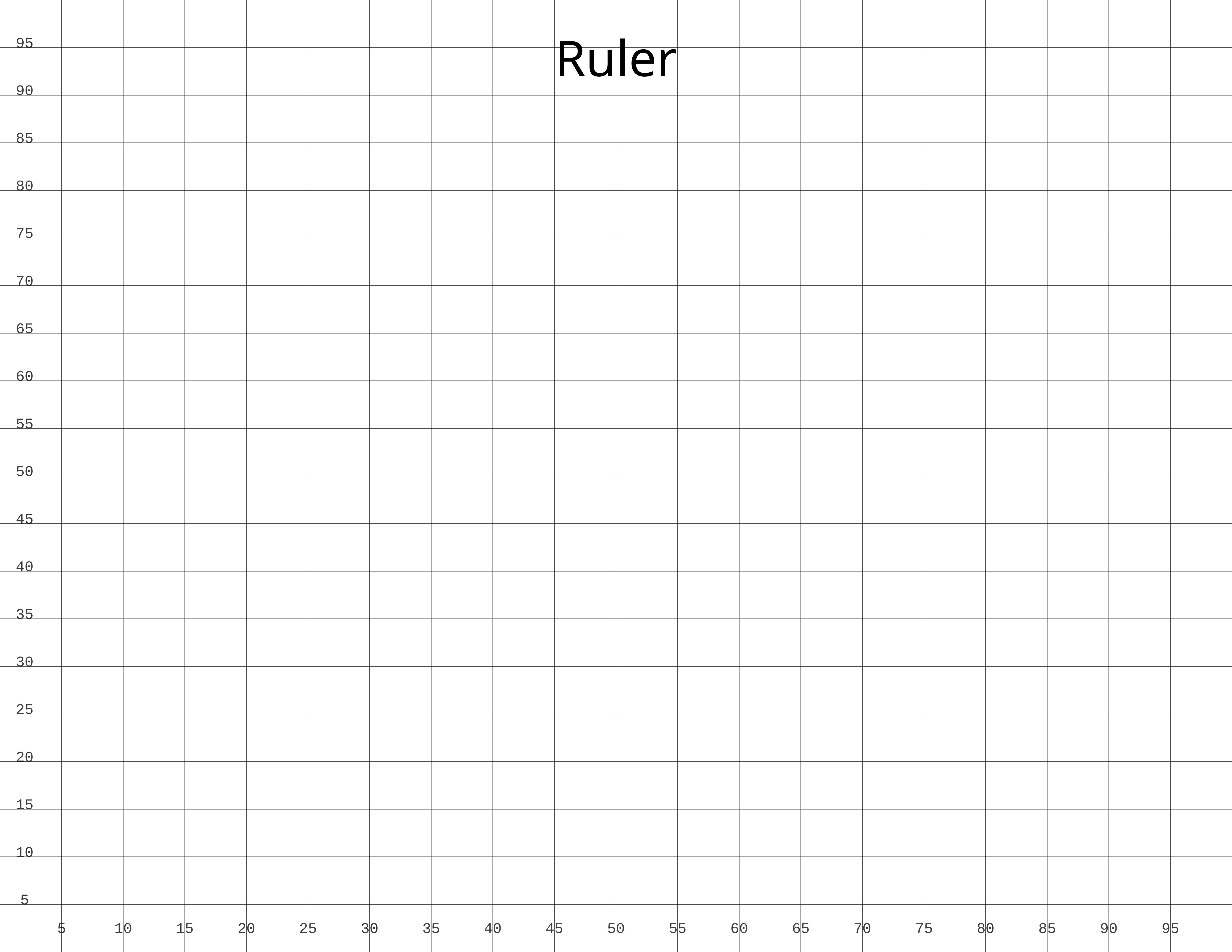
decksh tests

version

2025-11-15-1.0.0

Empty

Ruler


95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95

Ruler 20

80

60

40

20

20 40 60 80

Ruler 10 colored

90

80

70

60

50

40

30

20

10

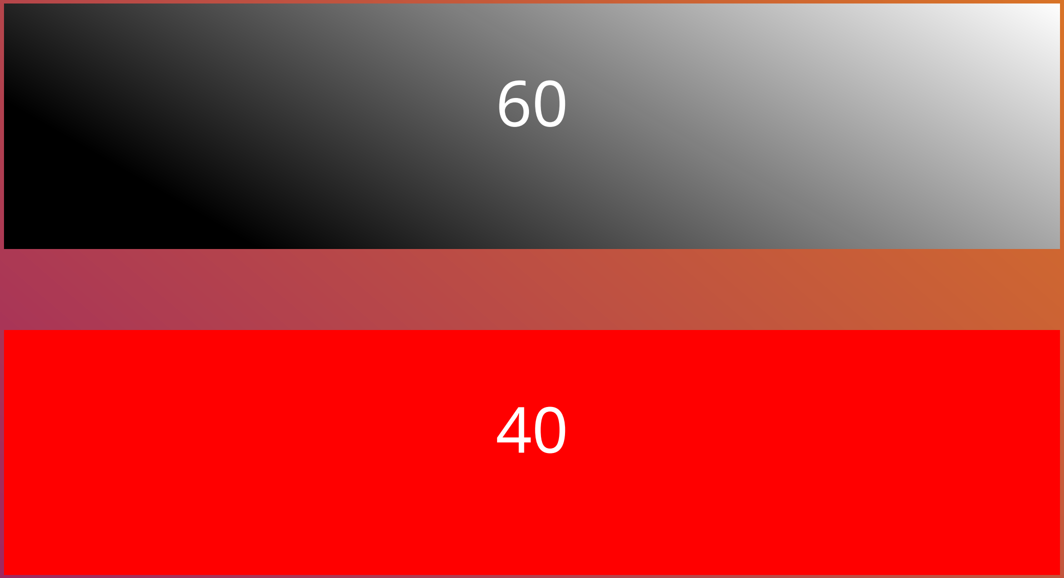
10 20 30 40 50 60 70 80 90

Background color only

Background and Foreground

Gradiant only

Gradient and Foreground

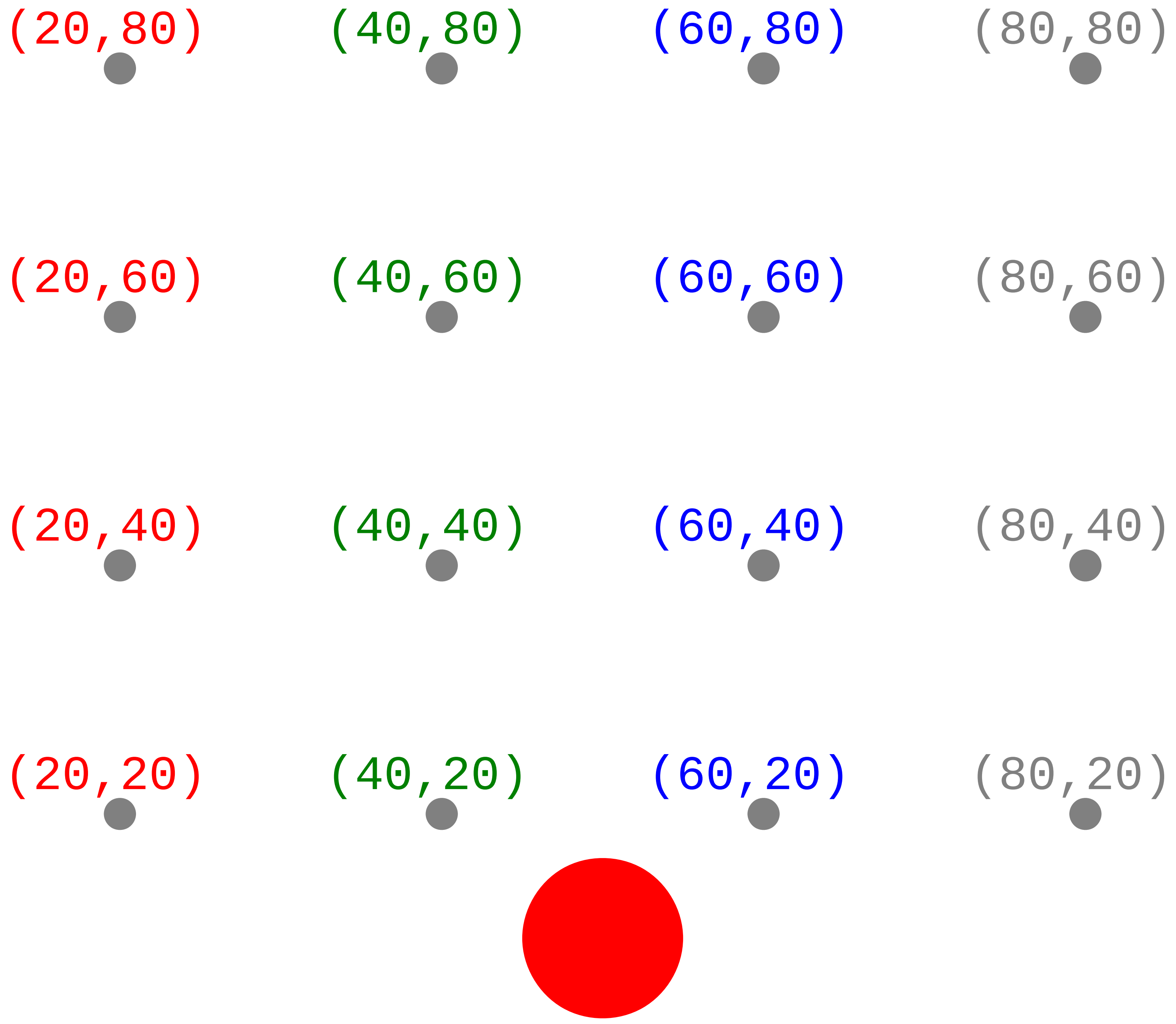


Colors, fonts, opacity

Colors	Fonts		Opacity (0-100)	
"steelblue"	"sans"	Sans Serif	100	🟥
"#4682b4"	"serif"	Serif	50	🟥
"rgb(70,130,180)"	"mono"	Monospace	20	🟥
"hsv(207,61,71)"	"symbol"	❀✷❁❄❅		
maroon/blue/90 🟪				

maroon	🟥
#800000	🟥
rgb(128,0,0)	🟥
hsv(0,100,50)	🟥

Functions



(20,80)
(40,80)
(60,80)
(80,80)
(20,60)
(40,60)
(60,60)
(80,60)
(20,40)
(40,40)
(60,40)
(80,40)
(20,20)
(40,20)
(60,20)
(80,20)

