>
</thing>
```

Picture

Little:This is small/blue

Med:This is medium/green

Big:This is large/red

```
<barchart title="2015 MacBook Benchmarks">
    <note>Source: AnandTech, April 14, 2015</note>
    <bdata scale="0,6000,1000"
        title="Mozilla Kraken 1.1 (native browser, milliseconds)">
       <bitem value="1729.7" name="Lenovo Yoga 3 Pro"/>
       <bitem value="1997.0" name="ASUS Zenbook UX305"/>
       <bitem color="steelblue" value="2589.0" name="12 inch MacBook"/>
       <bitem value="3621.7" name="Google Nexus 9"/>
       <bitem value="4014.3" name="Apple iPad Air 2"/>
       <bitem value="4296.7" name="NVIDIA Shield Tablet"/>
       <bitem value="5028.2" name="Apple iPad Air"/>
    </bdata>
</barchart>
```

# 2015 MacBook Benchmarks

*Source: AnandTech, April 14, 2015*

# f50 sunset

https://api.flickr.com/services/rest/
?method=flickr.photos.search
&api_key=...
&text=sunset
&per_page=50
&sort=interestingness-desc

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
   <photos page="1" pages="105615" perpage="50" total="5280747">
      <photo id="4671838925" ... secret="b070f3363e" server="4068" farm="5 ... />
      <photo id="3590142202" ... secret="c46752e4d8" server="2441" farm="3" .../>
      ...
 </photos>
</rsp>
```

```go
func vmap(value, low1, high1, low2, high2 float64) float64 {
    return low2 + (high2-low2)*(value-low1)/(high1-low1)
}

x := vmap(datax, 10, 20, 0, 767)
y := vmap(datay, 0, 10000, 0, 1023)
```

datax = 15, datay = 5000

# SVGo Clients

Doc Mgmt
Records
Retention
Metadata
Enterprise App
Portal
Customer Requests
Integrated Data
App Data
AP
Payments
A/P
Hub
Integration
Sales
DW
Reporting
Analytics
Pricing
Order to Cash
Sales
Cust/Prod
Master Data
EDI
Local Systems
Product Data
MDM
Customer
Product
List Price
EDI
Sales
Applications
Data and Integration
Highly Restricted
Restricted
Confidential
Public

# benchpng.txt

*image/png package: Go 1.3 vs Go 1.4*

```
structlayout -json bytes Reader | structlayout-svg -t "bytes.Reader" > reader.svg
```

# Go Programming Language Release History

## 4Q 2009

- 11/10 — Initial Release
- 12/09 — release 2009-12-09

## 1Q 2010

- 01/05 — release 2010-01-05
- 01/13 — release 2010-01-13
- 01/27 — release 2010-01-27
- 02/04 — release 2010-02-04
- 02/17 — release 2010-02-17
- 02/23 — release 2010-02-23
- 03/04 — release 2010-03-04

## 2Q 2010

- 04/13 — release 2010-04-13
- 04/27 — release 2010-04-27
- 05/04 — release 2010-05-04
- 05/27 — release 2010-05-27
- 06/09 — release 2010-06-09

## 3Q 2010

- 07/01 — release 2010-07-01
- 07/14 — release 2010-07-14
- 07/29 — release 2010-07-29
- 08/04 — release 2010-08-04
- 08/11 — release 2010-08-11
- 08/25 — release 2010-08-25
- 09/06 — release 2010-09-06

## 4Q 2010

- 10/13 — release 2010-10-13
- 10/20 — release 2010-10-20

# Sample Bullet Graph

The bullet graph features a single, primary measure (for example, current year-to-date revenue) compares that measure to one or more other measures to enrich its meaning, for example, compared to a target), and displays it in the context of qualitative ranges of performance, such as poor, satisfactory, and good.

Ability to Execute
Definite
Likely
Moderate
Unlikely
Thing 13
Item 16
Thing 11
Thing 1
Thing 9
Item 19
Item 18
Thing 12
Thing 10
Thing 8
Thing 7
Thing 2
Thing 21
Thing 14
Thing 6
Item 15
Item 20
Item 17
Thing 3
Thing 4
Thing 5
Weak
Moderate
Strong
Extereme
Strategic Alignment

