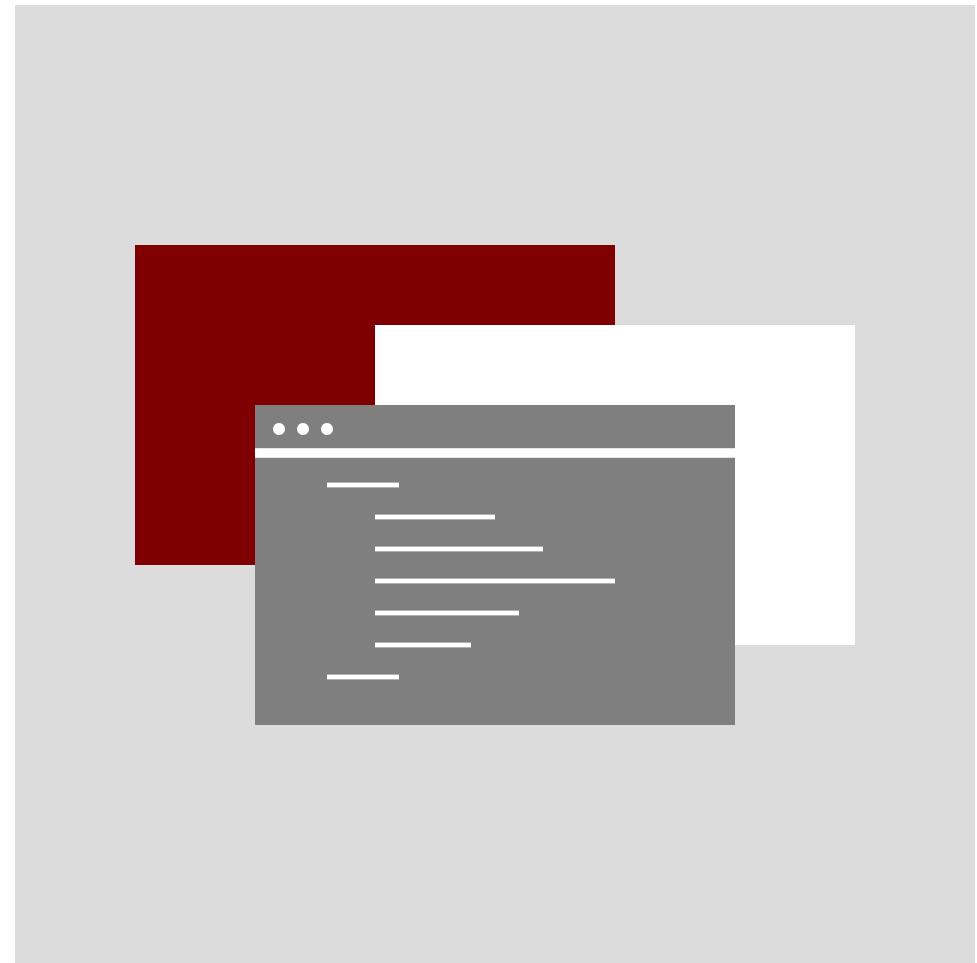


decksh reference



Version
2025-12-22-1.0.0

Introduction

decksh is a Domain-Specific Language (DSL) for making presentations, visualizations, and information displays.

This reference describes the keywords and elements of the language, how to structure decksh code, along with how to use variables, assignments, and binary operations.

Also included is a color reference and a detailed description, with examples, for all decksh elements.

Keywords

Structure Text

deck/edeck text
slide/eslide btext
canvas ctext
def/edef etext
for/efor rtext
func arctext
grid textblock
import textblockfile
include textfile
if/else/eif textcode

Graphics

acircle polygon
arc polyline
circle rect
curve rrect
ellipse ruler
hline square
line star
pill vline

Lists

list
blist
nlist
clist
li
elist

Braces Arrows

lbrace arrow
rbrace rarrow
ubrace larrow
dbrace uarrow
lbracket darrow
rbracket darrow
dbracket ubarrow
ubracket

Maps

geoarc
geoborder
geoimage
geolabel
geoloc
geomark
geopath
geopathfile
georegion

Images

image
cimage

Charts

dchart
legend
areachart
barchart
hbarchart
wbarchart
linechart
scatterchart

Assignments

polar
polarx
polary
random
vmap
dump

Math

cosine
sine
sqrt
tangent

Data

data
edata
content

Keywords and arguments

keyword

arguments

mandatory

optional

text

"..string...." x y n

"font" "color" op

text

"hello, world" 80 50 2

hello, world

text

"hello, world" 80 40 2

"serif"

hello, world

text

"hello, world" 80 30 2

"serif" "red"

hello, world

text

"hello, world" 80 20 2

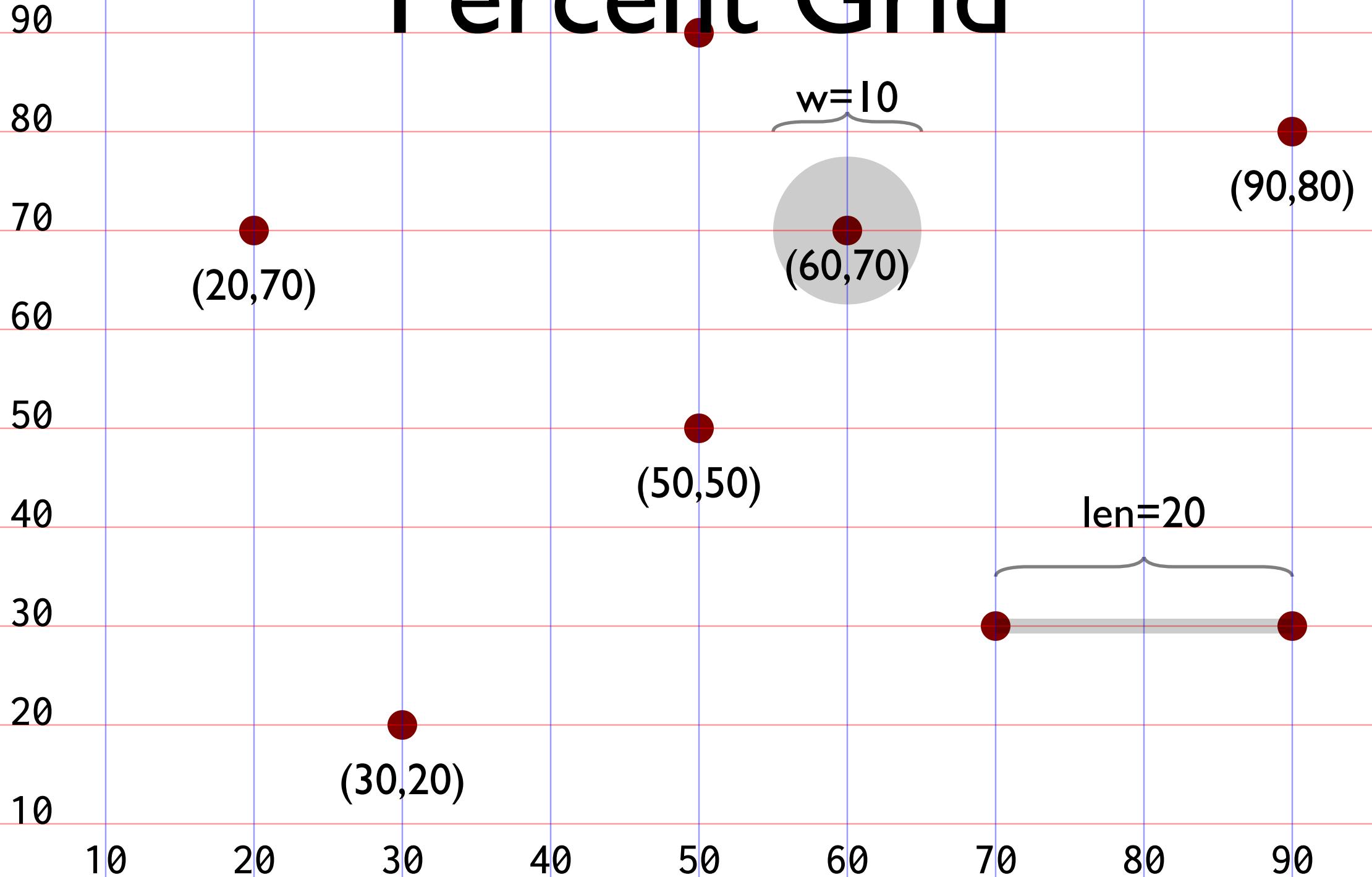
"serif" "red" 50

hello, world

Structure

```
// This is a comment           ← comment
deck
    canvas 1920 1080          ← canvas size hint (width height)
    variables {                ←
        x=20      // define x ← inline comment
        y=80
    }
    slide {                    ←
        slide 1 {                ←
            text "first" x y 2
            eslides
        }
        slide 2 {                ←
            slide "black" "white"
            include "file.dsh" {←
                ctext "hello, world" 50 25 10
                circle 50 0 100 "blue"
                for x=20 80 10
                    circle x 75 2
                efor
            }
            eslides
        }
    }
edek
```

Percent Grid



Defining and using variables

name	is	thing	
x	=	3.14159265	number
s	=	"hello"	string
y	=	x	another variable
x	=	a + b	binary operation
x	*=	10	assignment operator
text	s	x	variable use
dump	[var	list]	dump the value of variables

Binary operators

<code>x = a + b</code>	addition
<code>x = a - b</code>	subtraction
<code>x = a * b</code>	multiplication
<code>x = a / b</code>	division
<code>x = a % b</code>	modulo

Assignment operators

`x += 10`

increase x by 10

`x -= 10`

decrease x by 10

`x *= 10`

multiply x by 10

`x /= 10`

divide x by 10

Special Assignments

p=(expr, expr)	coordinates (p_x, p_y)
p=polar cx cy r theta	polar coordinate (p_x, p_y)
x=polarx cx cy r theta	polar coordinate (x)
y=polary cx cy r theta	polar coordinate (y)
v=format string expr	number formatting
v=substr string begin end	substring
v=random v1 v2	random number
v=vmap data v1 v2 v3 v4	range map
v=area expr	area
v=cosine expr	cosine
v=sine expr	sine
v=sqrt expr	square root
v=tangent expr	tangent

Colors, fonts, opacity, gradient

Colors	Fonts			Opacity (0-100)
"steelblue"	"sans"	Sans Serif	100	
"#4682b4"	"serif"	Serif	50	
"rgb(70,130,180)"	"mono"	Monospace	25	
"hsv(207,61,71)"	"symbol"	✿❀❀✿❀✿	10	

"red/blue/90"

(applies to rect and square only)

Color Index

name	hex	RGB	name	hex	RGB
aliceblue	#f0f8ff	rgb(240,248,255)	coral	#ff7f50	rgb(255,127,80)
antiquewhite	#faebd7	rgb(250,235,215)	cornflowerblue	#6495ed	rgb(100,149,237)
aqua	#00ffff	rgb(0,255,255)	cornsilk	#fff8dc	rgb(255,248,220)
aquamarine	#7fffad	rgb(127,255,212)	crimson	#dc143c	rgb(220,20,60)
azure	#f0ffff	rgb(240,255,255)	cyan	#00ffff	rgb(0,255,255)
beige	#f5f5dc	rgb(245,245,220)	darkblue	#00008b	rgb(0,0,139)
bisque	#ffe4c4	rgb(255,228,196)	darkcyan	#008b8b	rgb(0,139,139)
black	#000000	rgb(0,0,0)	darkgoldenrod	#b8860b	rgb(184,134,11)
blanchedalmond	#ffebcd	rgb(255,235,205)	darkgray	#a9a9a9	rgb(169,169,169)
blue	#0000ff	rgb(0,0,255)	darkgreen	#006400	rgb(0,100,0)
blueviolet	#8a2be2	rgb(138,43,226)	darkgrey	#a9a9a9	rgb(169,169,169)
brown	#a52a2a	rgb(165,42,42)	darkkhaki	#bdb76b	rgb(189,183,107)
burlywood	#deb887	rgb(222,184,135)	darkmagenta	#8b008b	rgb(139,0,139)
cadetblue	#5f9ea0	rgb(95,158,160)	darkolivegreen	#556b2f	rgb(85,107,47)
chartreuse	#7fff00	rgb(127,255,0)	darkorange	#ff8c00	rgb(255,140,0)
chocolate	#d2691e	rgb(210,105,30)	darkorchid	#9932cc	rgb(153,50,204)