Conditionals

`r=37.22  x=14.07  b=36.28`

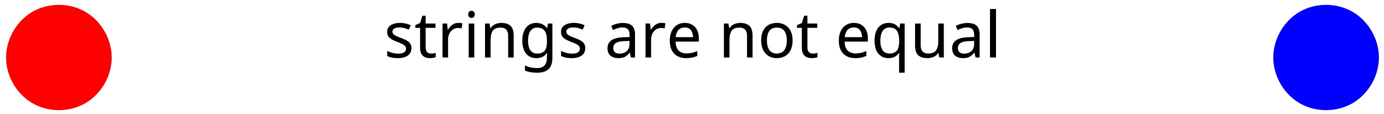
equal to	`r == x`	NO
not equal to	`r != x`	YES
greater than	`r > x`	YES
less than	`r < x`	NO
greater than or equal to	`r >= x`	YES
less than or equal to	`r <= x`	NO
between	`r >< x b`	NO

Conditionals (if -- else -- eif)

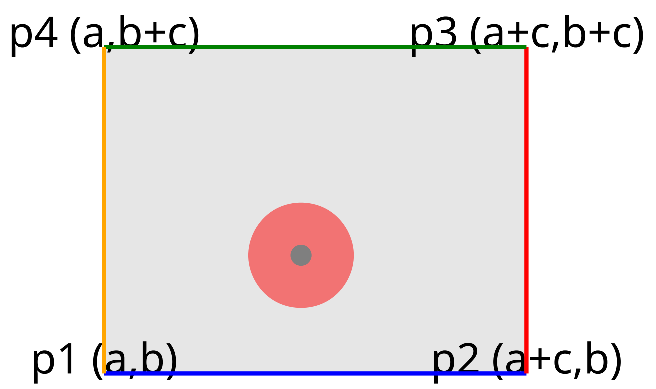
```
if rv > xv
    ctext "rv is greater than xv" 50 75 4
    ctext rval 10 75 3
    ctext xval 90 75 3
    rect 50 52 100 20 "red" 20
else
    ctext "in the else clause" 50 5 4
    ctext rval 10 5 3
    ctext xval 90 5 3
    rect 50 25 100 20 "blue" 20
eif
```

rv=18.87 in the else clause xv=51.82

String Conditionals


strings are not equal

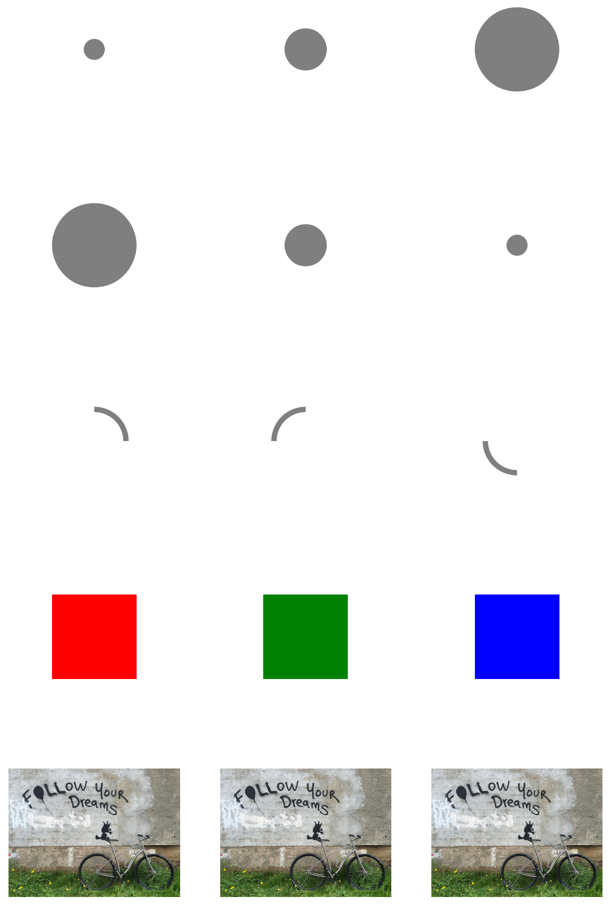
Coordinates



Included data from another file

Content (see test.md.pdf)

Grid



```
circle x y 1
circle x y 2
circle x y 4

circle x y 4
circle x y 2
circle x y 1

arc x y 3 3 0 90
arc x y 3 3 90 180
arc x y 3 3 180 270

square x y 4 "red"
square x y 4 "green"
square x y 4 "blue"

image "follow.jpg" x y 640 480 10
image "follow.jpg" x y 640 480 10
image "follow.jpg" x y 640 480 10
```

Now is the time for all good men to come to the aid of the party & 'do it now'

Now is the time for all good men to come to the aid of the party & 'do it now'

```
Now is the
time for
all good
men to come
to the aid
of the party
& 'do it
now'
```

Now is the time for all good men to come to the aid of the party & 'do it now' (read from a file)

```
package main

import (
    "fmt"
)

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

```
package main

import (
    "fmt"
)

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

# AAPL Volume (Millions)

2017-09-01	679.879
2017-10-01	504.291
2017-11-01	600.663
2017-12-01	531.184
2018-01-01	659.181
2018-02-01	927.894
2018-03-01	713.728
2018-04-01	666.154
2018-05-01	617.408
2018-06-01	527.298
2018-07-01	393.691
2018-08-01	163.768

# AAPL Volume (Millions)

2017-09-01	679.879
2017-10-01	504.291
2017-11-01	600.663
2017-12-01	531.184
2018-01-01	659.181
2018-02-01	927.894
2018-03-01	713.728
2018-04-01	666.154
2018-05-01	617.408
2018-06-01	527.298
2018-07-01	393.691
2018-08-01	163.768

```
# AAPL Volume (Millions)
2017-09-01      679.879
2017-10-01      504.291
2017-11-01      600.663
2017-12-01      531.184
2018-01-01      659.181
2018-02-01      927.894
2018-03-01      713.728
2018-04-01      666.154
2018-05-01      617.408
2018-06-01      527.298
2018-07-01      393.691
2018-08-01      163.768
```

Text and Alignment

one one one

two two two

three three three

four four four

three (180)

two (90)

one (0)

four (270)

moving on up

hello there world

this is only a test

coming down

Binary and Assignment Operators

a+b (y+=60)

a-b (y-=10)

a%b

a/b (y*-1.5)

a*b (y/=3)

Lists

one	● one	1. one
two	● two	2. two
three	● three	3. three
one	● one	1. one
two	● two	2. two
three	● three	3. three
one	● one	1. one
two	● two	2. two
three	● three	3. three
one	● one	1. one
two	● two	2. two
three	● three	3. three
one	● one	1. one
two	● two	2. two
three	● three	3. three