# PLANNING TO PLAN

Year
0
1
2
Mobility
VoWiFi
HotSpots
3G Data
802.11n
NG VPN
Bluetooth
Messaging and Collaboration
Non Blackberry Wireless PDAs
Blogs
Wiki
Groove
RSS
Next-Gen Email
Extension of Presence/IM
Inter-Enterprise Secure IM
Productivity
Linux Desktop
MS Next Generation
Alternate Form PC
Desktop Video
Document Management
Infrastructure
Directory-Enabled Apps
Windows Rights Management
Toll Bypass
Portals
IPv6
Video Over IP
E&Y TLM Roadmap
Effort
Impact

Year 3
Year 2
Year 1
Messaging and Collaboration
Productivity
Mobility
Infrastructure
Next-Gen Email
Linux Desktop
Groove
Inter-Enterprise Secure IM
MS Next Generation
Desktop Video
Wiki
RSS
Alternate Form PC
Non Blackberry Wireless PDAs
Extension of Presence/IM
Blogs
Document Management
READY
Bluetooth
3G Data
Directory-Enabled Apps
NG VPN
Video Over IP
HotSpots
Toll Bypass
Windows Rights Management
802.11n
Portals
VoWiFi
IPv6

# Apple Purchases and Stock Price

# Twitter Update Frequency

You are here

# JONES
- LAYER TENNIS SEASON 4 WEEK 6 MATCH COMMENTARY BY MIKE MONTEIRO -
# MONTIEL

# DKNG
- LAYER TENNIS SEASON 4 WEEK 7 MATCH COMMENTARY BY JOHN GRUBER -
# DDL

# ANDERSON
- LAYER TENNIS SEASON 4 PLAYOFF QUARTERFINAL COMMENTARY BY BRYAN BEDELL -
# DKNG

CONTINO
CASSARO
A FIGHT TO THE DEATH!
THE BATTLE OF LONG ISLAND
DIVIDE AND CONQUER
DIVIDE AND CONQUER?
JON CONTINO SOMEHOW TURNS INTO AN EAGLE!!
CUT OFFS
PUTNAM
STOCKS
WHITE
TAYLOR
2022: A SPACE ODDITY
LANDING... 5 4 3 1
HALL
WARREN
The Great Autumn
Autumn
HARVEST
OR
DECK THE HALL THIS CHRISTMAS EVE
HALLO ON A SEC AREN'T WE GETTING A BIT HASTEVE?! IT'S ONLY OCTOBER
SORRY AM I THE ONLY ONE WITH A TIME MACHINE?
STRAWBERRYLUNA
DOUBLENAUT
ABOUT
REGULAR
SUPER SIZE ME!
VISIT THE "GREAT" WHITE NORTH
THE GREAT WHITE NORTH
THEY MAY EAT YOUR PENGUIN
WHAT THE PUCK?
CANADIANS HAVE BIGGER BALLS
JONES
MONTIEL
DKNG
DDL
SHAWNA X
STEVENS
TAYLOR
JONES
PENGUIN BURGER
PUTNAM
RAJKUMAR
bronze
whom!
it wasn't me
REYES
WHITE
She won't be missed for long.
ANDERSON
DKNG
TAYLOR
WHITE
CASSANDRA THUNDER
RAJKUMAR
ANDERSON
07:30
OINK
OOPS!
SNOOZE
ZOOM
ANDERSON
WHITE
LET'S GO CRAZY

Russell Westbrook
1471 out of 627
42.6%

Blake Griffin
1144 out of 574
50.2%

James Harden
1470 out of 647
44.0%

Stephen Curry
1341 out of 653
48.7%

```go
// The carpet pattern from "The Shining"
package main

import (
    "fmt"
    "os"

    "github.com/ajstarks/svgo"
)

func main() {
    xp := []int{50, 70, 70, 50, 30, 30} // x coords of the hexagon
    yp := []int{40, 50, 75, 85, 75, 50} // y coords of the hexagon
    xl := []int{0, 0, 50, 100, 100}     // x coords of the "house"
    yl := []int{100, 40, 10, 40, 100}   // y coords of the "house"

    stwidth := 12
    bgcolor := "rgb(227,78,25)" // background color
    bkcolor := "rgb(153,29,40)" // color of the hexagon
    stcolor := "rgb(65,52,44)"  // color of the stripes
    stylefmt := "stroke:%s;stroke-width:%d;fill:%s"

    width, height := 800, 800
    canvas := svg.New(os.Stdout)
    canvas.Start(width, height)

    // Define the units of the pattern
    canvas.Def()
    canvas.Gid("unit") // a polygon, within a "house" shape
    canvas.Polyline(xl, yl, "fill:none")
    canvas.Polygon(xp, yp)
    canvas.Gend()

    canvas.Gid("runit") // rotate and translate the unit shape
    canvas.TranslateRotate(150, 180, 180)
    canvas.Use(0, 0, "#unit")
    canvas.Gend()
    canvas.Gend()
    canvas.DefEnd()

    // Loops to define the pattern
    canvas.Rect(0, 0, width, height, "fill:"+bgcolor)
    canvas.Gstyle(fmt.Sprintf(stylefmt, stcolor, stwidth, bkcolor))
    for y := 0; y < height; y += 130 { // vertical
        for x := -50; x < width; x += 100 { // horizontal
            canvas.Use(x, y, "#unit")
            canvas.Use(x, y, "#runit")
        }
    }
    canvas.Gend()
    canvas.End()
}
```

```go
func main() {

    width := 200
    height := 200
    a := 1.0
    ai := 0.03
    ti := 15.0

    canvas := svg.New(os.Stdout)
    canvas.Start(width, height)
    canvas.Rect(0, 0, width, height)
    canvas.Gstyle("font-family:serif;font-size:100pt")

    for t := 0.0; t <= 360.0; t += ti {
        canvas.TranslateRotate(width/2, height/2, t)
        canvas.Text(0, 0, "i", canvas.RGBA(255, 255, 255, a))
        canvas.Gend()
        a -= ai
    }
    canvas.Gend()
    canvas.End()
}
```