Color Index (2)

name	hex	RGB	name	hex	RGB
darkred	#8b0000	rgb(139,0,0)	fuchsia	#ff00ff	rgb(255,0,255)
darksalmon	#e9967a	rgb(233,150,122)	gainsboro	#dcdcdc	rgb(220,220,220)
darkseagreen	#8fbcbf	rgb(143,188,143)	ghostwhite	#f8f8ff	rgb(248,248,255)
darkslateblue	#483d8b	rgb(72,61,139)	gold	#ffd700	rgb(255,215,0)
darkslategray	#2f4f4f	rgb(47,79,79)	goldenrod	#daa520	rgb(218,165,32)
darkslategrey	#2f4f4f	rgb(47,79,79)	gray	#808080	rgb(128,128,128)
darkturquoise	#00ced1	rgb(0,206,209)	green	#008000	rgb(0,128,0)
darkviolet	#9400d3	rgb(148,0,211)	greenyellow	#adff2f	rgb(173,255,47)
deeppink	#ff1493	rgb(255,20,147)	grey	#808080	rgb(128,128,128)
deepskyblue	#00bfff	rgb(0,191,255)	honeydew	#f0ffff	rgb(240,255,240)
dimgray	#696969	rgb(105,105,105)	hotpink	#ff69b4	rgb(255,105,180)
dimgrey	#696969	rgb(105,105,105)	indianred	#cd5c5c	rgb(205,92,92)
dodgerblue	#1e90ff	rgb(30,144,255)	indigo	#4b0082	rgb(75,0,130)
firebrick	#b22222	rgb(178,34,34)	ivory	#fffff0	rgb(255,255,240)
floralwhite	#ffffaf	rgb(255,250,240)	khaki	#f0e68c	rgb(240,230,140)
forestgreen	#228b22	rgb(34,139,34)	lavender	#e6e6fa	rgb(230,230,250)

Color Index (3)

name	hex	RGB	name	hex	RGB
lavenderblush	#fff0f5	rgb(255, 240, 245)	lightsteelblue	#b0c4de	rgb(176, 196, 222)
lawngreen	#7cfcc0	rgb(124, 252, 0)	lightyellow	#ffffe0	rgb(255, 255, 224)
lemonchiffon	#ffffacd	rgb(255, 250, 205)	lime	#00ff00	rgb(0, 255, 0)
lightblue	#add8e6	rgb(173, 216, 230)	limegreen	#32cd32	rgb(50, 205, 50)
lightcoral	#f08080	rgb(240, 128, 128)	linen	#faf0e6	rgb(250, 240, 230)
lightcyan	#e0ffff	rgb(224, 255, 255)	magenta	#ff00ff	rgb(255, 0, 255)
lightgoldenrodyellow	#fafad2	rgb(250, 250, 210)	maroon	#800000	rgb(128, 0, 0)
lightgray	#d3d3d3	rgb(211, 211, 211)	mediumaquamarine	#66cdaa	rgb(102, 205, 170)
lightgreen	#90ee90	rgb(144, 238, 144)	mediumblue	#0000cd	rgb(0, 0, 205)
lightgrey	#d3d3d3	rgb(211, 211, 211)	mediumorchid	#ba55d3	rgb(186, 85, 211)
lightpink	#ffb6c1	rgb(255, 182, 193)	mediumpurple	#9370db	rgb(147, 112, 219)
lightsalmon	#ffa07a	rgb(255, 160, 122)	mediumseagreen	#3cb371	rgb(60, 179, 113)
lightseagreen	#20b2aa	rgb(32, 178, 170)	mediumslateblue	#7b68ee	rgb(123, 104, 238)
lightskyblue	#87cefa	rgb(135, 206, 250)	mediumspringgreen	#00fa9a	rgb(0, 250, 154)
lightslategray	#778899	rgb(119, 136, 153)	mediumturquoise	#48d1cc	rgb(72, 209, 204)
lightslategrey	#778899	rgb(119, 136, 153)	mediumvioletred	#c71585	rgb(199, 21, 133)

Color Index (4)

name	hex	RGB	name	hex	RGB
midnightblue	#191970	rgb(25,25,112)	papayawhip	#ffefd5	rgb(255,239,213)
mintcream	#f5ffff	rgb(245,255,250)	peachpuff	#ffdab9	rgb(255,218,185)
mistyrose	#ffe4e1	rgb(255,228,225)	peru	#cd853f	rgb(205,133,63)
moccasin	#ffe4b5	rgb(255,228,181)	pink	#ffc0cb	rgb(255,192,203)
navajowhite	#ffdead	rgb(255,222,173)	plum	#dda0dd	rgb(221,160,221)
navy	#000080	rgb(0,0,128)	powderblue	#b0e0e6	rgb(176,224,230)
oldlace	#fdf5e6	rgb(253,245,230)	purple	#800080	rgb(128,0,128)
olive	#808000	rgb(128,128,0)	red	#ff0000	rgb(255,0,0)
olivedrab	#6b8e23	rgb(107,142,35)	rosybrown	#bc8f8f	rgb(188,143,143)
orange	#ffa500	rgb(255,165,0)	royalblue	#4169e1	rgb(65,105,225)
orangered	#ff4500	rgb(255,69,0)	saddlebrown	#8b4513	rgb(139,69,19)
orchid	#da70d6	rgb(218,112,214)	salmon	#fa8072	rgb(250,128,114)
palegoldenrod	#eee8aa	rgb(238,232,170)	sandybrown	#f4a460	rgb(244,164,96)
palegreen	#98fb98	rgb(152,251,152)	seagreen	#2e8b57	rgb(46,139,87)
paleturquoise	#afeeee	rgb(175,238,238)	seashell	#fff5ee	rgb(255,245,238)
palevioletred	#db7093	rgb(219,112,147)	sienna	#a0522d	rgb(160,82,45)

Color Index (5)

name	hex	RGB	name	hex	RGB
silver	#c0c0c0	rgb(192,192,192)	whitesmoke	#f5f5f5	rgb(245,245,245)
skyblue	#87ceeb	rgb(135,206,235)	yellow	#ffff00	rgb(255,255,0)
slateblue	#6a5acd	rgb(106,90,205)	yellowgreen	#9acd32	rgb(154,205,50)
slategray	#708090	rgb(112,128,144)			
slategrey	#708090	rgb(112,128,144)			
snow	#fffafa	rgb(255,250,250)			
springgreen	#00ff7f	rgb(0,255,127)			
steelblue	#4682b4	rgb(70,130,180)			
tan	#d2b48c	rgb(210,180,140)			
teal	#008080	rgb(0,128,128)			
thistle	#d8bfd8	rgb(216,191,216)			
tomato	#ff6347	rgb(255,99,71)			
turquoise	#40e0d0	rgb(64,224,208)			
violet	#ee82ee	rgb(238,130,238)			
wheat	#f5deb3	rgb(245,222,179)			
white	#ffffff	rgb(255,255,255)			

Neutrals

name	hex	RGB	name	hex	RGB
aliceblue	#f0f8ff	rgb(240, 248, 255)	dimgray	#696969	rgb(105, 105, 105)
antiquewhite	#faebd7	rgb(250, 235, 215)	dimgrey	#696969	rgb(105, 105, 105)
azure	#f0ffff	rgb(240, 255, 255)	floralwhite	#fffaf0	rgb(255, 250, 240)
beige	#f5f5dc	rgb(245, 245, 220)	gainsboro	#dcdcdc	rgb(220, 220, 220)
bisque	#ffe4c4	rgb(255, 228, 196)	ghostwhite	#f8f8ff	rgb(248, 248, 255)
black	#000000	rgb(0, 0, 0)	gray	#808080	rgb(128, 128, 128)
blanchedalmond	#ffebcd	rgb(255, 235, 205)	grey	#808080	rgb(128, 128, 128)
brown	#a52a2a	rgb(165, 42, 42)	honeydew	#f0fff0	rgb(240, 255, 240)
burlywood	#deb887	rgb(222, 184, 135)	ivory	#fffff0	rgb(255, 255, 240)
chocolate	#d2691e	rgb(210, 105, 30)	lavender	#e6e6fa	rgb(230, 230, 250)
cornsilk	#fff8dc	rgb(255, 248, 220)	lavenderblush	#fff0f5	rgb(255, 240, 245)
darkgray	#a9a9a9	rgb(169, 169, 169)	lightgray	#d3d3d3	rgb(211, 211, 211)
darkgrey	#a9a9a9	rgb(169, 169, 169)	lightgrey	#d3d3d3	rgb(211, 211, 211)
darksalmon	#e9967a	rgb(233, 150, 122)	lightslategray	#778899	rgb(119, 136, 153)
darkslategray	#2f4f4f	rgb(47, 79, 79)	lightslategrey	#778899	rgb(119, 136, 153)
darkslategrey	#2f4f4f	rgb(47, 79, 79)	linen	#faf0e6	rgb(250, 240, 230)

Neutrals (2)

name	hex	RGB	name	hex	RGB
mintcream	#f5ffff	rgb(245, 255, 250)	slategrey	#708090	rgb(112, 128, 144)
mistyrose	#ffe4e1	rgb(255, 228, 225)	snow	#fffafa	rgb(255, 250, 250)
moccasin	#ffe4b5	rgb(255, 228, 181)	tan	#d2b48c	rgb(210, 180, 140)
navajowhite	#ffdead	rgb(255, 222, 173)	wheat	#f5deb3	rgb(245, 222, 179)
oldlace	#fdf5e6	rgb(253, 245, 230)	white	#ffffff	rgb(255, 255, 255)
papayawhip	#ffefdf	rgb(255, 239, 213)	whitesmoke	#f5f5f5	rgb(245, 245, 245)
peachpuff	#ffdab9	rgb(255, 218, 185)			
peru	#cd853f	rgb(205, 133, 63)			
rosybrown	#bc8f8f	rgb(188, 143, 143)			
saddlebrown	#8b4513	rgb(139, 69, 19)			
salmon	#fa8072	rgb(250, 128, 114)			
sandybrown	#f4a460	rgb(244, 164, 96)			
seashell	#ffff5ee	rgb(255, 245, 238)			
sienna	#a0522d	rgb(160, 82, 45)			
silver	#c0c0c0	rgb(192, 192, 192)			
slategray	#708090	rgb(112, 128, 144)			

Reds

name	hex	RGB	name	hex	RGB
coral	#ff7f50	rgb(255,127,80)	palevioletred	#db7093	rgb(219,112,147)
crimson	#dc143c	rgb(220,20,60)	pink	#ffc0cb	rgb(255,192,203)
darkmagenta	#8b008b	rgb(139,0,139)	plum	#dda0dd	rgb(221,160,221)
darkred	#8b0000	rgb(139,0,0)	red	#ff0000	rgb(255,0,0)
deeppink	#ff1493	rgb(255,20,147)	thistle	#d8bfd8	rgb(216,191,216)
firebrick	#b22222	rgb(178,34,34)	tomato	#ff6347	rgb(255,99,71)
fuchsia	#ff00ff	rgb(255,0,255)			
hotpink	#ff69b4	rgb(255,105,180)			
indianred	#cd5c5c	rgb(205,92,92)			
lightcoral	#f08080	rgb(240,128,128)			
lightpink	#ffb6c1	rgb(255,182,193)			
lightsalmon	#ffa07a	rgb(255,160,122)			
magenta	#ff00ff	rgb(255,0,255)			
maroon	#800000	rgb(128,0,0)			
orangered	#ff4500	rgb(255,69,0)			
orchid	#da70d6	rgb(218,112,214)			