Centered List

one

two

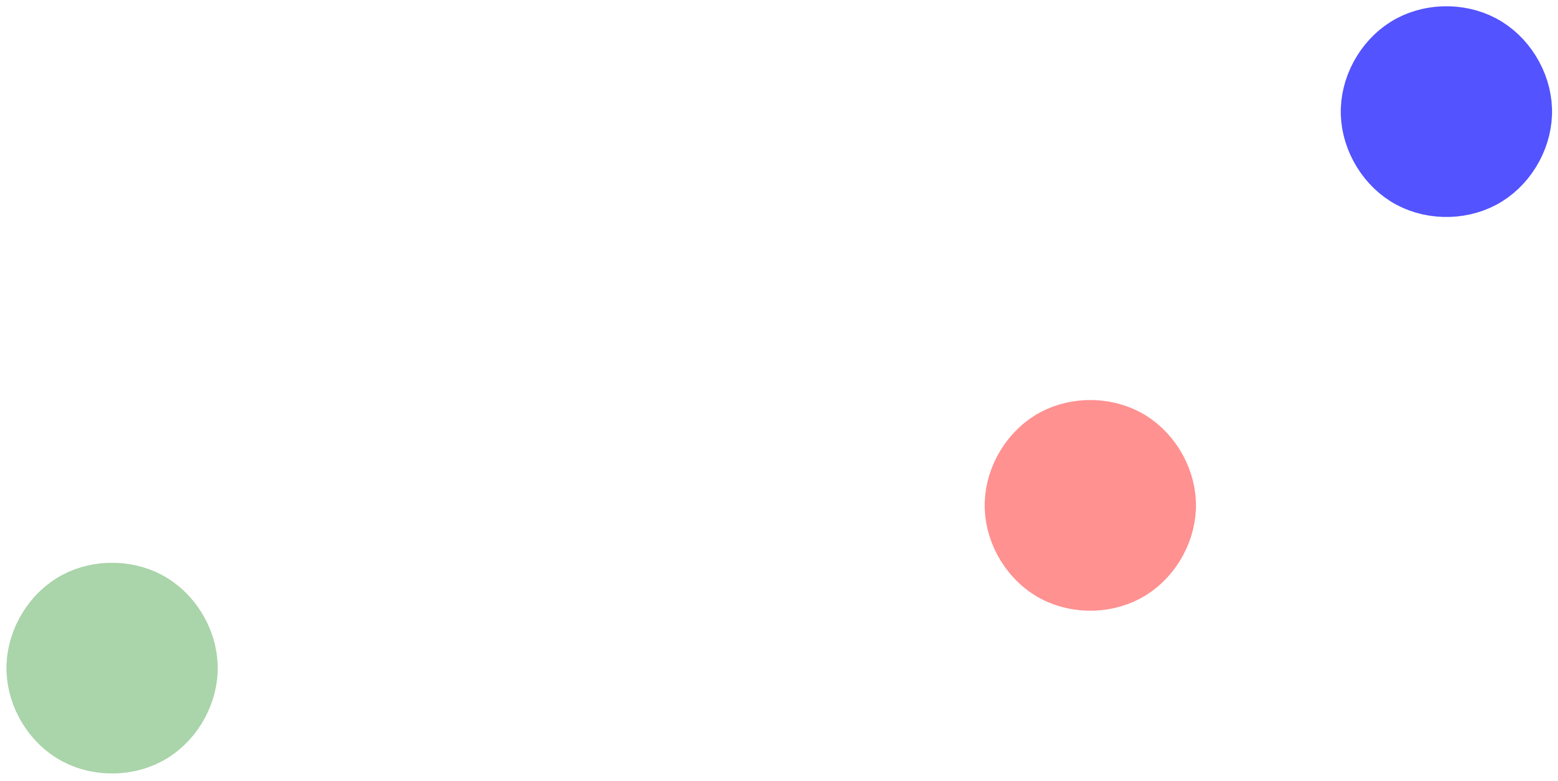
three

four

one
two
three
four

Loops

Random



Square Root

sqrt 8 = 2.8284271247461903

sqrt 8 + 6 = 3.7416573867739413

sqrt 8 - 6 = 1.4142135623730951

sqrt 8 * 6 = 6.928203230275509

sqrt 8 / 6 = 1.1547005383792515

Sine

sine 3.1415926 = 5.35897931700057245e-08

sine 3.1415926 + 0.707 = -0.6495557148113534

sine 3.1415926 - 0.707 = 0.6495557963014893

sine 3.1415926 * 0.707 = 0.7958963696196476

sine 3.1415926 / 0.707 = -0.9640809602990886

Cosine

cosine 3.1415926 = -0.9999999999999986

cosine 3.1415926 + 0.707 = -0.7603139965539972

cosine 3.1415926 - 0.707 = -0.7603139269348801

cosine 3.1415926 * 0.707 = -0.6054328772260928

cosine 3.1415926 / 0.707 = -0.2656085502930713

Tangent

tangent 3.1415926 = -5.358979317005727e-08

tangent 3.1415926 + 0.707 = 0.8543256046256702

tangent 3.1415926 - 0.707 = -0.8543257900326782

tangent 3.1415926 * 0.707 = -1.31459060047449

tangent 3.1415926 / 0.707 = 3.629706043857873

Format

Widget 1: 10.00

Widget 2: 120.000

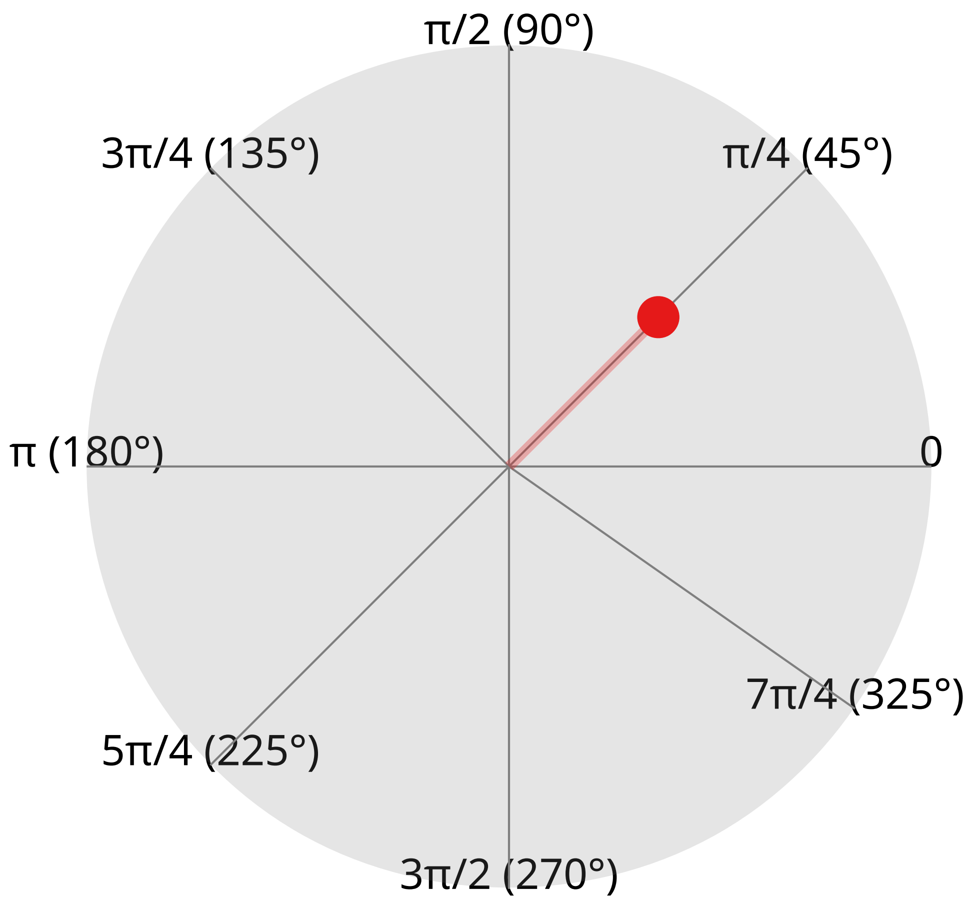
Total Widgets: 130

```
123,456,789,012,345
 12,345,678,901,234
  1,234,567,890,123
    123,456,789,012
     12,345,678,901
      1,234,567,890
        123,456,789
         12,345,678
          1,234,567
            123,456
             12,345
              1,234
                123
```

Format (2)

```
x=10
(x=10.00, y=20.00)
(x=10 y=20 z=30)
x=10 y=20 z=30 x1=66
x=10 y=20 z=30 x1=66 x2=33
x plus y=30
x minus y=-10
x divided by y=0.5
x times y=200
x mod y=10
```

Polar Coordinates



Map Ranges

1958

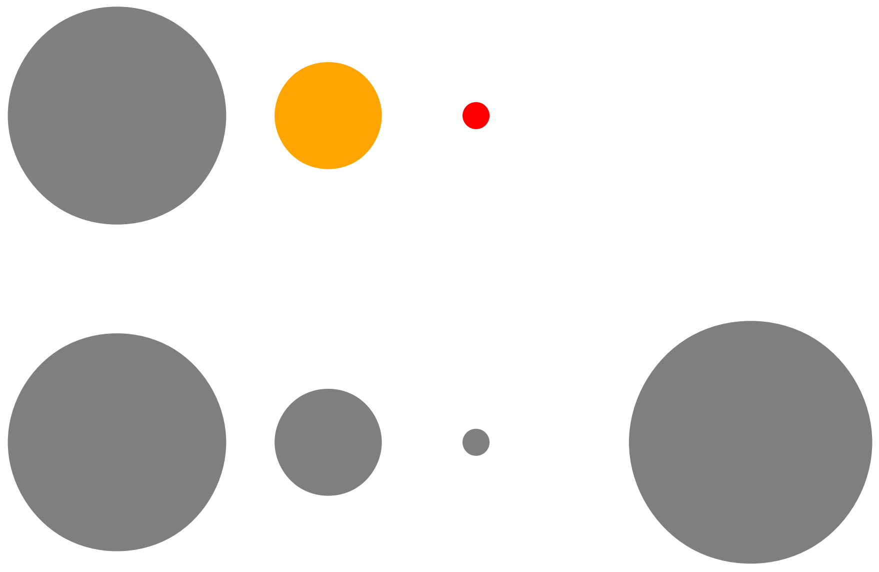
1980

1990

2020

1958 1978 1980 end

Areas

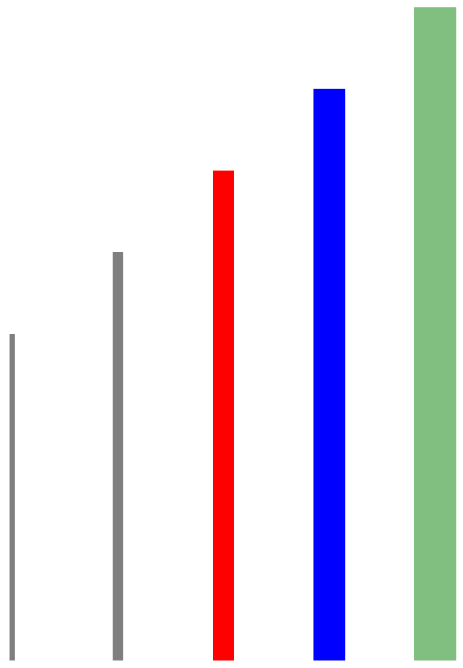


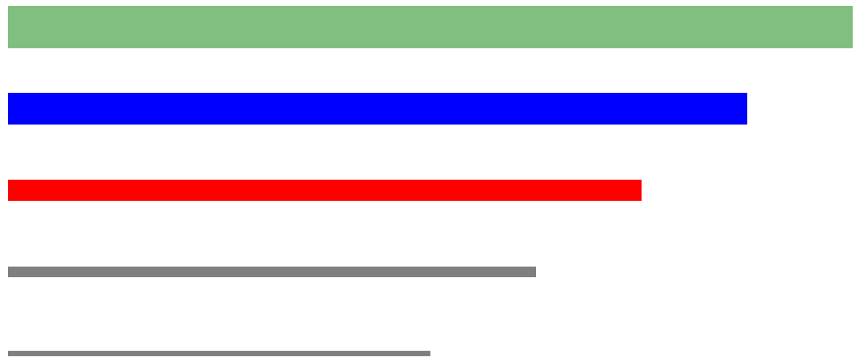
substr s begin

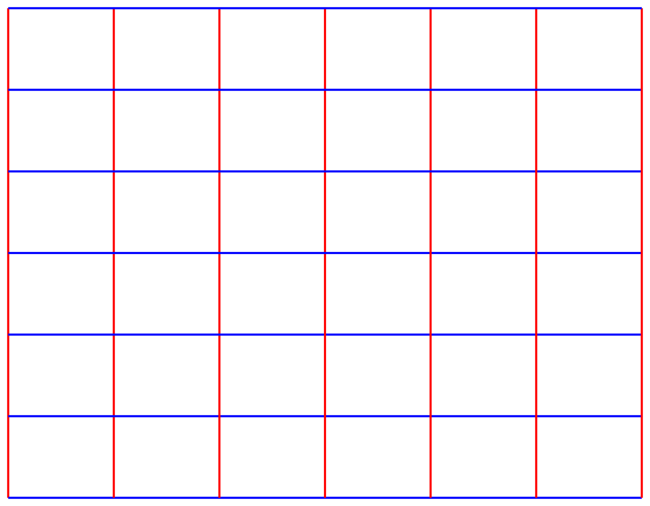
```
s="hello, world"
```

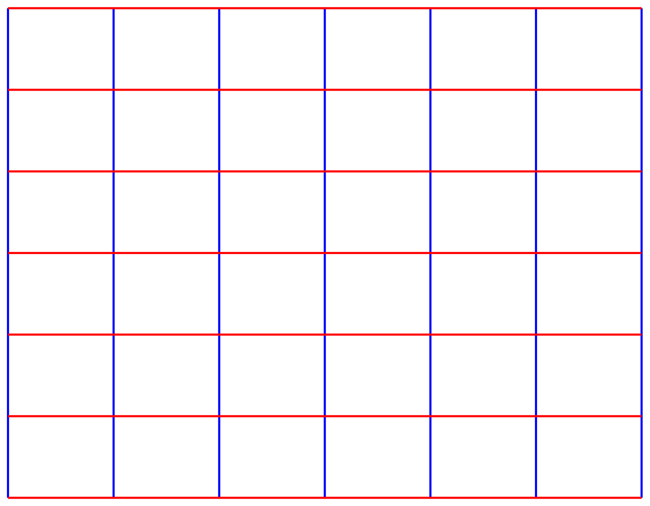
`substr s - -`	hello, world
`substr s - 4`	hello
`substr s 7 -`	world
`substr s 3 8`	lo, wo
`substr "This is a test" 5 8`	is a