ti = 15

ti = 30

ti = 45

# OpenVG

- Circle
- Ellipse
- Rectangle
- Rounded Rectangle
- Line
- Polyline
- Polygon
- Arc
- Quadratic Bezier
- Cubic Bezier
- Image

## OpenVG on the Raspberry Pi

# Why I use Go

Anthony Starks

@ajstarks
ajstarks@gmail.com

```
package main
import "fmt"

func main() {
    fmt.Println("hello, world")
}
```

This is the traditional hello, world program.
It's six lines long, and imports the fmt package,
leading some to be suprised at the size of the resulting binary.

```go
package main

import (
    "bufio"
    "github.com/ajstarks/openvg"
    "os"
)

func main() {
    width, height := openvg.Init()

    w2 := openvg.VGfloat(width / 2)
    h2 := openvg.VGfloat(height / 2)
    w := openvg.VGfloat(width)

    openvg.Start(width, height)                // Start the picture
    openvg.BackgroundColor("black")            // Black background
    openvg.FillRGB(44, 100, 232, 1)            // Big blue marble
    openvg.Circle(w2, 0, w)                    // The "world"
    openvg.FillColor("white")                  // White text
    openvg.TextMid(w2, h2, "hello, world", "serif", width/10) // Greetings
    openvg.End()                               // End the picture
    bufio.NewReader(os.Stdin).ReadBytes('\n') // Pause until [RETURN]
    openvg.Finish()                            // Graphics cleanup
}
```

# OpenVG Functions

| | |
|---|---|
| Circle | (x, y, r VGfloat) |
| Ellipse | (x, y, w, h VGfloat) |
| Rect | (x, y, w, h VGfloat) |
| Roundrect | (x, y, w, h, rw, rh VGfloat) |
| Line | (x1, y1, x2, y2 VGfloat) |
| Polyline | (x, y []VGfloat) |
| Polygon | (x, y []VGfloat) |
| Arc | (x, y, w, h, sa, aext VGfloat) |
| Qbezier | (sx, sy, cx, cy, ex, ey VGfloat) |
| Cbezier | (sx, sy, cx, cy, px, py, ex, ey VGfloat) |
| Image | (x, y VGfloat, w, h int, s string) |
| Text | (x, y VGfloat, s, font string, size int) |
| TextMid | (x, y VGfloat, s, font string, size int) |
| TextEnd | (x, y VGfloat, s, font string, size int) |

Mostly Cloudy

51°

Thursday April 20

8:16 am

Group Therapy and Chastened Lawmakers at Raucous Town Halls

Bold, Unpredictable Foreign Policy Lifts Trump, but Has Risks

Mississippi District Ordered to Desegregate Its Schools

Mitch McConnell and Paul Ryan Are of Two Minds About Divided Government

Trump Adviser's Visit to Moscow Got the F.B.I.'s Attention

Mostly Cloudy
47°
(feels like 45°)
Sunday March 13
3:03 am
Farhad and Mike Discuss the Apple Case and a Go-Playing Computer Program
Getting the Weather Report From Windows
The Government Answers Apple in the iPhone Case
Apple and U.S. Bitterly Turn Up Volume in iPhone Privacy Fight
F.C.C. Proposes Privacy Rules for Internet Providers
MECHANICAL DRAWING