Greens

name	hex	RGB	name	hex	RGB
aquamarine	#7ffffd4	rgb(127,255,212)	olive	#808000	rgb(128,128,0)
chartreuse	#7fff00	rgb(127,255,0)	olivedrab	#6b8e23	rgb(107,142,35)
darkgreen	#006400	rgb(0,100,0)	palegreen	#98fb98	rgb(152,251,152)
darkkhaki	#bdb76b	rgb(189,183,107)	seagreen	#2e8b57	rgb(46,139,87)
darkolivegreen	#556b2f	rgb(85,107,47)	springgreen	#00ff7f	rgb(0,255,127)
darkseagreen	#8fbcb8	rgb(143,188,143)	teal	#008080	rgb(0,128,128)
forestgreen	#228b22	rgb(34,139,34)	yellowgreen	#9acd32	rgb(154,205,50)
green	#008000	rgb(0,128,0)			
greenyellow	#adff2f	rgb(173,255,47)			
lawngreen	#7cfcc00	rgb(124,252,0)			
lightgreen	#90ee90	rgb(144,238,144)			
lightseagreen	#20b2aa	rgb(32,178,170)			
lime	#00ff00	rgb(0,255,0)			
limegreen	#32cd32	rgb(50,205,50)			
mediumseagreen	#3cb371	rgb(60,179,113)			
mediumspringgreen	#00fa9a	rgb(0,250,154)			

Blues

name	hex	RGB	name	hex	RGB
aqua	#00ffff	rgb(0, 255, 255)	mediumblue	#0000cd	rgb(0, 0, 205)
blue	#0000ff	rgb(0, 0, 255)	mediumslateblue	#7b68ee	rgb(123, 104, 238)
cadetblue	#5f9ea0	rgb(95, 158, 160)	mediumturquoise	#48d1cc	rgb(72, 209, 204)
cornflowerblue	#6495ed	rgb(100, 149, 237)	midnightblue	#191970	rgb(25, 25, 112)
cyan	#00ffff	rgb(0, 255, 255)	navy	#000080	rgb(0, 0, 128)
darkblue	#00008b	rgb(0, 0, 139)	paleturquoise	#afeeee	rgb(175, 238, 238)
darkcyan	#008b8b	rgb(0, 139, 139)	powderblue	#b0e0e6	rgb(176, 224, 230)
darkslateblue	#483d8b	rgb(72, 61, 139)	royalblue	#4169e1	rgb(65, 105, 225)
darkturquoise	#00ced1	rgb(0, 206, 209)	skyblue	#87ceeb	rgb(135, 206, 235)
deepskyblue	#00bfff	rgb(0, 191, 255)	slateblue	#6a5acd	rgb(106, 90, 205)
dodgerblue	#1e90ff	rgb(30, 144, 255)	steelblue	#4682b4	rgb(70, 130, 180)
lightblue	#add8e6	rgb(173, 216, 230)	turquoise	#40e0d0	rgb(64, 224, 208)
lightcyan	#e0ffff	rgb(224, 255, 255)			
lightskyblue	#87cefa	rgb(135, 206, 250)			
lightsteelblue	#b0c4de	rgb(176, 196, 222)			
mediumaquamarine	#66cdaa	rgb(102, 205, 170)			

Violets

	name	hex	RGB
	blueviolet	#8a2be2	rgb(138, 43, 226)
	darkorchid	#9932cc	rgb(153, 50, 204)
	darkviolet	#9400d3	rgb(148, 0, 211)
	indigo	#4b0082	rgb(75, 0, 130)
	mediumorchid	#ba55d3	rgb(186, 85, 211)
	mediumpurple	#9370db	rgb(147, 112, 219)
	mediumvioletred	#c71585	rgb(199, 21, 133)
	purple	#800080	rgb(128, 0, 128)
	violet	#ee82ee	rgb(238, 130, 238)

Yellows

name	hex	RGB
darkgoldenrod	#b8860b	rgb(184,134,11)
darkorange	#ff8c00	rgb(255,140,0)
gold	#ffd700	rgb(255,215,0)
goldenrod	#daa520	rgb(218,165,32)
khaki	#f0e68c	rgb(240,230,140)
lemonchiffon	#ffffacd	rgb(255,250,205)
lightgoldenrodyellow	#fafad2	rgb(250,250,210)
lightyellow	#ffffe0	rgb(255,255,224)
orange	#ffa500	rgb(255,165,0)
palegoldenrod	#eee8aa	rgb(238,232,170)
yellow	#ffff00	rgb(255,255,0)

(b)text ctext etext rtext arctext textbox textfile textcode line

begin center end

rotate

hello,
world

Now is the time
for all good men
to come

This is the contents
of a file. it has lines of text.
Reading is fundamental.

```
import "fmt"  
func main() {  
    fmt.Println("Go")  
}
```



hline vline arc curve polyline circle acircle ellipse square

rect rrect pill polygon star image cimage lbrace rbrace



ubrace dbrace lbracket rbracket ubracket dbracket sky arrow larrow rarrow



dcarrow uarrow list blist nlist clist

one
two
three

- one
- two
- three

1. one
2. two
3. three

first
second item
third

decksh version
2025-12-22-1.0.0

Textual Elements

description	keyword	mandatory	optional
Left-aligned	<code>text</code>	"..." x y <code>fontsize</code>	font color op link
Centered	<code>ctext</code>	"..." x y <code>fontsize</code>	font color op link
End-aligned	<code>etext</code>	"..." x y <code>fontsize</code>	font color op link
Rotated	<code>rtext</code>	"..." x y <code>angle fontsize</code>	font color op link
Text on an arc	<code>arctext</code>	"..." x y <code>rad a1 a2 fontsize</code>	font color op link
Block text	<code>textblock</code>	"..." x y <code>w fontsize</code>	font color op link
Block text from file	<code>textblockfile</code>	"file" x y <code>w fontsize</code>	font color op link
File contents	<code>textfile</code>	"file" x y <code>fontsize</code>	font color op spacing
Code listing	<code>textcode</code>	"file" x y <code>w fontsize</code>	font color



hello, world
(x,y)

```
text "..." x y fontsize font color op link
```

abc

```
text "abc" 20 20 4
```

abc

```
text "abc" 75 20 7 "mono" "maroon"
```

hello,
world
(x,y)

```
ctext "... " x y fontsize font color op link
```

abc

```
ctext "abc" 20 20 4
```

abc

```
ctext "abc" 80 20 7 "mono" "maroon"
```

hello, world
•
(x, y)

```
etext "..." x y fontsize font color op link
```

abc

```
etext "abc" 20 20 4
```

abc

```
etext "abc" 80 20 7 "mono" "maroon"
```

hello, world

(x, y)

```
rtext "..." x y angle fontsize font color op link
```

abc

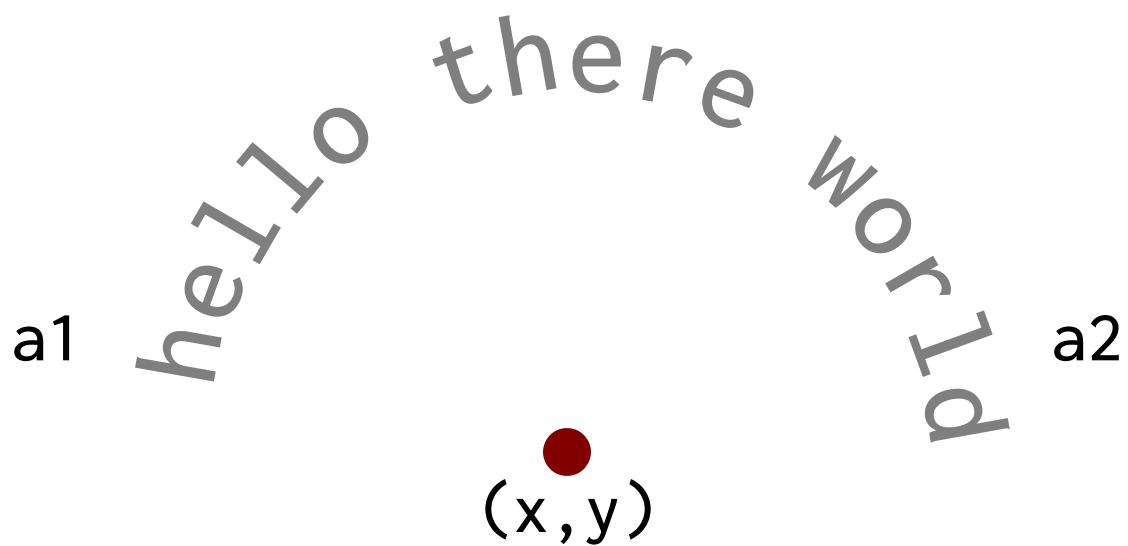
```
rtext 20 20 30 3
```

abc

```
rtext 50 20 90 5
```

abc

```
rtext 80 20 270 4 "sans" "maroon"
```



```
arctext "... " x y radius a1 a2 fontsize font color op link
```

Two examples of curved text. On the left, the text "What is up" is curved upwards from left to right. On the right, the text "This is curvy" is curved downwards from left to right.

```
arctext "What is up" 25 20 10 180 90 3 "mono"    arctext "This is curvy" 75 30 10 180 360 3 "mono"
```

(x,y) “Where justice is denied, where poverty is enforced,
where ignorance prevails, and where any one class
is made to feel that society is an organized conspiracy
to oppress, rob and degrade them, neither persons
nor property will be safe.”

w

textblock "..." x y w fontsize font color op link

“Where justice is denied, where poverty is enforced,
where ignorance prevails, and where any one class
is made to feel that society is an organized conspiracy
to oppress, rob and degrade them, neither persons
nor property will be safe.”

“Where justice is denied,
where poverty is enforced,
where ignorance prevails,
and where any one class is
made to feel that society
is an organized conspiracy
to oppress, rob and degrade
them, neither persons nor
property will be safe.”

textblock "..." 10 35 30 2

textblock "..." 50 35 10 1 "sans" "maroon"

(x,y) This is the contents
of a file. it has lines of text.
Reading is fundamental.

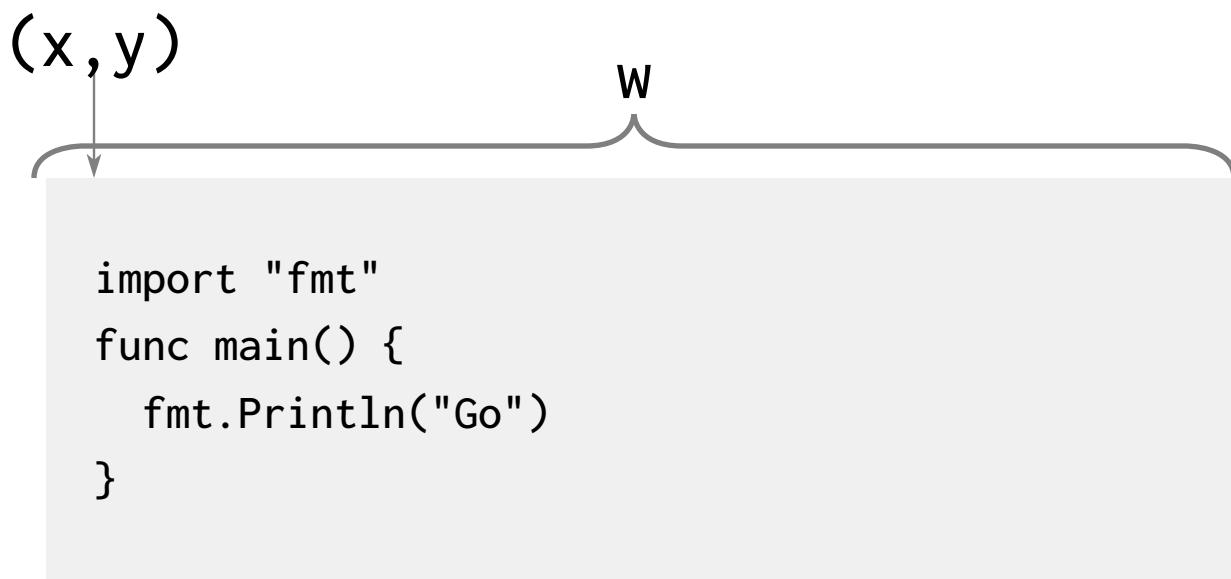
textfile "..." x y fontsize font color op link