Lines



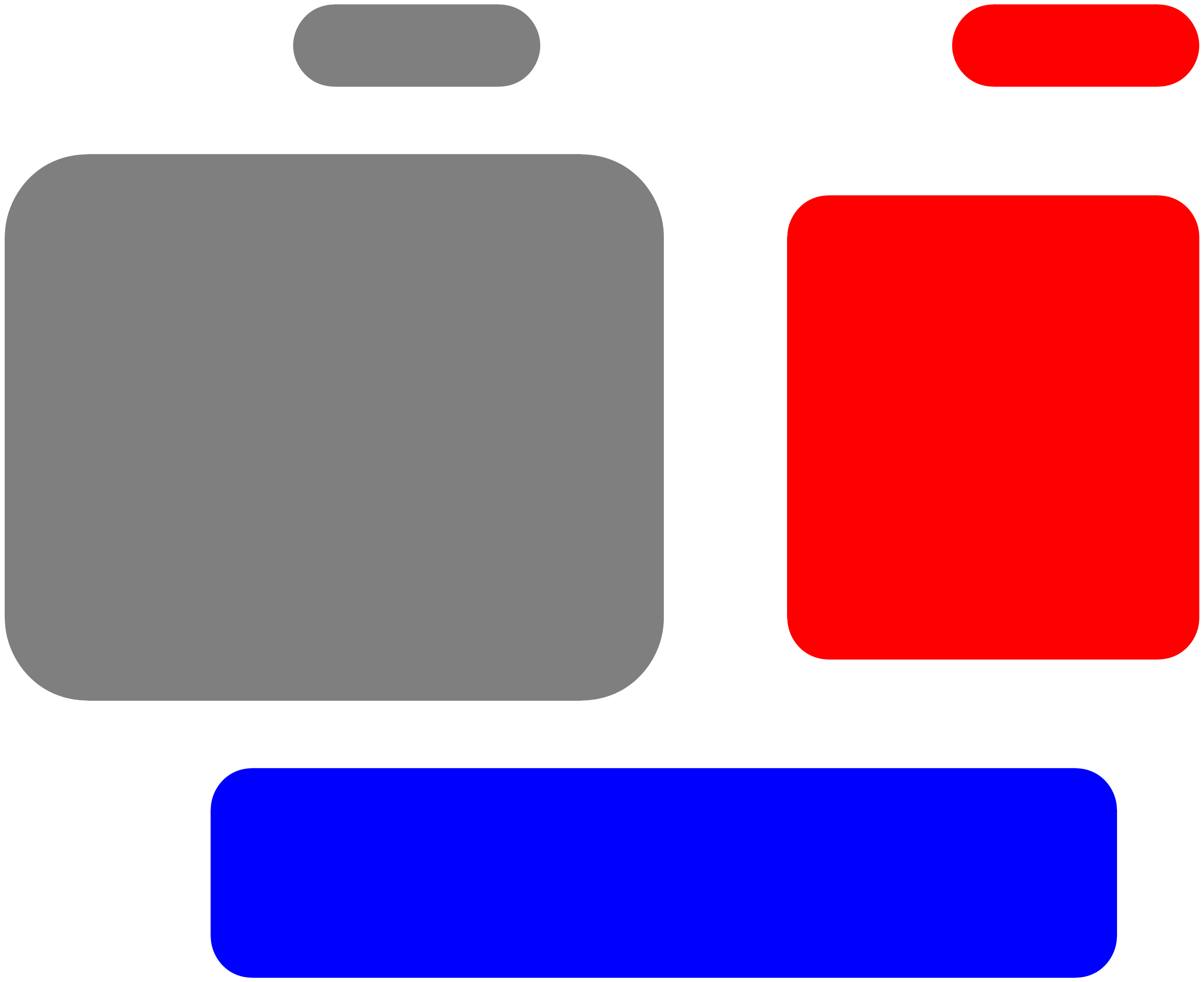


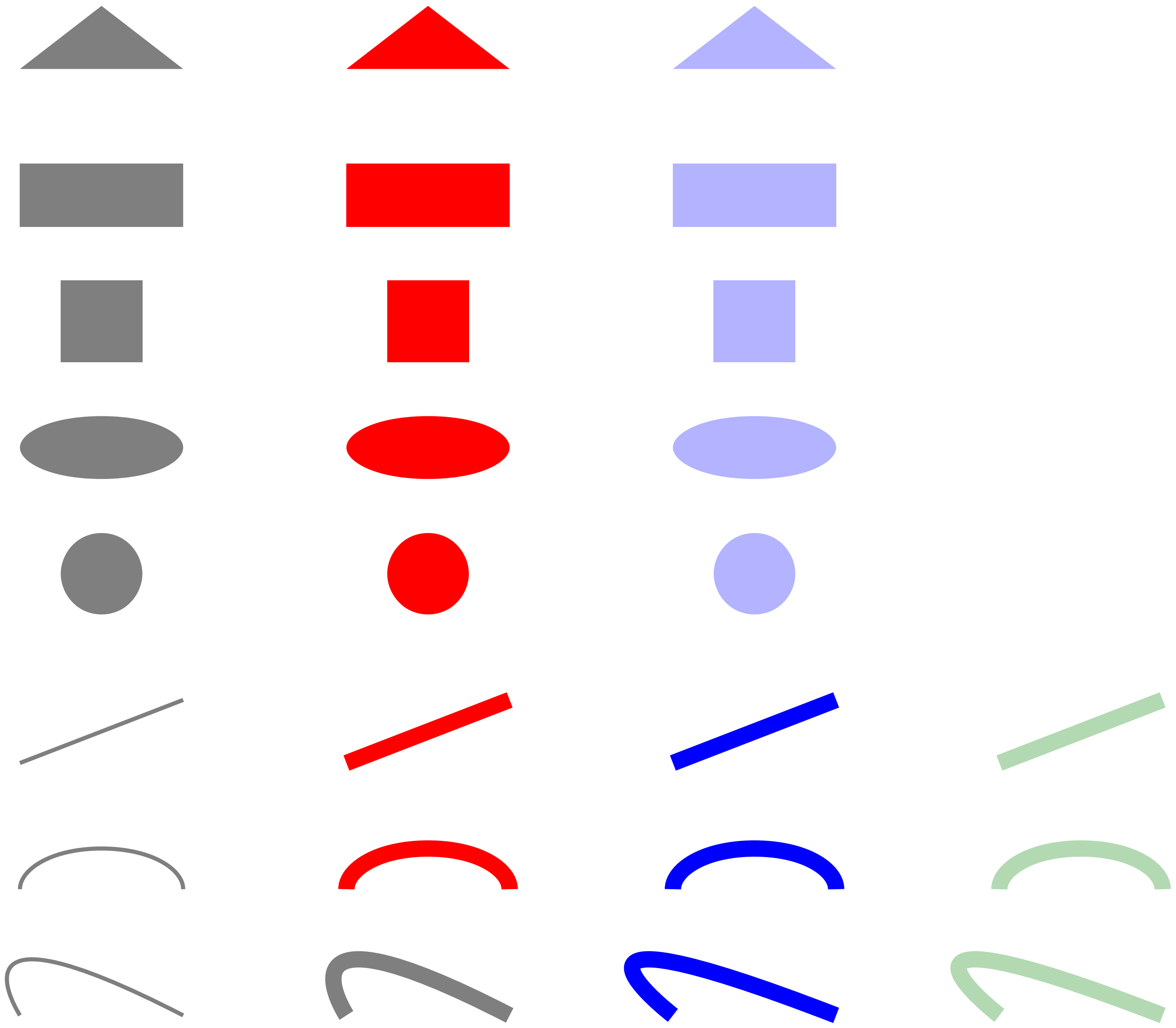




Stars

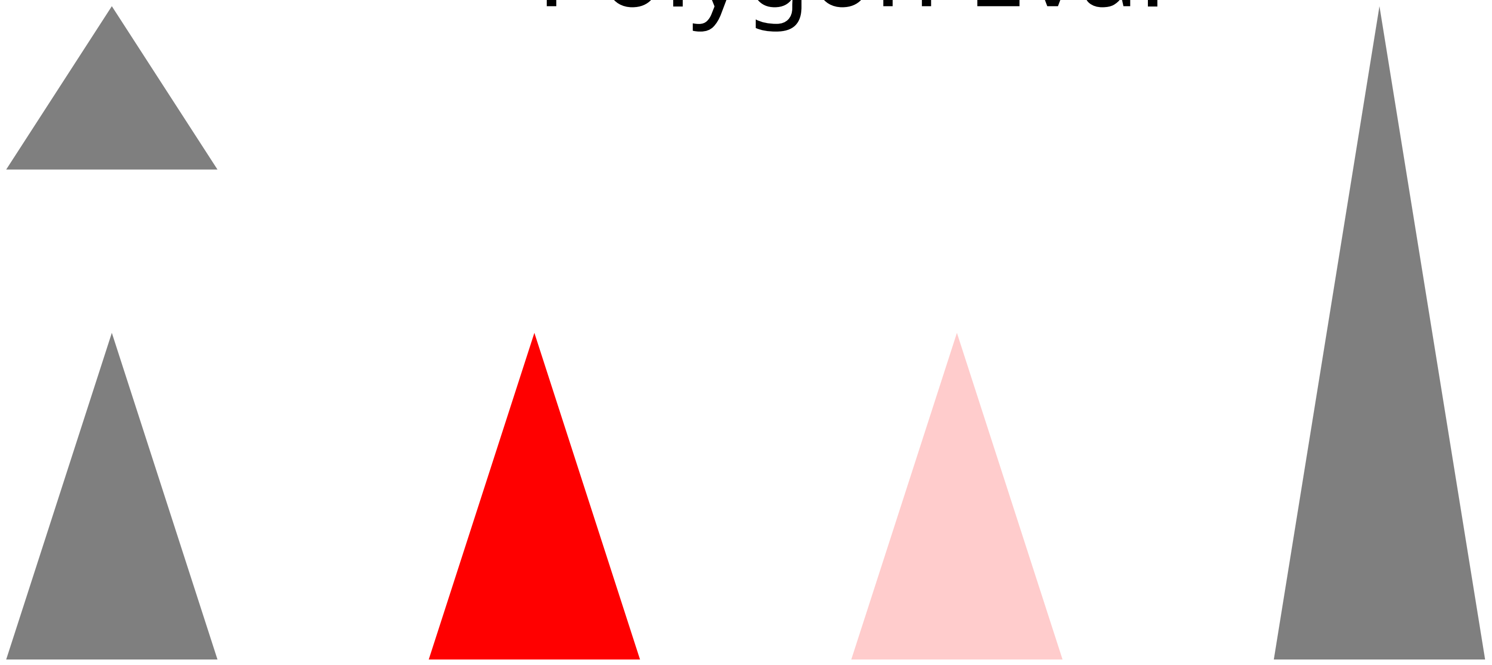
Pill/Rounded Rectangles



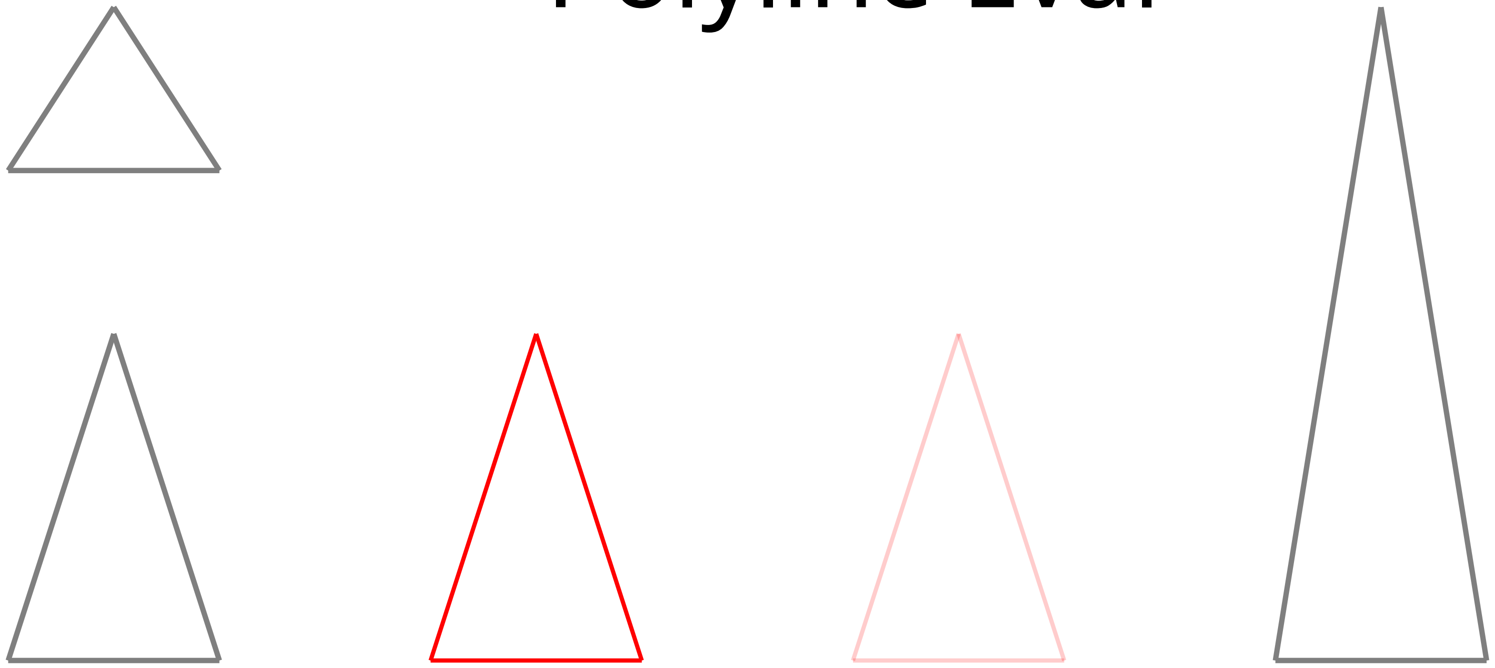


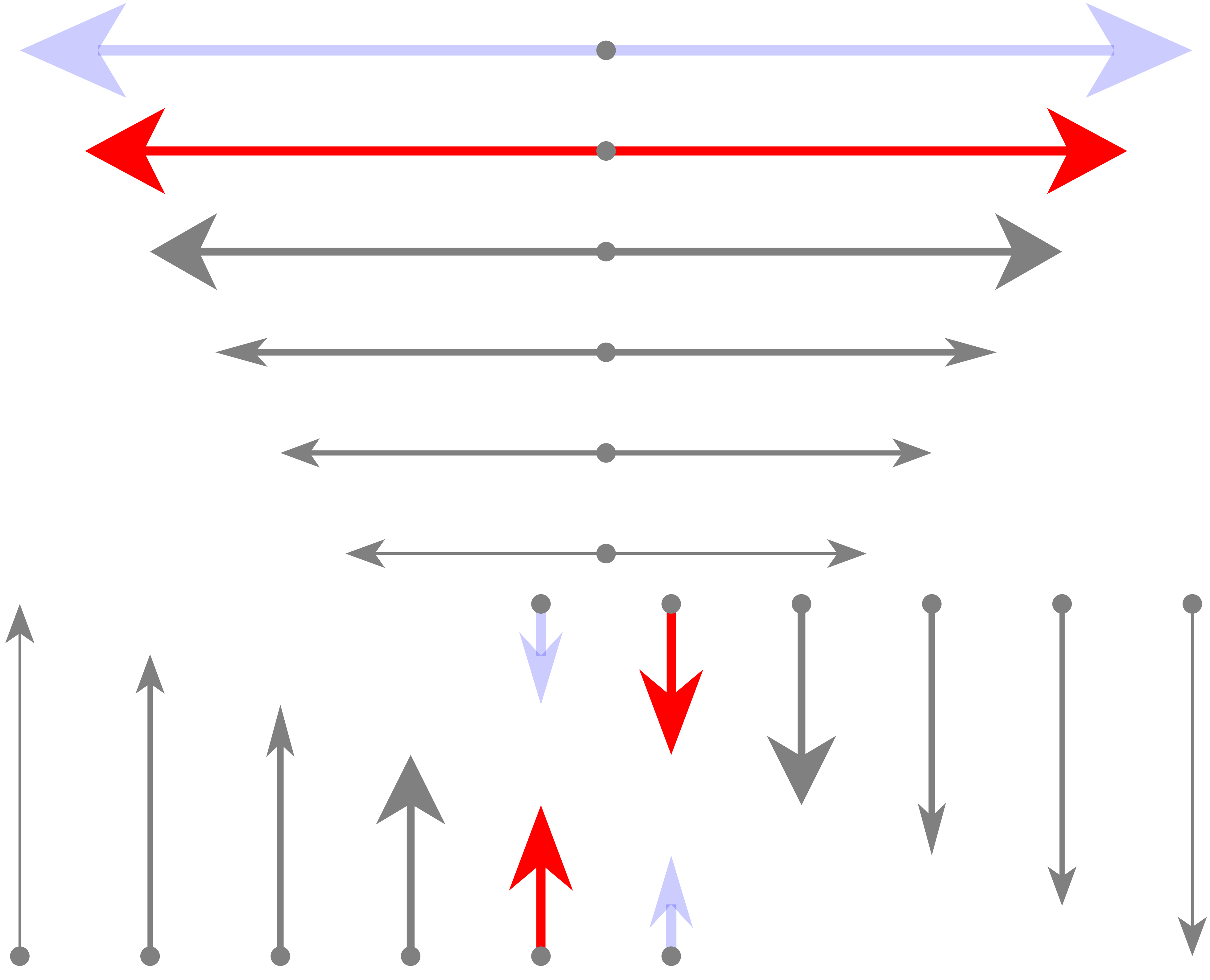
Shapes