Mostly Cloudy
Tuesday February 23
36°
9:41 pm
(feels like 29°)
29% Chance of precipitation
Justice Alito Addresses Prospect of an 8-Member Court
Michigan: State Senate Approves Aid to Help Flint Residents Pay Water Bills
Told to 'Take the Gloves Off,' John Kasich Says the Race Is Out of His Hands
Utah: Polygamist Sect Leaders Are Arrested on Fraud Charges
North Carolina: State Lawmakers Target Restroom Anti-Bias Law in Charlotte
HDMI Pi

Clear

66°

Friday April 14

5:55 pm

White House to Keep Its Visitor Logs Secret
Georgia Officers are Fired After Kicking Man at Traffic Stop
Does a Uniform Keep Officers in Line? The Baltimore Chief Thinks So
Wisconsin Man Who Mailed Manifesto to Trump Is Captured After 10-Day Manhunt
California Today: GoPros, Audiobooks and Other Perks of the Library

LG

```go
func main() {

    //...

    dateticker := time.NewTicker(1 * time.Minute)
    weatherticker := time.NewTicker(5 * time.Minute)
    headticker := time.NewTicker(10 * time.Minute)
    sigint := make(chan os.Signal, 1)
    signal.Notify(sigint, os.Interrupt)

    for {
        select {
        case <-weatherticker.C:
            canvas.weather(*location)
        case <-dateticker.C:
            canvas.clock(*smartcolor)
        case <-headticker.C:
            canvas.headlines(*section, *thumb)
        case <-sigint:
            openvg.Finish()
            os.Exit(0)
        }
    }
}
```

```go
func main() {

    //...

    dateticker := time.NewTicker(1 * time.Minute)
    weatherticker := time.NewTicker(5 * time.Minute)
    headticker := time.NewTicker(10 * time.Minute)
    sigint := make(chan os.Signal, 1)
    signal.Notify(sigint, os.Interrupt)

    for {
        select {
        case <-weatherticker.C:
            canvas.weather(*location)
        case <-dateticker.C:
            canvas.clock(*smartcolor)
        case <-headticker.C:
            canvas.headlines(*section, *thumb)
        case <-sigint:
            openvg.Finish()
            os.Exit(0)
        }
    }
}
```

```go
func main() {

    //...

    dateticker := time.NewTicker(1 * time.Minute)
    weatherticker := time.NewTicker(5 * time.Minute)
    headticker := time.NewTicker(10 * time.Minute)
    sigint := make(chan os.Signal, 1)
    signal.Notify(sigint, os.Interrupt)

    for {
        select {
        case <-weatherticker.C:
            canvas.weather(*location)
        case <-dateticker.C:
            canvas.clock(*smartcolor)
        case <-headticker.C:
            canvas.headlines(*section, *thumb)
        case <-sigint:
            openvg.Finish()
            os.Exit(0)
        }
    }
}
```

```go
func main() {

    //...

    dateticker := time.NewTicker(1 * time.Minute)
    weatherticker := time.NewTicker(5 * time.Minute)
    headticker := time.NewTicker(10 * time.Minute)
    sigint := make(chan os.Signal, 1)
    signal.Notify(sigint, os.Interrupt)

    for {
        select {
        case <-weatherticker.C:
            canvas.weather(*location)
        case <-dateticker.C:
            canvas.clock(*smartcolor)
        case <-headticker.C:
            canvas.headlines(*section, *thumb)
        case <-sigint:
            openvg.Finish()
            os.Exit(0)
        }
    }
}
```

# Deck

a Go package for presentations

| | |
|---|---|
| Start the deck | `<deck>` |
| Set the canvas size | `  <canvas width="1024" height="768" />` |
| Begin a slide | `  <slide bg="white" fg="black">` |
| Place an image | `  <image xp="70" yp="60" width="640" height="480" name="follow.png" sp="1" caption="Dreams"/>` |
| Draw some text | `  <text xp="20" yp="80" sp="5" link="http://goo.gl/Wm05Ex">Deck elements</text>` |
| Make a bullet list | `  <list xp="20" yp="70" sp="3" type="bullet">` |
| | `  <li>text, list, image</li>` |
| | `  <li>line, rect, ellipse</li>` |
| | `  <li>arc, curve, polygon</li>` |
| End the list | `  </list>` |
| Draw a line | `  <line xp1="20" yp1="10" xp2="30" yp2="10" />` |
| Draw a rectangle | `  <rect xp="35" yp="10" wp="4" hr="75" color="rgb(127,0,0)" />` |
| Draw an ellipse | `  <ellipse xp="45" yp="10" wp="4" hr="75" color="rgb(0,127,0)" />` |
| Draw an arc | `  <arc xp="55" yp="10" wp="4" hp="3" a1="0" a2="180" color="rgb(0,0,127)" />` |
| Draw a quadratic bezier | `  <curve xp1="60" yp1="10" xp2="75" yp2="20" xp3="70" yp3="10" />` |
| Draw a polygon | `  <polygon xc="75 75 80" yc="8 12 10" color="rgb(0,0,127)" />` |
| End the slide | `  </slide>` |
| End of the deck | `</deck>` |