This is the contents
of a file. it has lines of text.
Reading is fundamental.

```
import "fmt"
func main() {
    fmt.Println("Go")
}
```

textfile "example.txt" 10 35 2

textfile "hw-go" 55 35 1.6 "mono" "maroon"



textcode "..." **x** **y** **w** **fontsize** **font** **color**

```
import "fmt"
func main() {
    fmt.Println("Go")
}
```

```
import "fmt"
func main() {
    fmt.Println("Go")
}
```

textcode "hw-go" 10 35 25 1.0

textcode "hw-go" 55 35 40 1.6 "maroon"

Graphical Elements

description	keyword	mandatory	optional
Line	line	x1 y1 x2 y2 lw	color op
Horizontal line	hline	x y w	lw color op
Vertical line	vline	x y h	lw color op
Elliptical arc	arc	x y w h a1 a2	lw color op
Quadratic Bezier	curve	bx by cx cy ex ey	lw color op
Circle	circle	x y w	color op
Area circle	acircle	x y area	color op
Ellipse	ellipse	x y w h	color op
Square	square	x y w	color op
Rectangle	rect	x y w h	color op
Rounded rectangle	rrect	x y w h radius	color
Pill shape	pill	x y w h	color
Polygon	polygon	"x1 x2...xn" "y1 y2...yn"	color op
Polyline	polyline	"x1 x2...xn" "y1 y2...yn"	lw color op
N-sided star	star	x y sides inner outer	color op

`lw {`

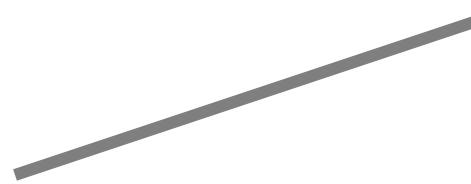
`(x1,y1)`



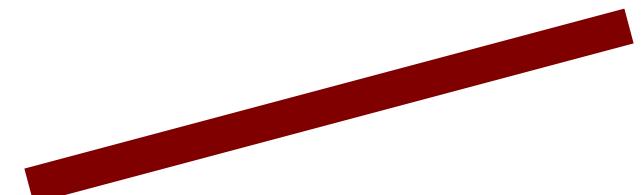
`(x2,y2)`

`line x1 y1 x2 y2 lw color op`

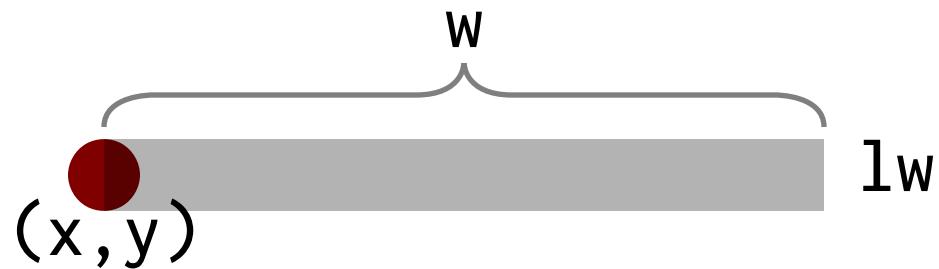
`line 10 20 30 20`



`line 40 20 60 30 0.5`



`line 70 20 95 30 1.5 "maroon"`



hline x y w lw color op

—

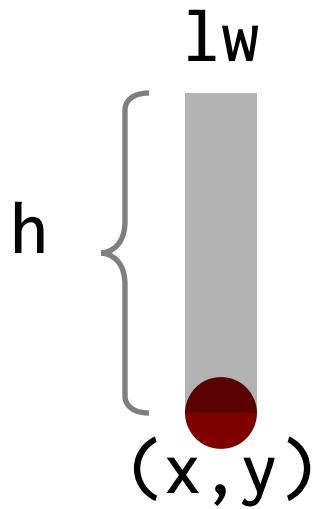
hline 15 20 10

—

hline 40 20 20 1



hline 70 20 20 5 "maroon" 20



vline x y w lw color op



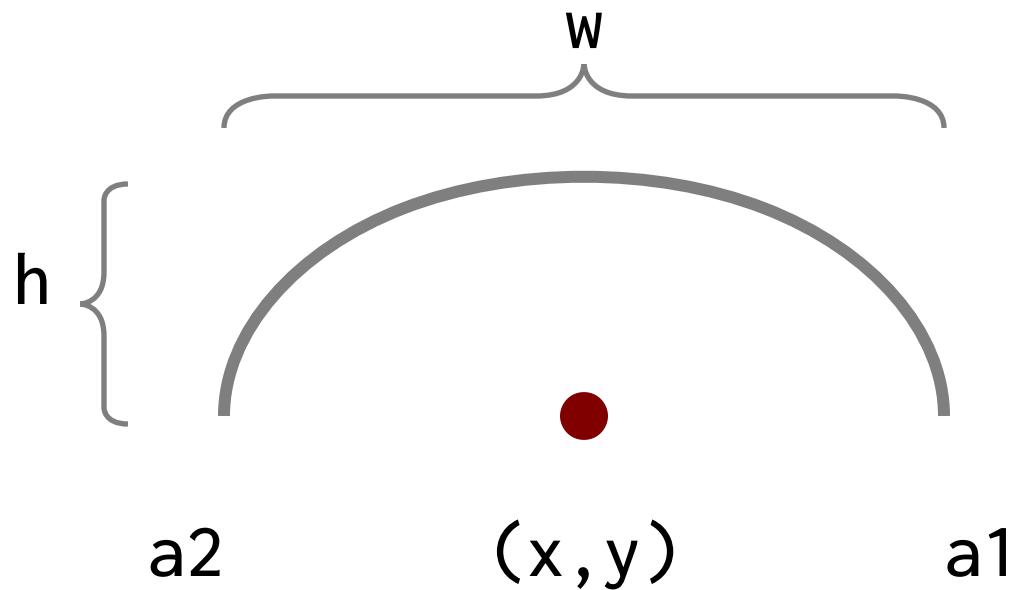
vline 20 20 15



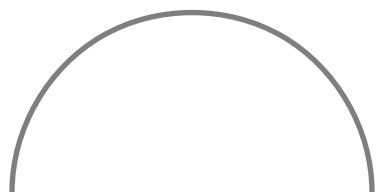
vline 50 20 15 2



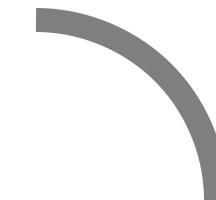
vline 80 20 15 10 "maroon" 20



arc **x** **y** **w** **h** **a1** **a2** **lw** **color** **op**



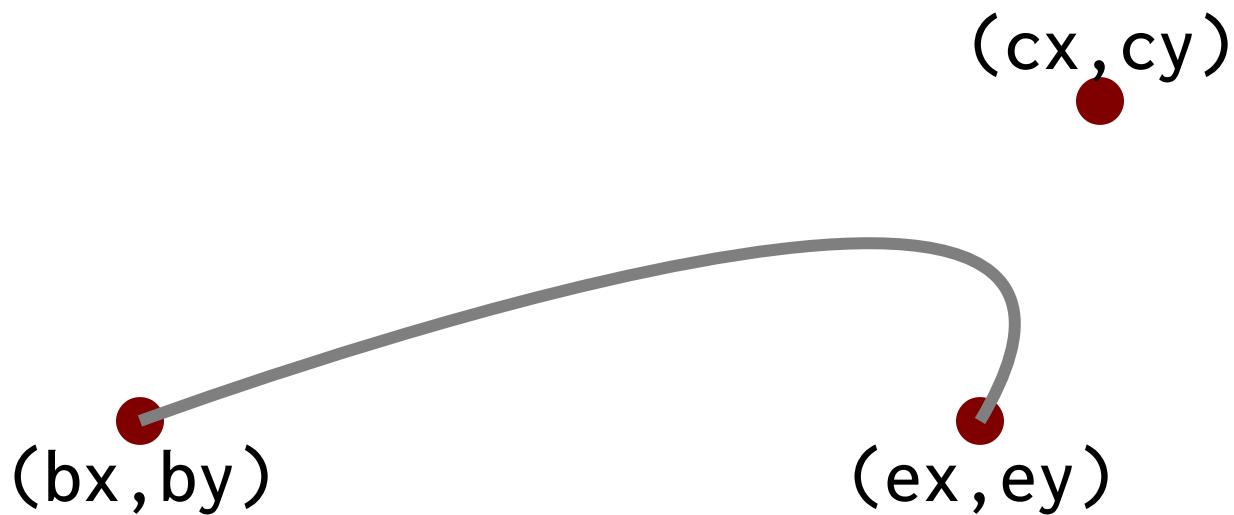
arc 20 20 15 15 0 180



arc 50 20 15 15 0 90 1



arc 80 20 5 5 0 180 5 "maroon"



curve bx by cx cy ex ey lw color op



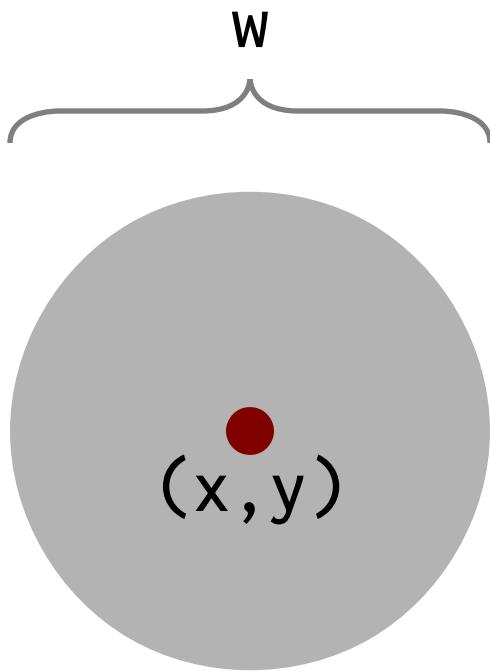
curve 15 20 25 30 25



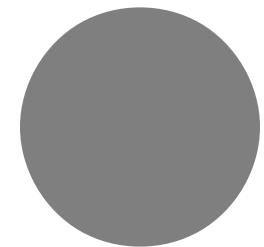
curve 15 20 25 30 25



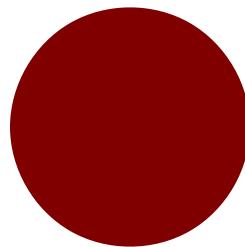
curve 70 20 70 30 90 25 0.5 "maroon"



circle x y w color op



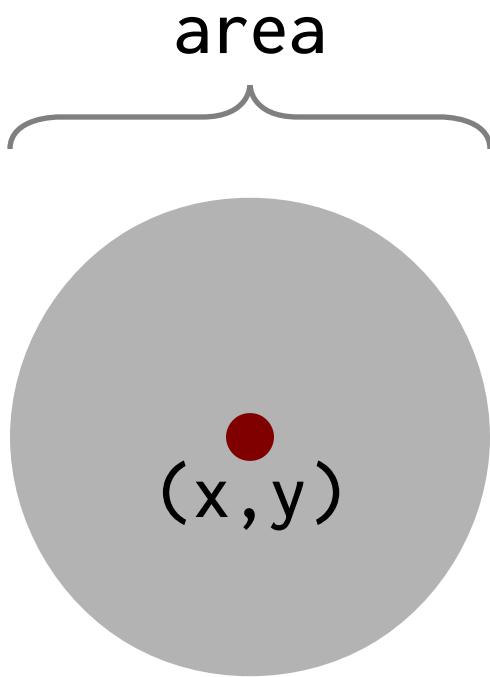
circle 20 20 10



circle 50 20 10 "maroon"