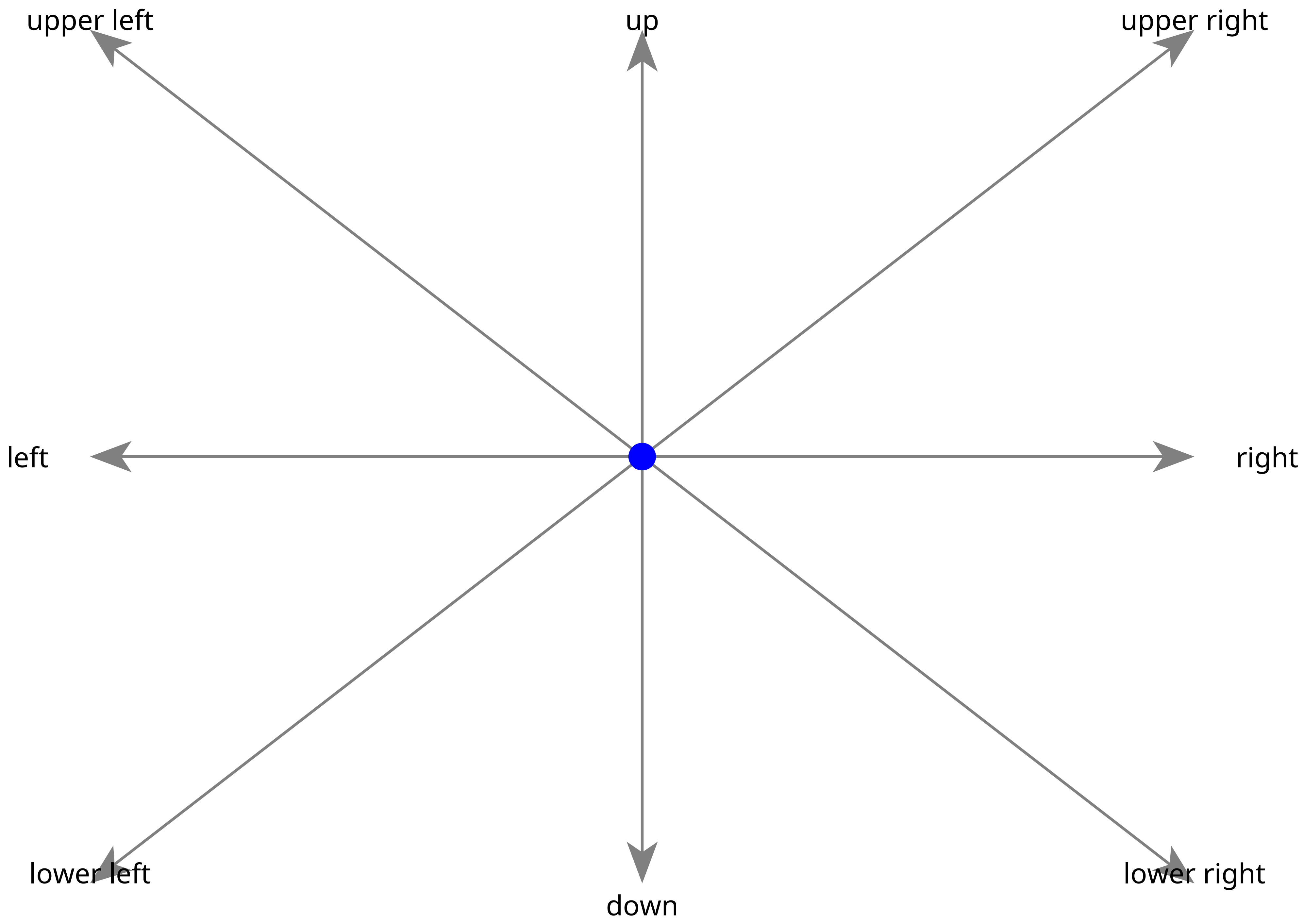
Polygon Eval

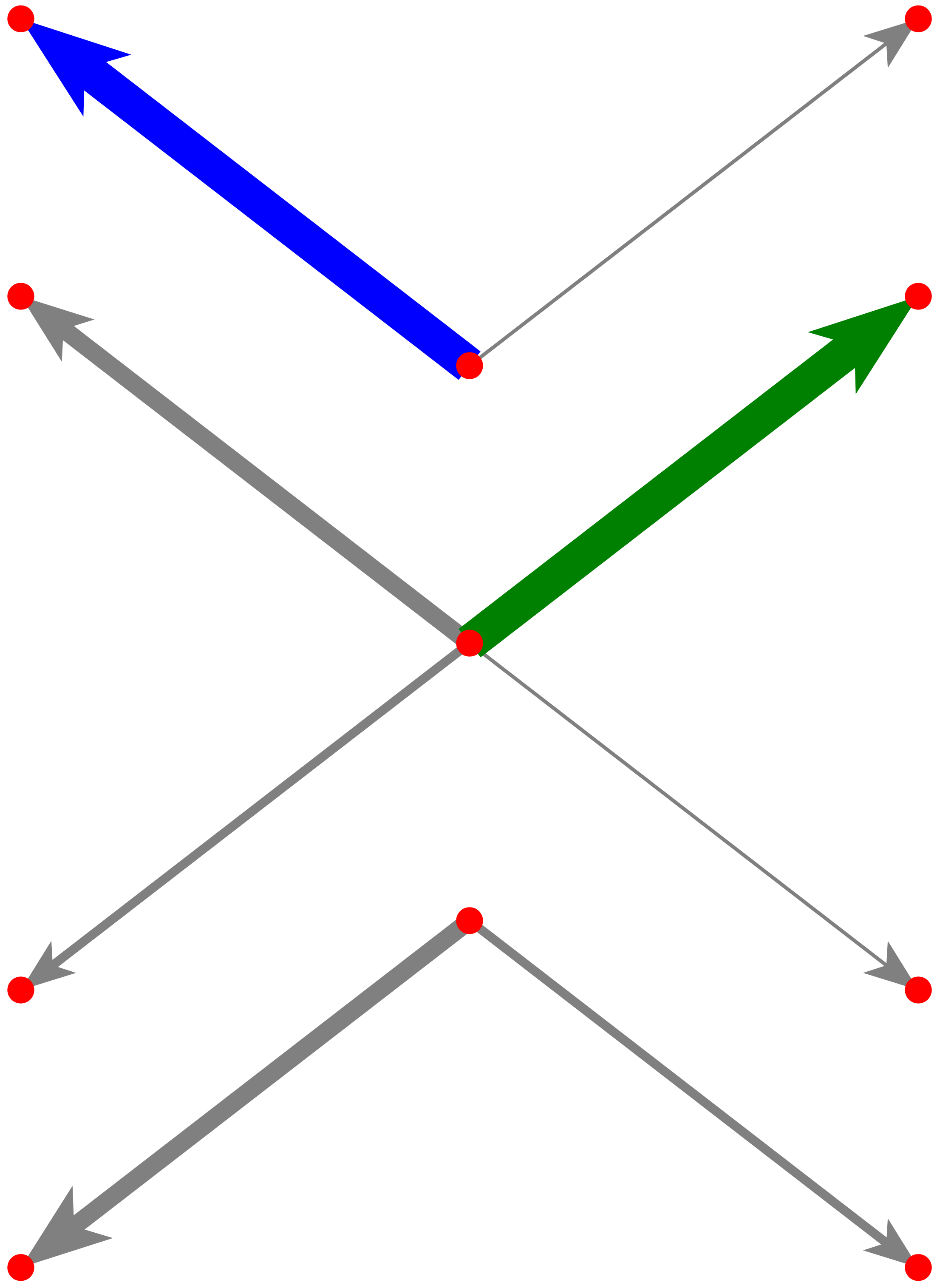


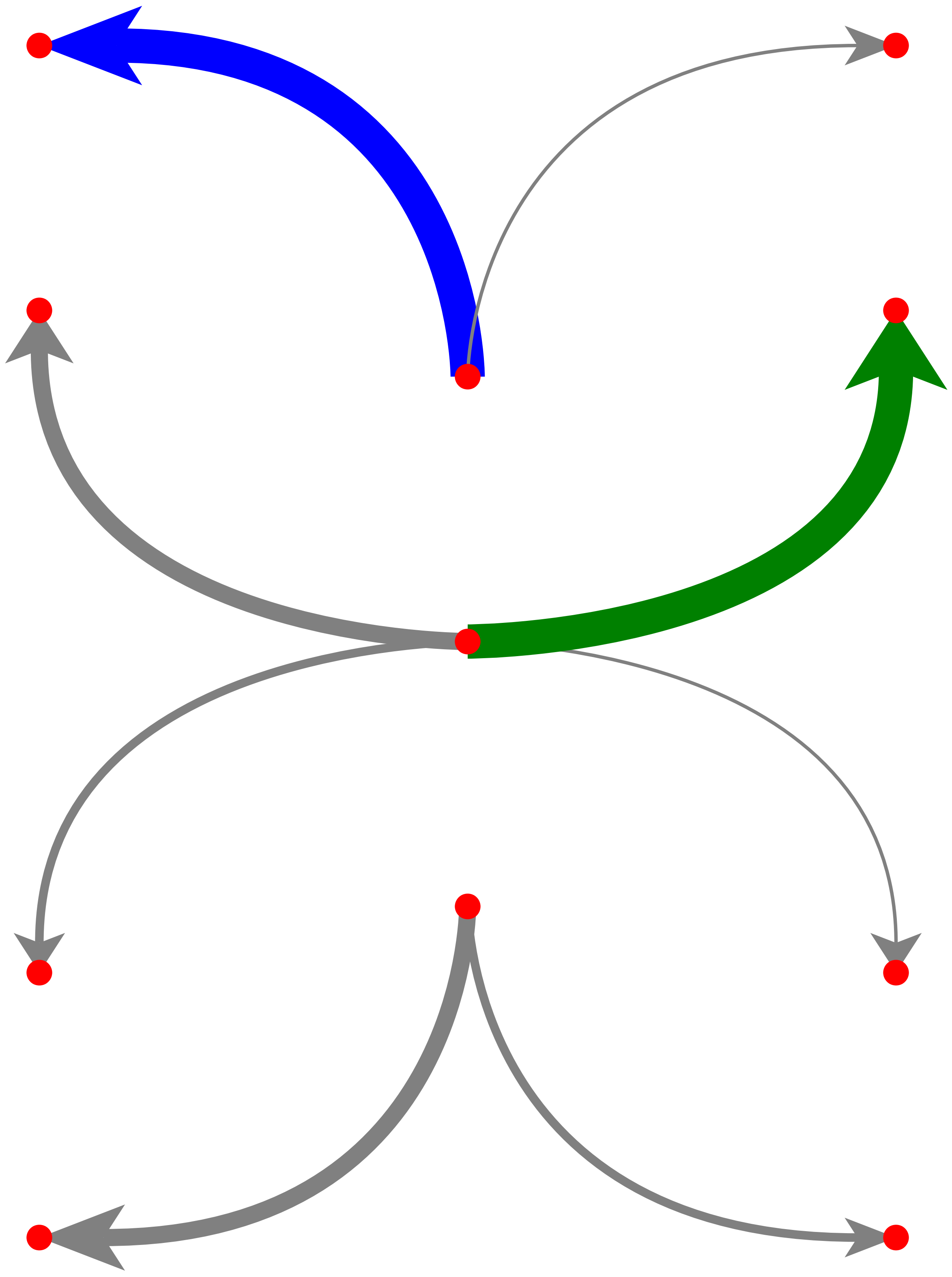
Polyline Eval

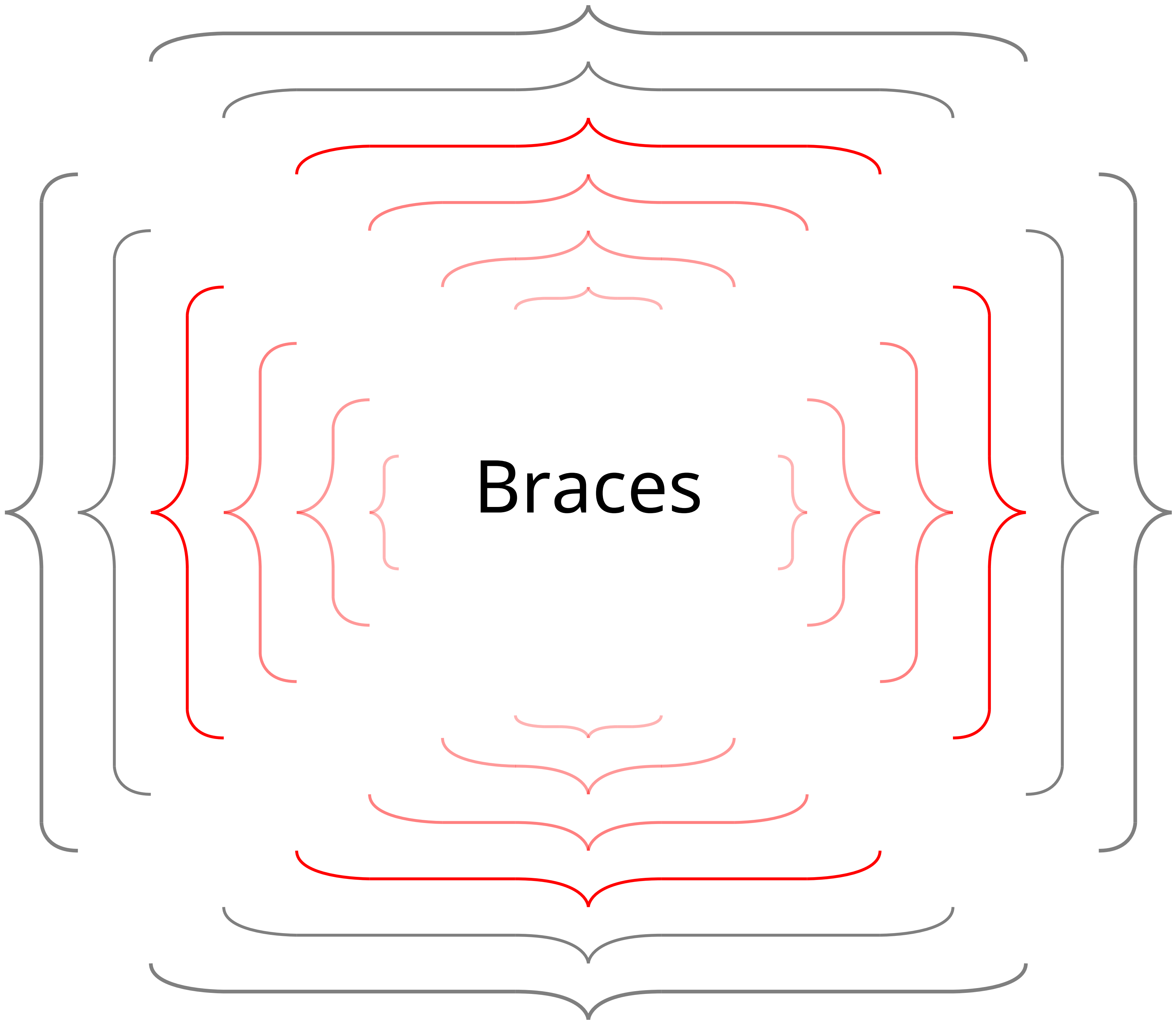


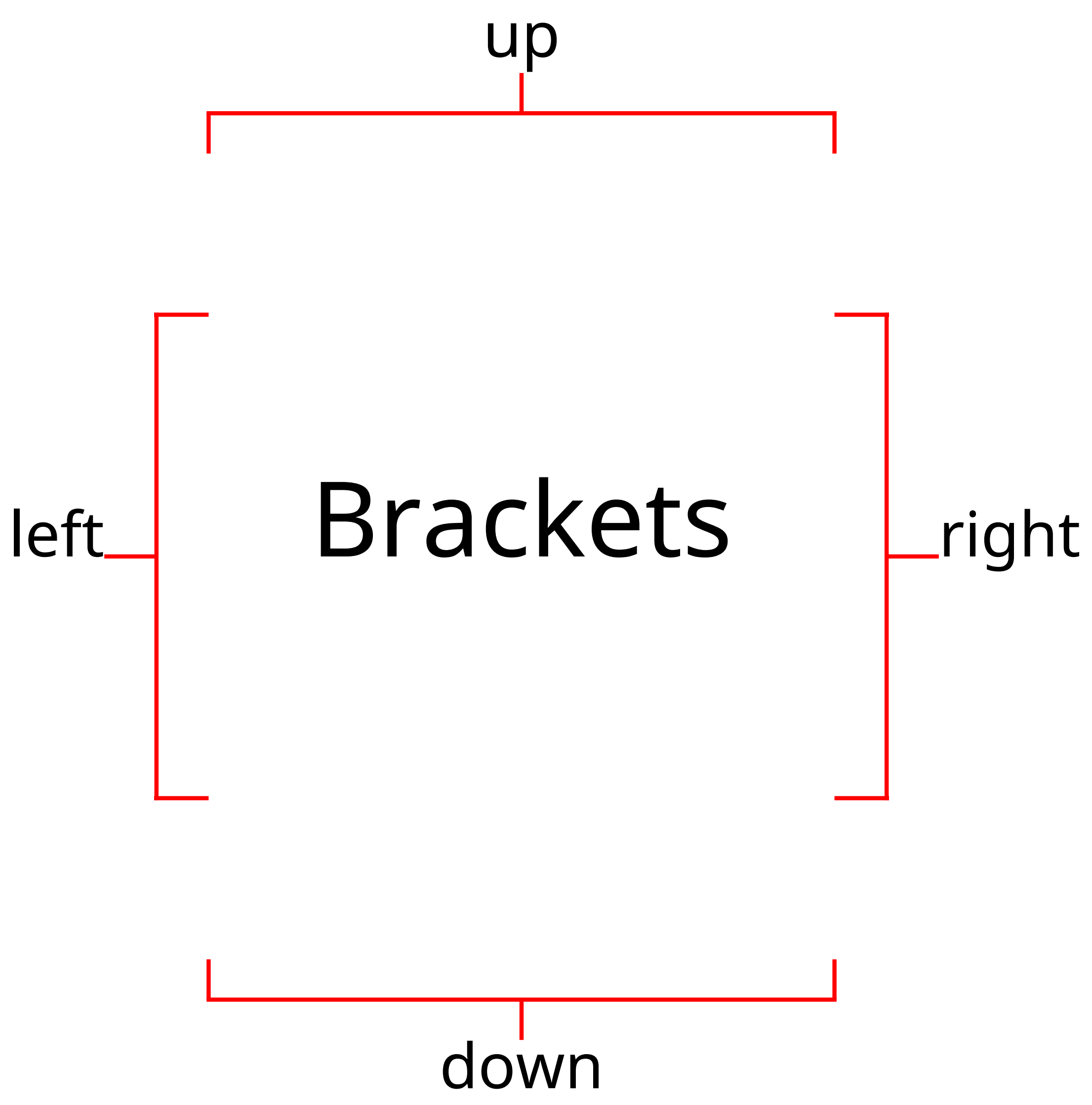


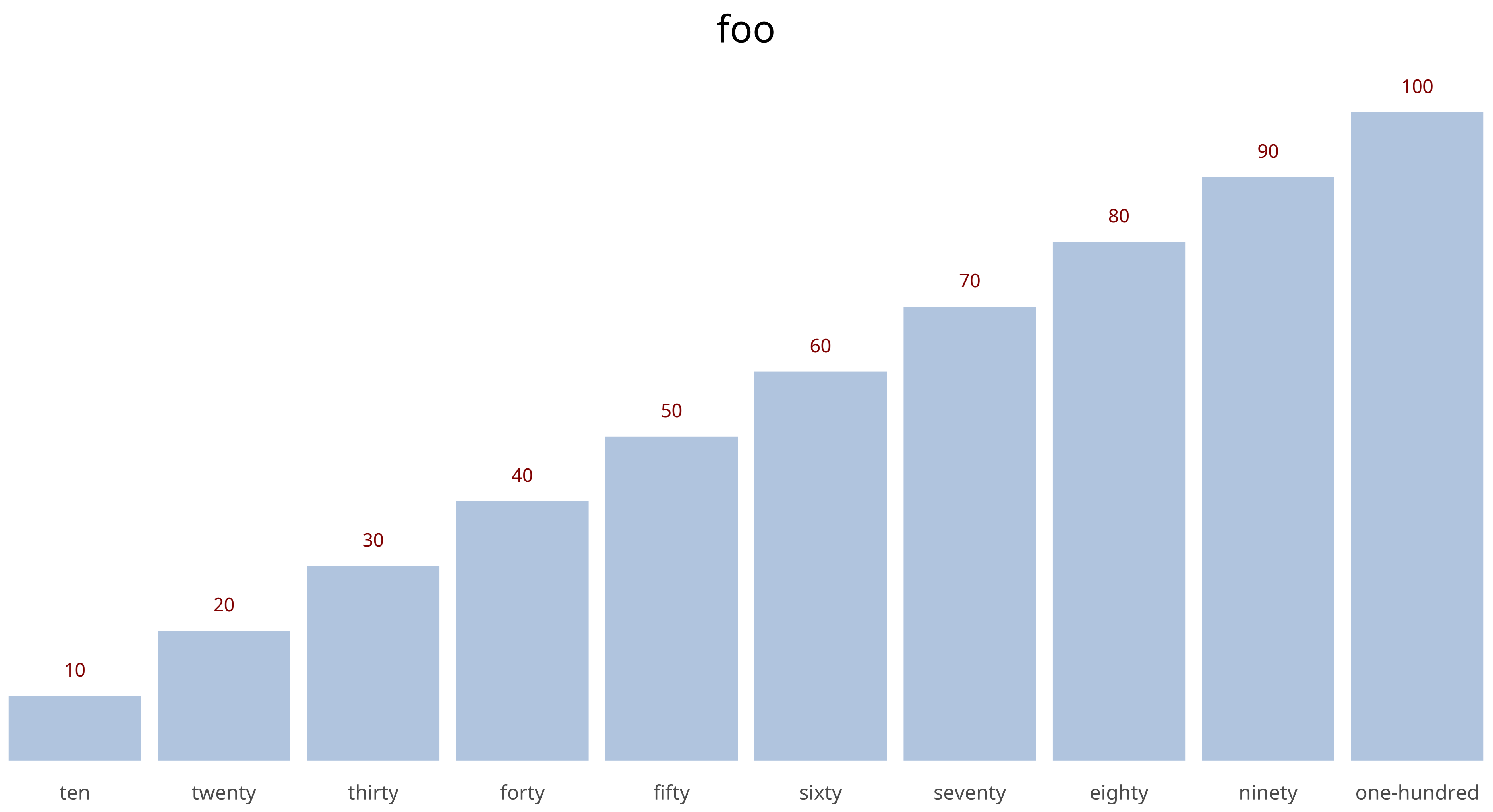