# Anatomy of a Deck

# Deck elements

- text, list, image
- line, rect, ellipse
- arc, curve, polygon

Dreams

# Percent Grid

10%, 50%

Hello

50%, 50%

90%, 50%

# Percentage-based layout

Landscape

Portrait

# Scaling the canvas

Process
deck code
interactive
PDF
SVG

# deck/generate text and list functions

| | |
|---|---|
| Text | (x, y float64, s, font string, size float64, color string, opacity ...float64) |
| TextBlock | (x, y float64, s, font string, size, margin float64, color string, opacity ...float64) |
| TextMid | (x, y float64, s, font string, size float64, color string, opacity ...float64) |
| TextEnd | (x, y float64, s, font string, size float64, color string, opacity ...float64) |
| Code | (x, y float64, s string, size, margin float64, color string, opacity ...float64) |
| List | (x, y, size float64, items []string, ltype, font, color string) |

# deck/generate graphic functions

| | |
|---|---|
| Image | (x, y float64, w, h int, name string) |
| Arc | (x, y, w, h, size, a1, a2 float64, color string, opacity ...float64) |
| Circle | (x, y, w float64, color string, opacity ...float64) |
| Ellipse | (x, y, w, h float64, color string, opacity ...float64) |
| Square | (x, y, w float64, color string, opacity ...float64) |
| Rect | (x, y, w, h float64, color string, opacity ...float64) |
| Curve | (x1, y1, x2, y2, x3, y3, size float64, color string, opacity ...float64) |
| Line | (x1, y1, x2, y2, size float64, color string, opacity ...float64) |
| Polygon | (x, y []float64, color string, opacity ...float64) |

```go
func main() {
    deck := generate.NewSlides(os.Stdout, 1600, 900) // 16x9 deck to stdout
    deck.StartDeck()                                  // start the deck

    deck.StartSlide("rgb(180,180,180)")
    // ...
    deck.EndSlide()

    deck.StartSlide()
    // ...
    deck.EndSlide()

    deck.StartSlide("black", "white")
    // ...
    deck.EndSlide()

    deck.EndDeck()                                    // end the deck
}
```

```
// Text alignment
deck.StartSlide("rgb(180,180,180)")
deck.Text(50, 80, "Left", "sans", 10, "black")
deck.TextMid(50, 50, "Center", "sans", 10, "gray")
deck.TextEnd(50, 20, "Right", "sans", 10, "white")
deck.Line(50, 100, 50, 0, 0.2, "black", 20)
deck.EndSlide()
```

```go
// List
items := []string{"First", "Second", "Third",
                  "Fourth", "Fifth"}
deck.StartSlide()
deck.Text(10, 90, "Important Items", "sans", 5, "")
deck.List(10, 70, 4, items, "bullet", "sans", "red")
deck.EndSlide()
```

Important Items

- First
- Second
- Third
- Fourth
- Fifth

```
// Picture with text annotation
quote := "Yours is some tepid, off-brand, generic ‘cola’. " +
         "What I’m making is “Classic Coke”"
person := "Heisenberg"
deck.StartSlide("black", "white")
deck.Image(50, 50, 1440, 900, "classic-coke.png")
deck.TextBlock(10, 80, quote, "sans", 2.5, 30, "")
deck.Text(65, 15, person, "sans", 1.2, "")
deck.EndSlide()
```

```
# Designing for People
title    A View of User Experience: Designing for People    Anthony Starks / ajstarks@gmail.com / @ajstarks
section    Design    gray    white
caption    eames.png    Ray Eames    What works good is better than what looks good, because what works good lasts.
```

```
capgen slides.txt | pdfdeck ... > slides.pdf
```