circle 80 20 5 "maroon" 20



circle x y area color op



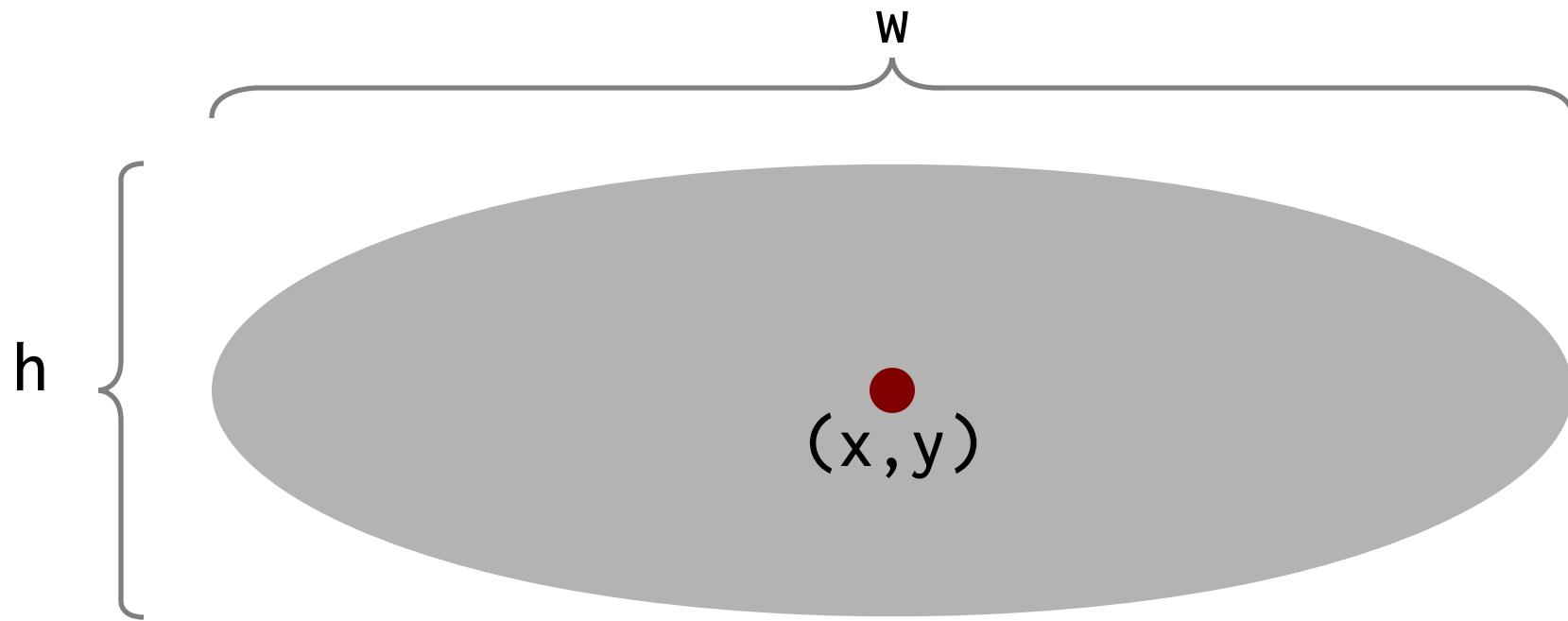
acircle 20 20 10



acircle 50 20 10 "maroon"



acircle 80 20 5 "maroon" 20



ellipse x y w h color op



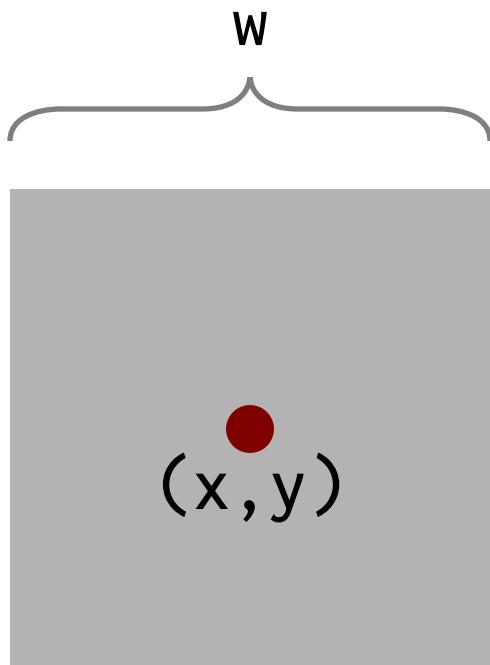
ellipse 20 20 10 5



ellipse 50 20 10 5 "maroon"



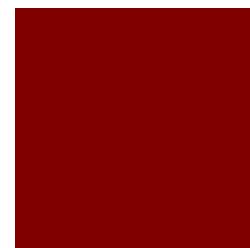
ellipse 80 20 5 10 "maroon" 20



square x y w color op



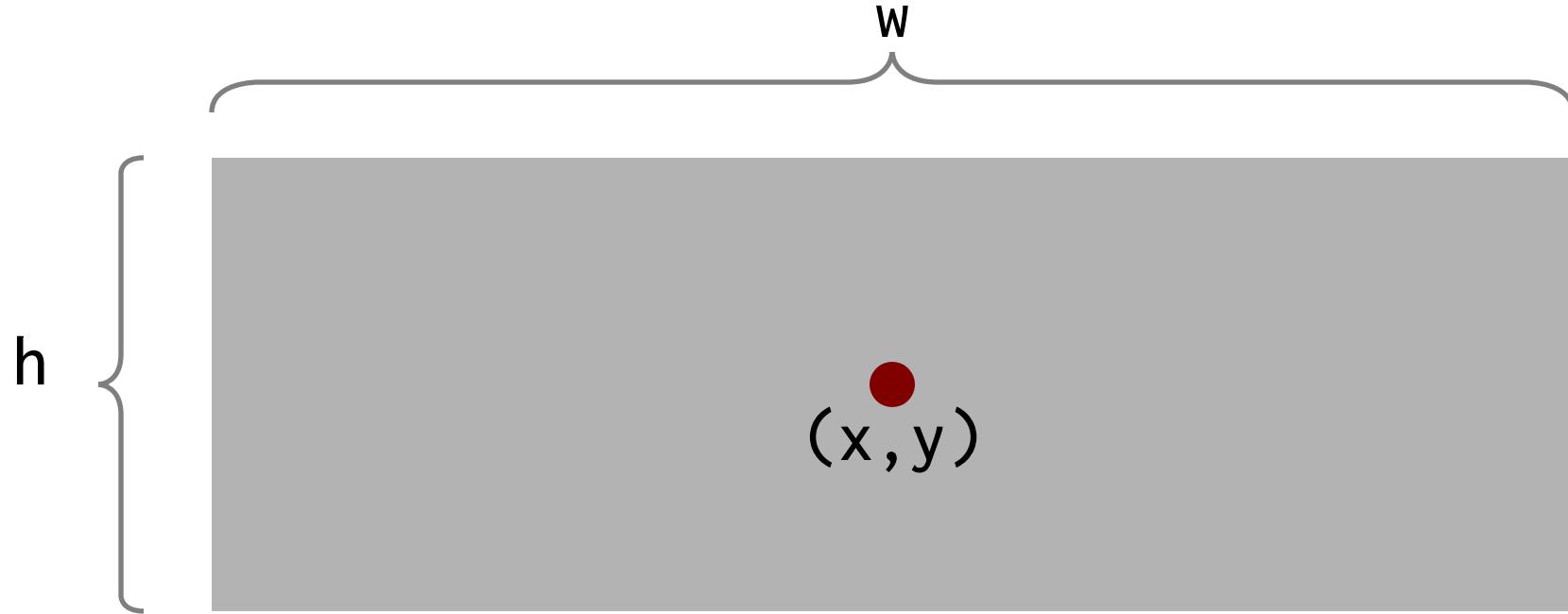
square 20 20 10



square 50 20 10 "maroon"



square 80 20 5 "maroon" 20



rect x y w h color op



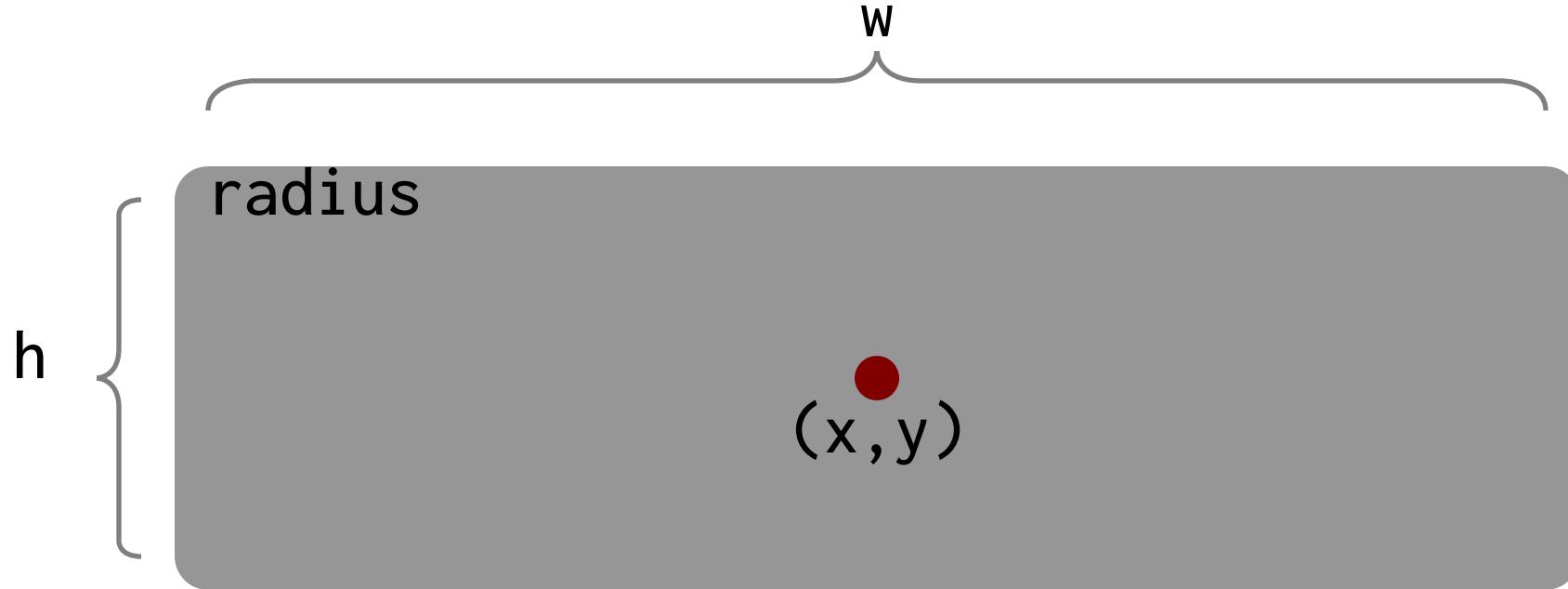
rect 20 20 10 5



rect 50 20 10 5 "maroon"



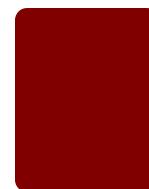
rect 80 20 5 10 "maroon" 20



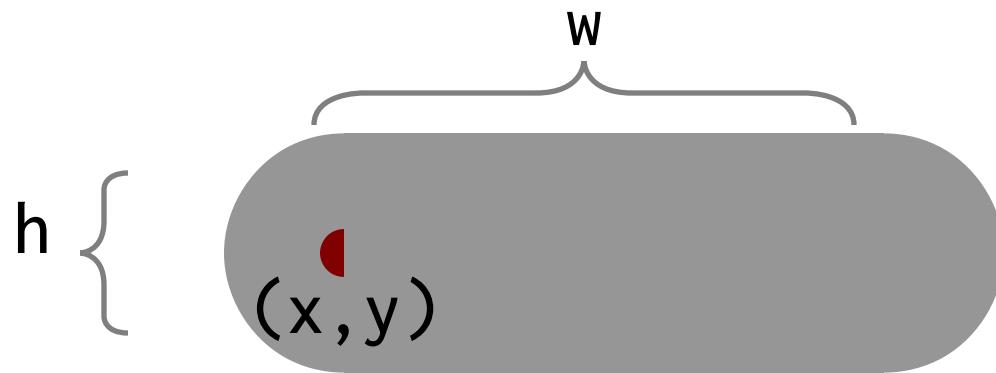
rrect x y w h radius color op



rrect 20 20 10 5 1



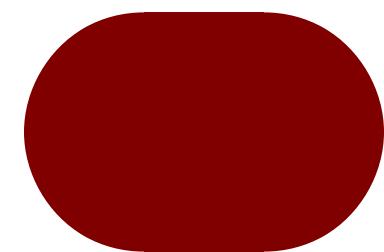
rrect 80 20 5 10 1 "maroon"