upper left
up
upper right
left
right
lower left
down
lower right

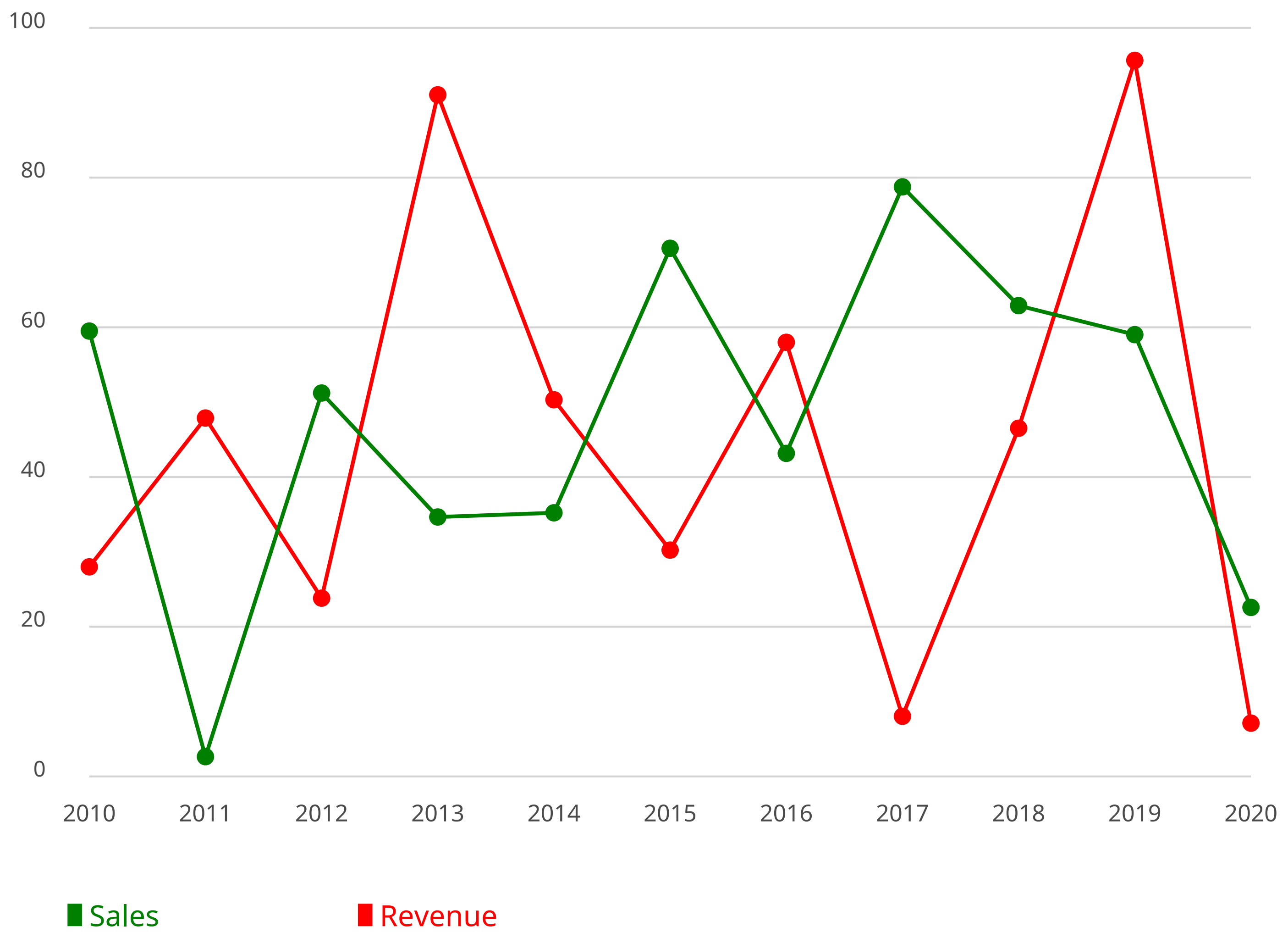





Braces


up
left
Brackets
right
down


foo
10
20
30
40
50
60
70
80
90
100
ten
twenty
thirty
forty
fifty
sixty
seventy
eighty
ninety
one-hundred


100
80
60
40
20
0