# Deck Web API

```
sex -dir [start dir] -listen [address:port] -maxupload [bytes]
```

| | | |
|---|---|---|
| GET | / | List the API |
| GET | /deck/ | List the content on the server |
| GET | /deck/?filter=[type] | List content filtered by deck, image, video |
| POST | /deck/content.xml?cmd=1s | Play a deck with the specified duration |
| POST | /deck/content.xml?cmd=stop | Stop playing a deck |
| POST | /deck/content.xml?slide=[num] | Play deck starting at a slide number |
| DELETE | /deck/content.xml | Remove content |
| POST | /upload/ Deck:content.xml | Upload content |
| POST | /table/ Deck:content.txt | Generate a table from a tab-separated list |
| POST | /table/?textsize=[size] | Specify the text size of the table |
| POST | /media/ Media:content.mov | Play the specified video |

René Magritte
The False Mirror
LG

Henri Matisse

Circus (Le Cirque) from Jazz

# Design Examples

Top

Left

Right

Bottom

10%

30%

70%

10%

Header (top 10%)

Summary (30%)

Detail (70%)

Footer (bottom 10%)

# My Story

Section

One

Two

Three

Four

Five

Six

A. Person, April 2017

# Document Links

Add web and mailto links with the link attribute of the text element.

Once rendered as a PDF, clicking on the link opens the default browser or email client.

BOS

SFO

Virgin America 351

Gate B38

8:35am

On Time

JFK

✈

IND

US Airways 1207

Gate C31C

5:35pm

Delayed

# Flight Information

Los Angeles (LAX) New York/Newark (EWR)

| | |
|---|---|
| Distance Traveled | 1,958 mi |
| | 3,151 km |
| Distance to Destination | 596 mi |
| | 798 km |

| | |
|---|---|
| Time to Destination | 1:20 |
| Estimated time of arrival | 12:14 am |
| Local time of arrival | 12:14 am |

Ground speed

547 mph

382 kph

Headwind

50 mph

80 kph

Outside Temperature

-30° F

-34.4 C

Current Altitude

25,000 ft

7620 m

| | | | |
|---|---|---|---|
| AAPL | 179.98 | 3.04 | (1.72%) |
| AMZN | 1578.89 | 27.03 | (1.74%) |
| FB | 185.23 | 2.89 | (1.58%) |
| GOOGL | 1160.84 | 31.46 | (2.79%) |
| MSFT | 96.54 | 2.11 | (2.23%) |

2018-03-10 13:08:31

| | | | |
|---|---|---|---|
| AAPL | 181.72 | 1.74 | (0.97%) |
| AMZN | 1598.39 | 19.50 | (1.24%) |
| FB | 184.76 | -0.47 | (-0.25%) |
| GOOGL | 1165.93 | 5.09 | (0.44%) |
| MSFT | 96.77 | 0.23 | (0.24%) |

2018-03-12 21:05:06

| | | |
|---|---|---|
| AAPL | 175.30 | -2.72 (-1.53%) |
| AMZN | 1544.93 | -26.75 (-1.70%) |
| FB | 172.56 | -12.53 (-6.77%) |
| GOOGL | 1100.07 | -34.35 (-3.03%) |
| MSFT | 92.89 | -1.71 (-1.81%) |

2018-03-20 17:42:56

```go
func stockslide(deck *generate.Deck, symbols []string) {
    x := left
    y := top

    rintr := (top-bottom)/float64(len(symbols))
    size := rintr/2.5
    cintr := size*4

    var color string
    for _, s := range symbols {
        stock, err := stockapi(s)
        if err != nil || stock.Symbol == "" {
            continue
        }
        if stock.Change < 0 {
            color = negcolor
        } else {
            color = poscolor
        }
        deck.Text(x, y, stock.Symbol, "sans", size, "")
        x += cintr * 2
        deck.TextEnd(x, y, fmt.Sprintf("%.2f", stock.Price), "sans", size, "")
        x += cintr
        deck.TextEnd(x, y, fmt.Sprintf("%.2f", stock.Change), "sans", size, color)
        x += cintr
        deck.TextEnd(x, y, fmt.Sprintf("(%.2f%%)", stock.PctChange), "sans", size, color)
        x = left
        y -= rintr
    }
    deck.Text(footx, footy, time.Now().Format("2006-01-02 15:04:05"), "sans", 1.5, "yellow")
}
```

```go
package main

import (
    "os"

    "github.com/ajstarks/svgo"
)

var (
    canvas = svg.New(os.Stdout)
    width  = 500
    height = 500
)

func main() {
    a := 1.0
    ai := 0.03
    ti := 10.0

    canvas.Start(width, height)
    canvas.Rect(0, 0, width, height)
    canvas.Gstyle("font-family:serif;font-size:244pt")
    for t := 0.0; t <= 360.0; t += ti {
        canvas.TranslateRotate(width/2, height/2, t)
        canvas.Text(0, 0, "s", canvas.RGBA(255, 255, 255, a))
        canvas.Gend()
        a -= ai
    }
    canvas.Gend()
    canvas.End()
}
```

rotext.go [36]