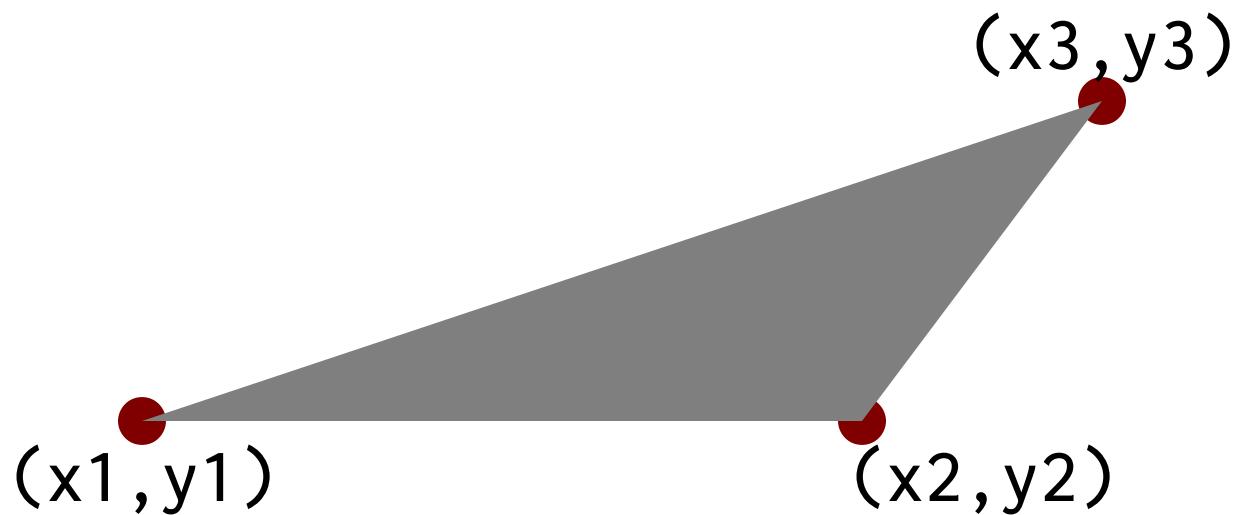
pill x y w h color



pill 20 20 10 5



pill 80 20 5 10 "maroon"



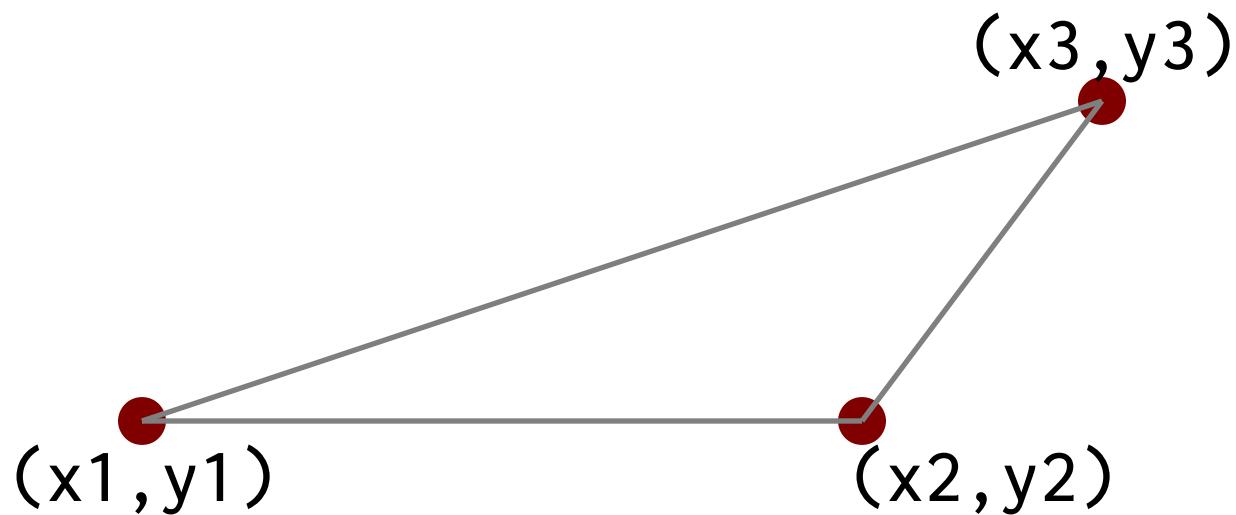
```
polygon "x1 x2...xn" "y1 y2...yn" color op
```



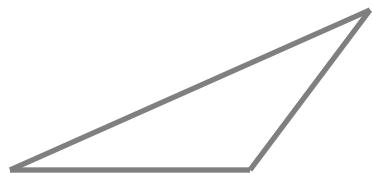
```
polygon "10 25 20" "20 30 20"
```



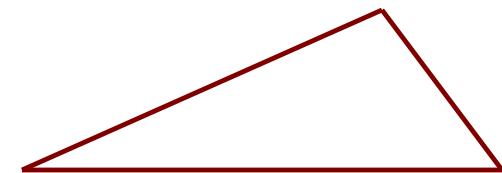
```
polygon "70 85 90" "20 30 20" "maroon"
```



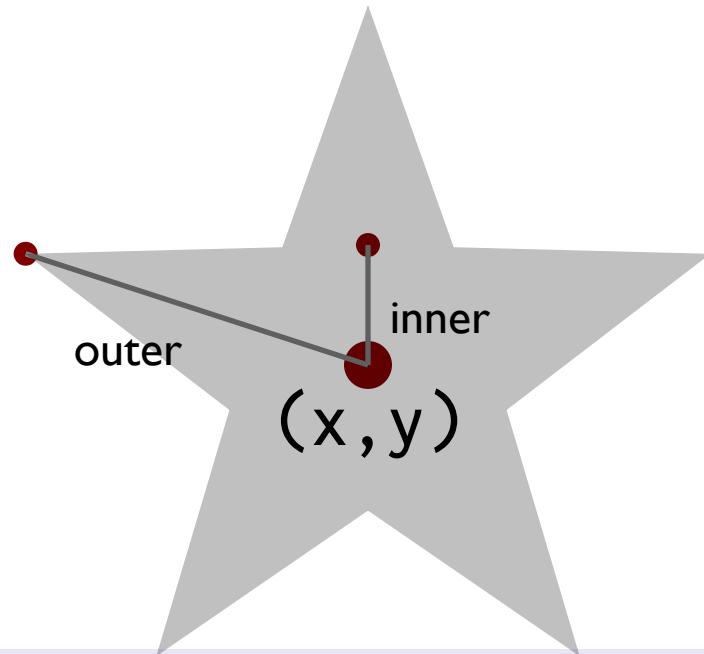
```
polyline "x1 x2...xn" "y1 y2...yn" lw color op
```



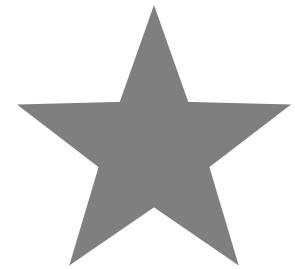
```
polyline "10 25 20" "20 30 20"
```



```
polyline "70 85 90" "20 30 20" 0.2 "maroon"
```



star x y sides inner outer color op



star 20 20 5 2 6



star 50 20 12 2 5 "maroon"



star 80 ey 24 2 8 "maroon" 20

Images

description	keyword	mandatory	optional
Image	image	"file" x y w h	scale "link"
Captioned image	cimage	"file" "caption" x y w h	scale "link" capsiz

If h = 0, w specifies the image width in terms of canvas width.

The scale value is a percentage from 1-100, and link is a URL. capsiz is the text size of the caption