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
🟩 Sales
🟥 Revenue

FOLLOW YOUR Dreams

LARGE

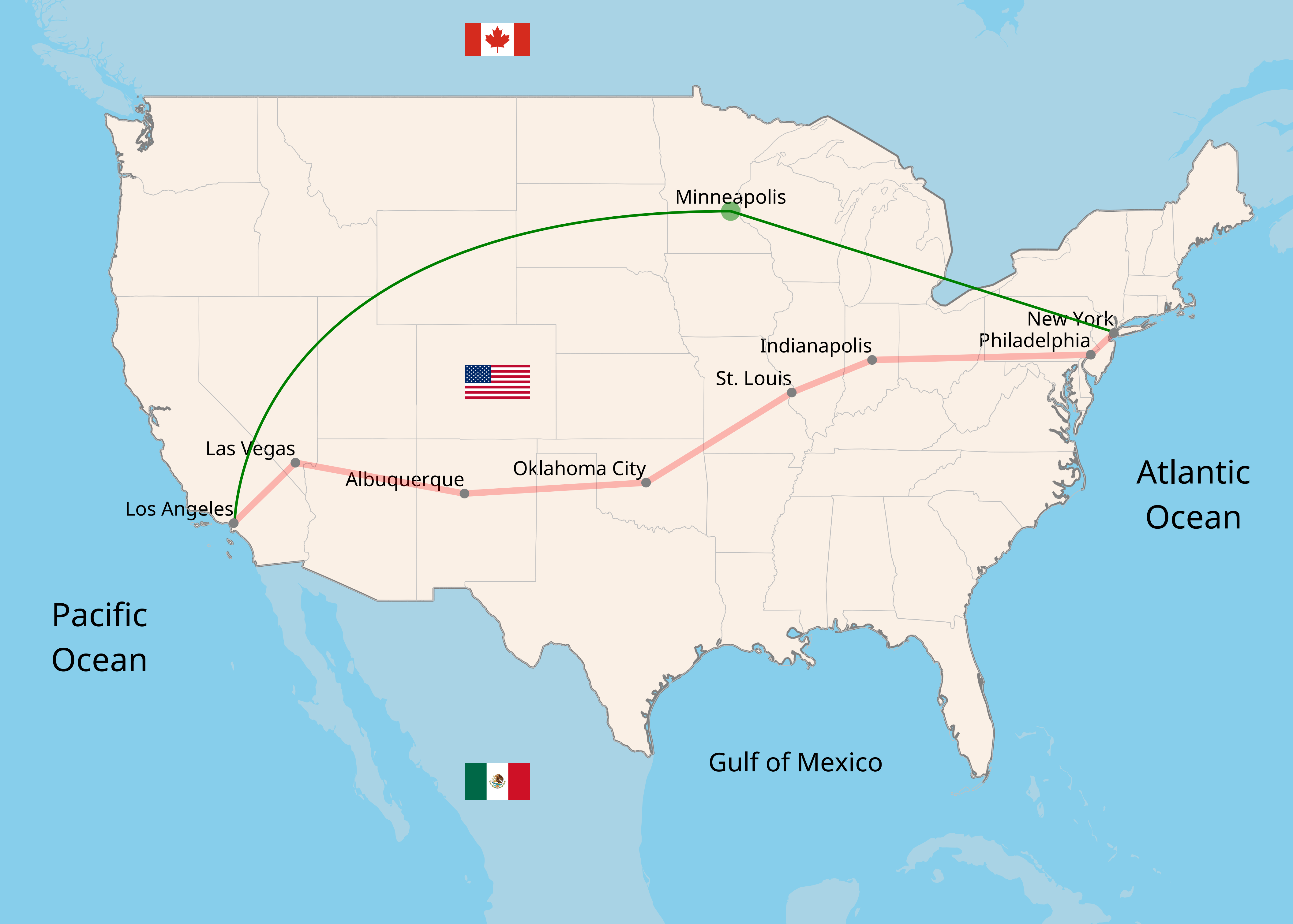
Width Scaled Image

10%

30%

50%

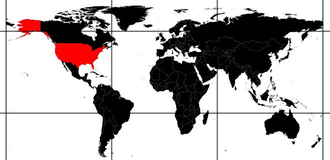
Geographic Funtions



text

geo

Deck elements