## SVG Examples

# codepicdeck

# Genesis 3

Now the serpent was more subtil than any beast
of the field which the LORD God had made. And
he said unto the woman, Yea, hath God said,
Ye shall not eat of every tree of the garden? And
the woman said unto the serpent, We may eat
of the fruit of the trees of the garden: But of the
fruit of the tree which is in the midst of the garden,
God hath said, Ye shall not eat of it, neither shall
ye touch it, lest ye die. And the serpent said unto
the woman, Ye shall not surely die: For God doth
know that in the day ye eat thereof, then your
eyes shall be opened, and ye shall be as gods,
knowing good and evil.

# Samir Nagheenanajar

ROLE Design and UX Head
LOCATION Bridgewater, NJ

"I'm all about applying great design to all our products. We should provide a good experience in everything we do."

## TECHNOLOGY

IT and Internet

Software

## GOALS/LIKES

- To have simple intuitive tools
- Be open to new tools and methods
- To apply good design across the whole environment

## FRUSTRATIONS

- One size fits all mentality
- IT imposing tools with no real input
- Not designing for multiple form-factors and use-cases

## PERSONALITY

Extravert — Introvert

Sensing — Intuition

Thinking — Feeling

Judging — Percieving

## SUMMARY

Technically savvy, and demanding, Samir appreciates the value of good design, and is frustrated when the collaboration products fall short in this regard. On the other hand, he will actively promote tools that are designed well, and is very willing to share his knowledge and insights. Further, Samir will be a strong advocate for new effective ways of working togther.

## RELATED QUESTIONS

- Why do we use outdated products?
- What are the cultural aspects of collaboration?
- When will design be included in our requirements?
- Is the concept of the Intranet outdated?
- What is the cost to start over with new technologies?

So, the next time you're about to make a subclass, think hard and ask yourself

**what would Go do**

Andrew Mackenzie-Ross

go

| | |
|---|---|
| build | compile packages and dependencies |
| clean | remove object files |
| doc | show documentation for package or symbol |
| env | print Go environment information |
| fix | run go tool fix on packages |
| fmt | run gofmt on package sources |
| generate | generate Go files by processing source |
| get | download and install packages and dependencies |
| install | compile and install packages and dependencies |
| list | list packages |
| run | compile and run Go program |
| test | test packages |
| tool | run specified go tool |
| version | print Go version |
| vet | run go tool vet on packages |

# Go Proverbs

- Don’t communicate by sharing memory, share memory by communicating.
- Concurrency is not parallelism.
- Channels orchestrate; mutexes serialize.
- The bigger the interface, the weaker the abstraction.
- Make the zero value useful.
- interface{} says nothing.
- Gofmt’s style is no one’s favorite, yet gofmt is everyone’s favorite.
- A little copying is better than a little dependency.
- Syscall must always be guarded with build tags.
- Cgo must always be guarded with build tags.
- Cgo is not Go.
- With the unsafe package there are no guarantees.
- Clear is better than clever.
- Reflection is never clear.
- Errors are values.
- Don’t just check errors, handle them gracefully.
- Design the architecture, name the components, document the details.
- Documentation is for users.
- Don’t panic.

Don’t communicate by sharing memory, share memory by communicating.

Channels orchestrate; mutexes serialize.

Concurrency is not parallelism.

The bigger the interface, the weaker the abstraction.

Make the zero value useful.

# The new masters of our universe are people who are essentially only half-educated.

They have had no exposure to the humanities or the social sciences, the academic disciplines that aim to provide some understanding of how society works, of history and of the roles that beliefs, philosophies, laws, norms, religion and customs play in the evolution of human culture

John Naughton, The Guardian, 19 November, 2017

# dchart: charts for deck

bar

hbar

wbar

dot

vol

scatter

line

donut

pmap

pgrid

```
# AAPL Volume
2017-01-01    563.122
2017-02-01    574.969
2017-03-01    561.628
2017-04-01    373.304
2017-05-01    653.755
2017-06-01    684.178
2017-07-01    421.992
2017-08-01    661.069
2017-09-01    679.879
2017-10-01    504.291
2017-11-01    600.663
2017-12-01    417.354
```

```
<deck>
    <slide>
        <text ...>AAPL Volume</text>
        <line ... color="lightsteelblue"/>
        <text ... color="rgb(127,0,0)">563.1</text>
        <text ... color="rgb(75,75,75)">2017-01-01</text>
    </slide>
</deck>
```