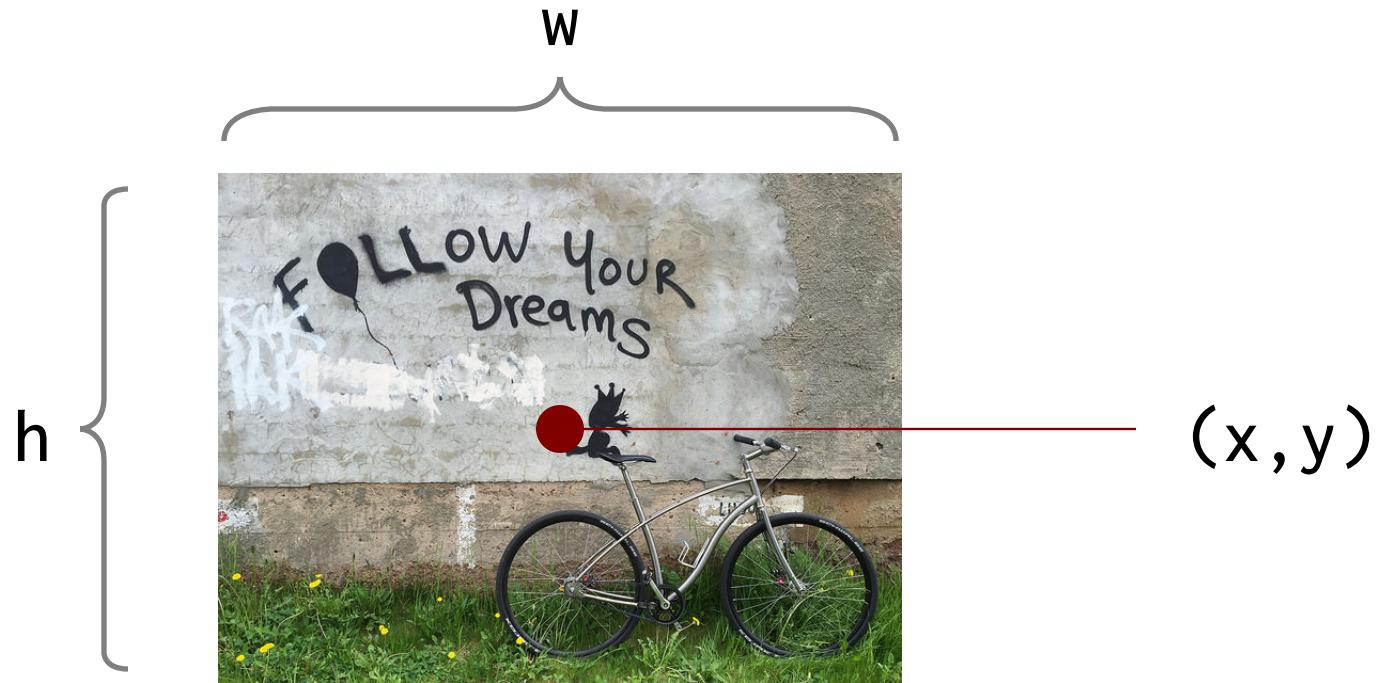


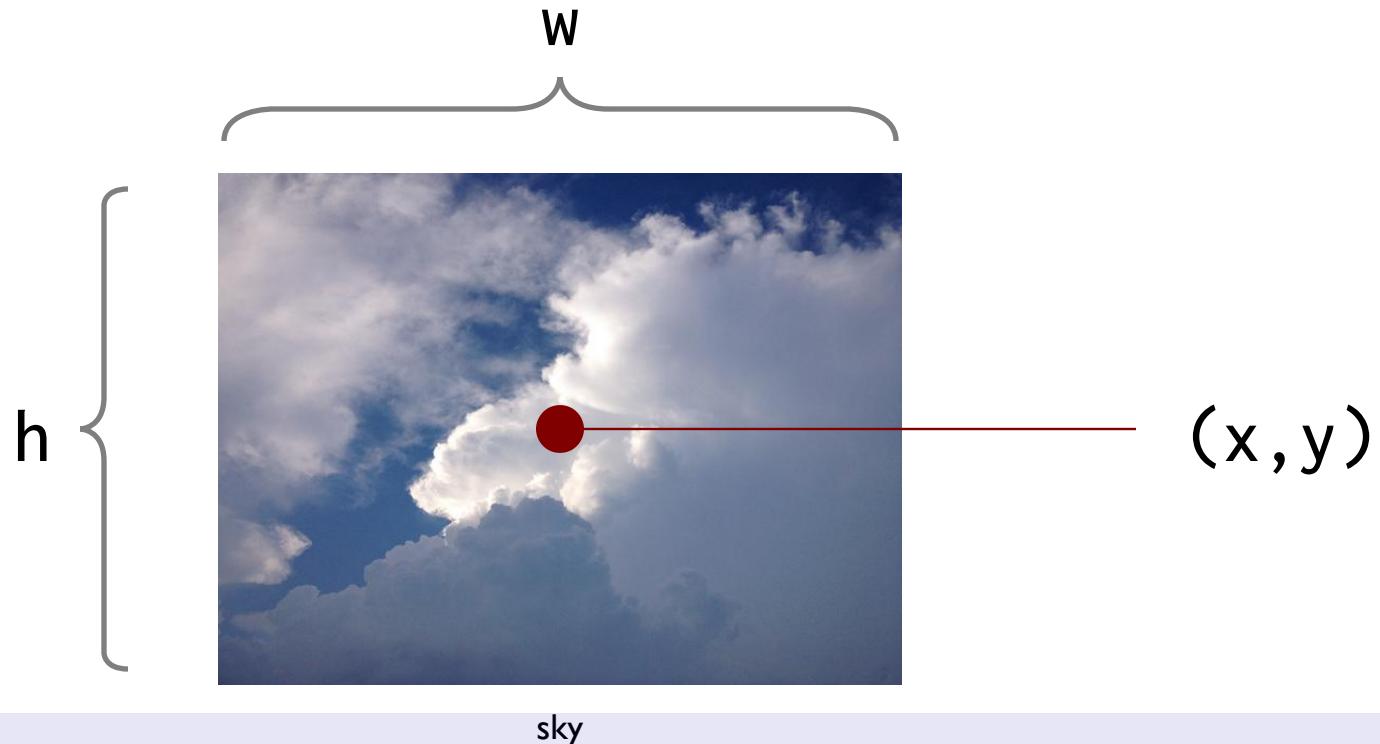
image "file" x y w h scale link



image "follow.jpg" 20 25 640 480 10



image "follow.jpg" 75 25 640 480 30



cimage "file" x y w h scale link



sky



sky

cimage "cloudy.jpg" "sky" 20 25 640 480 10

cimage "cloudy.jpg" "sky" 75 25 640 480 30 "" 1.5

Lists

description	keyword	mandatory	optional
Plain list	list	x y fontsize	font color op spacing
Bullet list	blist	x y fontsize	font color op spacing
Numbered list	nlist	x y fontsize	font color op spacing
Centered list	clist	x y fontsize	font color op spacing

list x y fs
(x,y) li "first"
li "second"
li "third"
elist

list x y fontsize font color op spacing

list 20 30 2.5 one
li "one"
li "two" two
li "three" three
elist

list 85 30 2.5 "serif" "maroon" 100 1.0 one
li "one"
li "two"
li "three" three
elist

```
blist x y fs  
(x,y) li "first"  
      li "second"  
      li "third"  
elist
```

blist x y fontsize font color op spacing

```
blist 20 30 2.5 ● one  
      li "one"  
      li "two"   ● two  
      li "three"  
elist           ● three
```

```
blist 85 30 2.5 "serif" "maroon" 100 1.0 ● one  
      li "one"  
      li "two"  
      li "three"  
elist           ● two  
                  ● three
```

```
nlist x y fs  
(x,y) li "first"  
      li "second"  
      li "third"  
elist
```

nlist x y fontsize font color op spacing

```
nlist 20 30 2.5 1. one  
      li "one"  
      li "two"   2. two  
      li "three"  
elist           3. three
```

```
nlist 85 30 2.5 "serif" "maroon" 100 1.0 1. one  
      li "one"  
      li "two"   2. two  
      li "three"  
elist           3. three
```

clist x y fs
(x,y) li "first"
li "second"
li "third"
elist

clist x y fontsize font color op spacing

clist 30 30 2.5	first one	clist 90 30 2.5 "serif" "maroon" 100 1.0	first
li "first one"		li "first"	next
li "next"	next	li "next"	and last
li "and last"	and last	li "and last"	
elist		elist	

Arrows

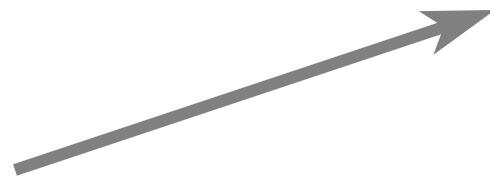
description	keyword	mandatory	optional
Straight	arrow	x1 y1 x2 y2	lw aw ah color op
Left curved	larrow	bx by cx cy ex ey	lw aw ah color op
Right curved	rarrow	bx by cx cy ex ey	lw aw ah color op
Up curved	ucarrow	bx by cx cy ex ey	lw aw ah color op
Down curved	darrow	bx by cx cy ex ey	lw aw ah color op



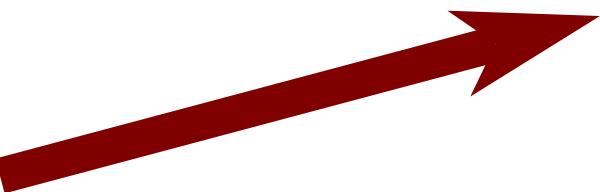
arrow x1 y1 x2 y2 lw aw ah color op



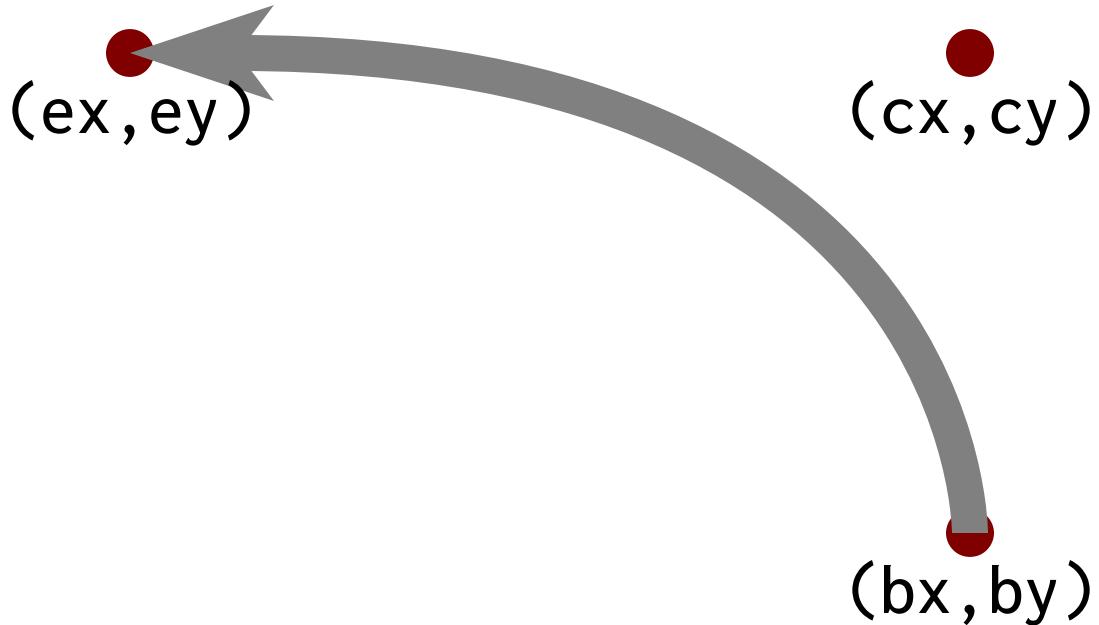
arrow 10 20 30 20



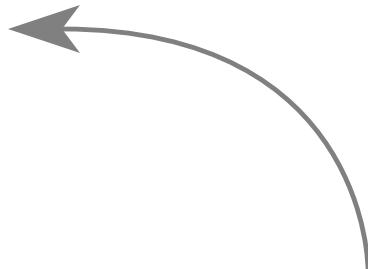
arrow 40 20 60 30 0.5



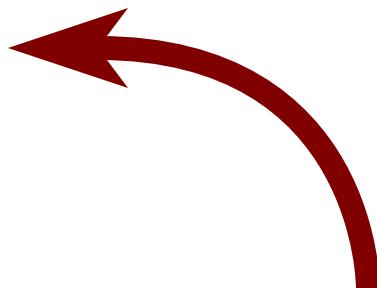
arrow 70 20 95 30 1.5 6 6 "maroon"



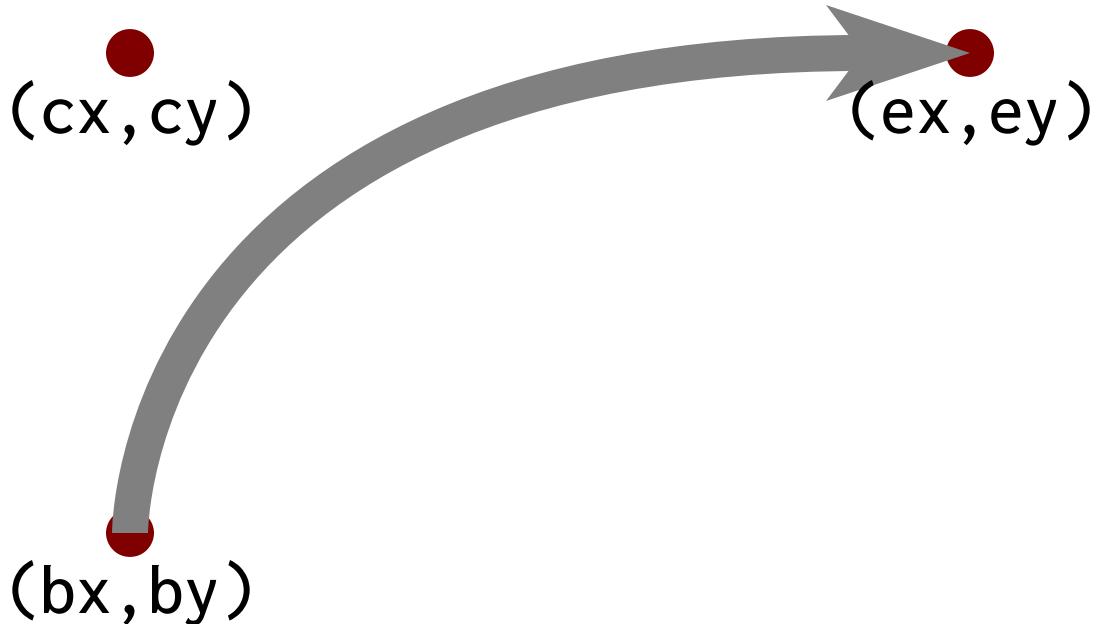
larrow bx by cx cy ex ey lw aw ah color op



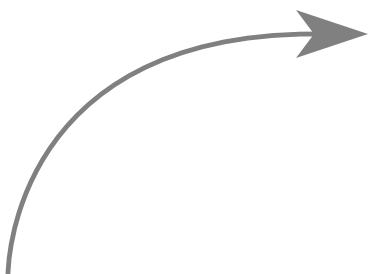
larrow 30 20 30 35 15 35



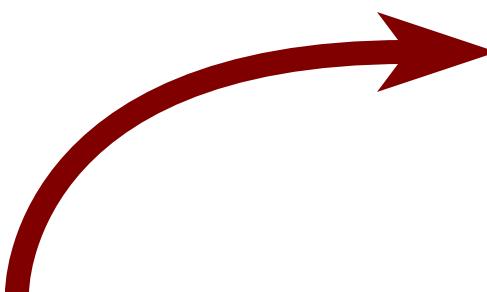
larrow 70 20 70 35 55 35 1 5 5 "maroon"