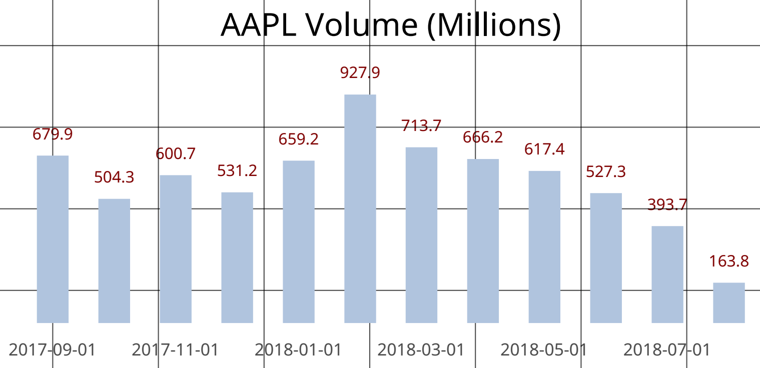
list

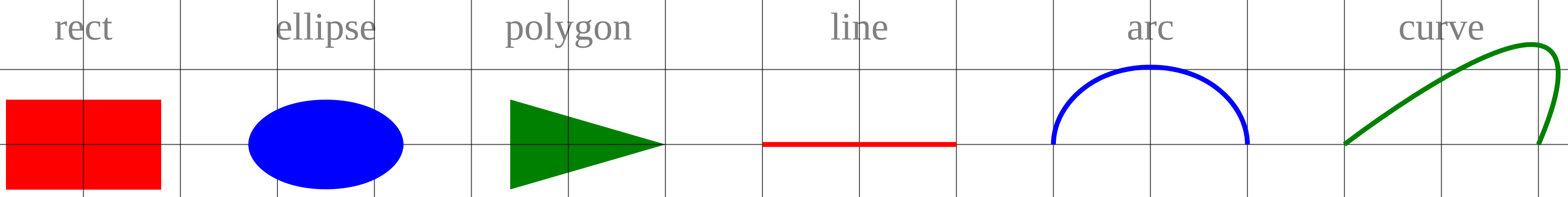
image

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

chart

Dreams





95 90 85 80 75 70 65 60 55 50 45 40 35 30 25 20 15 10 5

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95