# data | dchart | pdf

# Chart Types

| | |
|---|---|
| `-bar` | bar chart (default true) |
| `-wbar` | word bar chart (default false) |
| `-hbar` | horizontal bar chart (default false) |
| `-scatter` | scatter chart (default false) |
| `-dot` | dot plot (default false) |
| `-line` | line chart (default false) |
| `-vol` | volume plot (default false) |
| `-pgrid` | proportional grid (default false) |
| `-pmap` | proportional map (default false) |
| `-donut` | donut chart (default false) |

# Chart Elements

| | |
|---|---|
| `-grid` | show gridlines on the y axis (default false) |
| `-val` | show values (default true) |
| `-valpos` | value position (t=top, b=bottom, m=middle) (default "t") |
| `-yaxis` | show a y axis (default true) |
| `-yrange` | specify the y axis labels (min,max,step) |
| `-fulldeck` | generate full deck markup (default true) |
| `-title` | show the title (default true) |
| `-chartitle` | specify the title (overiding title in the data) |
| `-xlabel` | x axis label interval (default 1, 0 to supress all labels) |
| `-xlast` | show the last x label |
| `-hline` | horizontal line at value with label |

# Position and Scaling

| | |
|---|---|
| `-top` | top of the plot (default 80) |
| `-bottom` | bottom of the plot (default 30) |
| `-left` | left margin (default 20) |
| `-right` | right margin (default 80) |
| `-min` | set the minimum value |
| `-max` | set the maximum value |
| `-dmin` | data minimum (default false, min=0) |

# Measures and Attributes

| | |
|---|---|
| `-barwidth` | barwidth (default computed from data size) |
| `-ls` | linespacing (default 2.4) |
| `-textsize` | text size (default 1.5) |
| `-color` | data color (default "lightsteelblue") |
| `-vcolor` | value color (default "rgb(127,0,0)") |
| `-datafmt` | data format for values (default "%.1f") |
| `-psize` | diameter of the donut (default 30) |
| `-pwidth` | width of the donut or proportional map (default 3) |

# CSV

| | |
|---|---|
| `-csv` | read CSV files (default false) |
| `-csvcol` | specify the columns to use for label,value |

```
dchart AAPL.d
```

```
dchart -left=20 -right=80 -top=75 AAPL.d
```

```
dchart -left=20 -right=80 -top=75 -yaxis -xlabel=2 -grid AAPL.d
```

```go
package main

import (
    "fmt"
    "math"
)

func main() {
    fmt.Println("# y=sin(x)")
    for x := 0.0; x < math.Pi*2; x += 0.1 {
        fmt.Printf("%.2f\t%.4f\n", x, math.Sin(x))
    }
}
```

```
# y=sin(x)
0.00    0.0000
0.10    0.0998
0.20    0.1987
0.30    0.2955
0.40    0.3894
0.50    0.4794
0.60    0.5646
0.70    0.6442
0.80    0.7174
...
5.80    -0.4646
5.90    -0.3739
6.00    -0.2794
6.10    -0.1822
6.20    -0.0831
```

```
sine | dchart -val=f -bottom=50 -xlabel=10
```

```
sine | dchart -val=f -bottom=50 -xlabel=10 -bar=f -line -vol
```

# One Year Stock Performance (2017-02-03 to 2018-02-02)

# What is it about Go?

# fmt

# func

io.Writer

io.Reader

# net/http

encoding/xml
encoding/json

# cgo

# The Community

# gofpdf

From: Russ Cox

Subject: Re: [go-nuts] Visualizing Random Number Generators...

Date: March 5, 2010 1:14:44 EST

To: ajstarks <ajstarks@gmail.com>

---

are you going to share the library
or just tease us with pictures? ;-)

Thompson wanted to create a comfortable computing environment constructed according to his own design, using whatever means were available.

Dennis M. Ritchie, “The Development of the C Language”

Go is not the product of a Whiggish development process. We were just trying to get something that worked for us.

Rob Pike, “Origin of Go’s interface design”, golang-nuts

# Making Tools

# Fun

# Reducing the distance from the idea to the picture

Picasso

Turing

# Thank you

@ajstarks

ajstarks@gmail.com

github.com/ajstarks

speakerdeck.com/ajstarks

flickr.com/photos/ajstarks