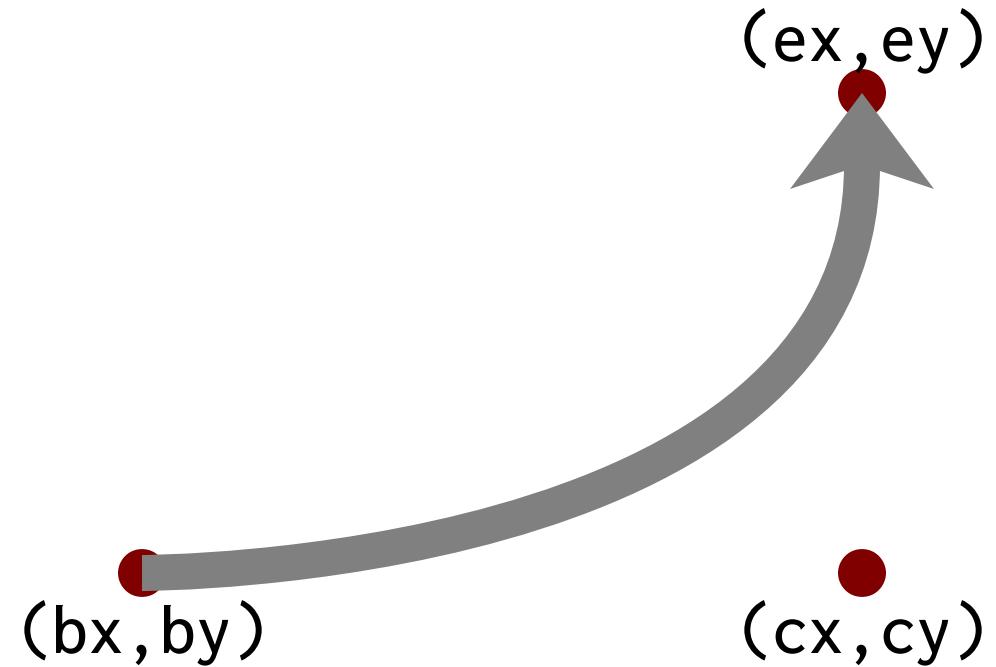
rcarrow bx by cx cy ex ey lw aw ah color op



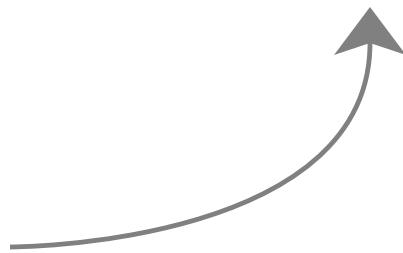
rcarrow 15 20 15 35 30 35



rcarrow 50 20 50 35 70 35 1 5 5 "maroon"



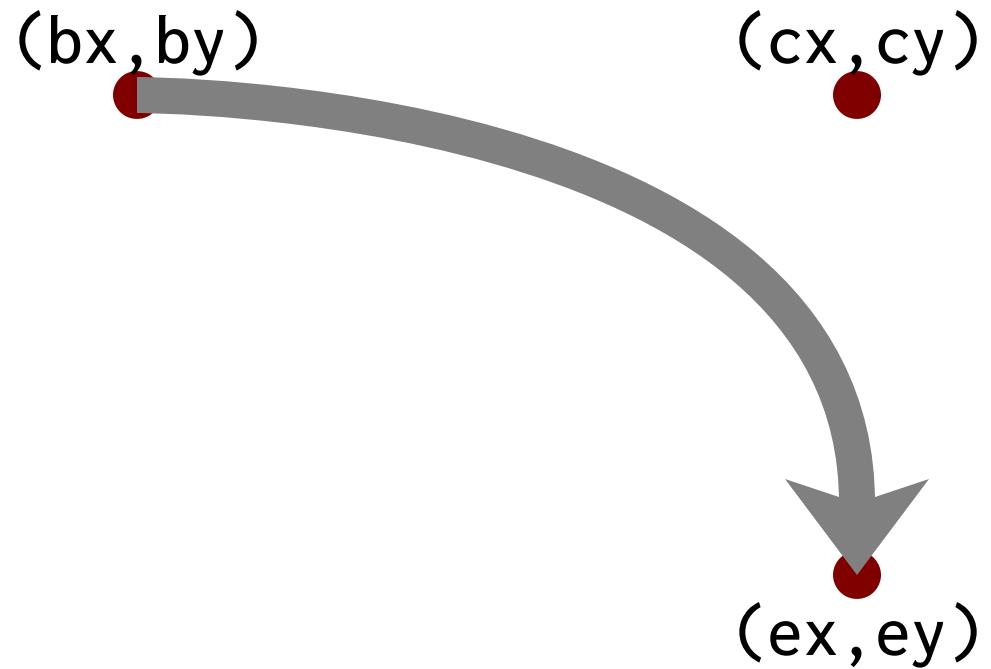
ucarrow bx by cx cy ex ey lw aw ah color op



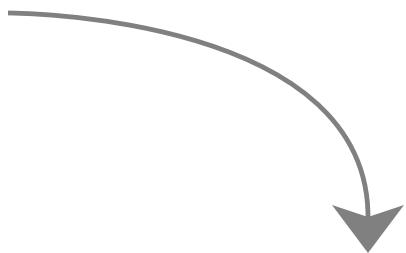
ucarrow 15 20 30 20 30 35



rcarrow 50 20 70 20 70 35 1 5 5 "maroon"



dcarrow bx by cx cy ex ey lw aw ah color op



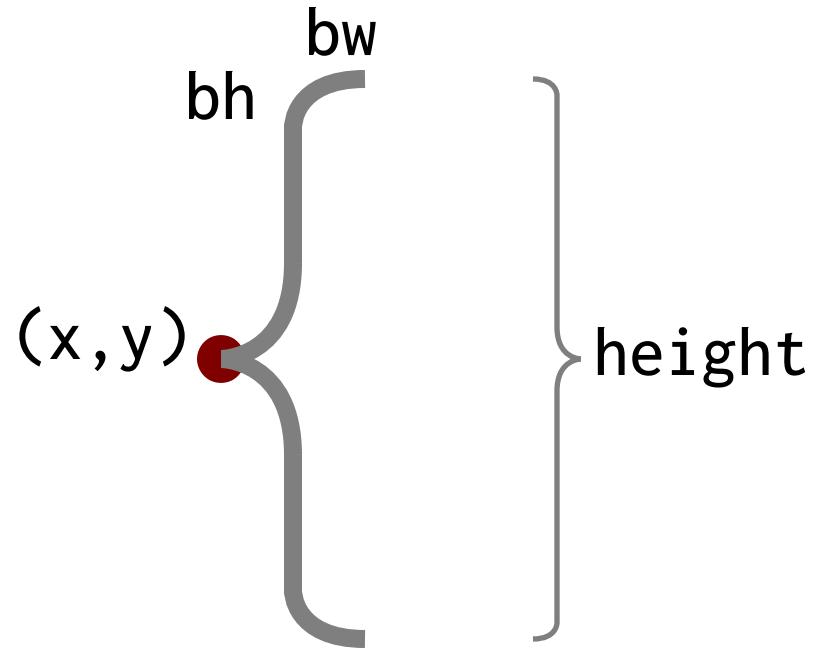
dcarrow 15 35 30 30 20



dcarrow 50 35 70 35 70 20 1 5 5 "maroon"

Braces and Brackets

description	keyword	mandatory	optional
Left brace	<code>lbrace</code>	<code>x y height bw bh</code>	<code>lw color op</code>
Right brace	<code>rbrace</code>	<code>x y height bw bh</code>	<code>lw color op</code>
Up brace	<code>ubrace</code>	<code>x y width bw bh</code>	<code>lw color op</code>
Down brace	<code>dbrace</code>	<code>x y width bw bh</code>	<code>lw color op</code>
Left bracket	<code>lbracket</code>	<code>x y width height</code>	<code>lw color op</code>
Right bracket	<code>rbracket</code>	<code>x y width height</code>	<code>lw color op</code>
Up bracket	<code>ubracket</code>	<code>x y width height</code>	<code>lw color op</code>
Down bracket	<code>dbracket</code>	<code>x y width height</code>	<code>lw color op</code>



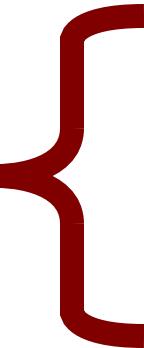
lbrace x y height bw bh lw color op



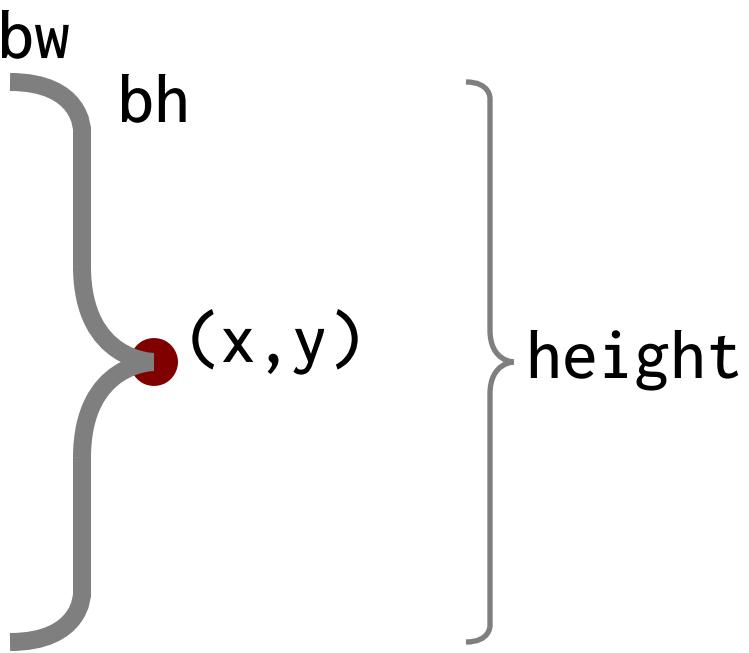
lbrace 20 25 20 2 2



lbrace 50 25 20 4 4 1

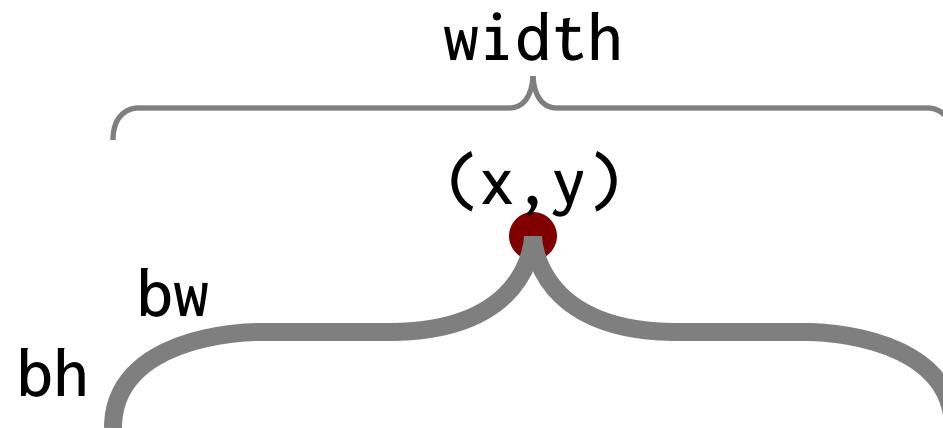


lbrace 80 25 20 6 3 1 "maroon"



rbrace x y height bw bh 1w color op

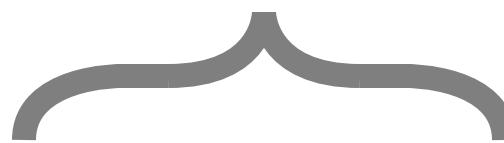
rbrace 20 25 20 2 2 rbrace 50 25 20 4 4 1 rbrace 80 25 20 6 3 1 "maroon"



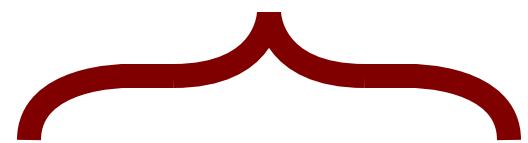
ubrake x y width bw bh lw color op



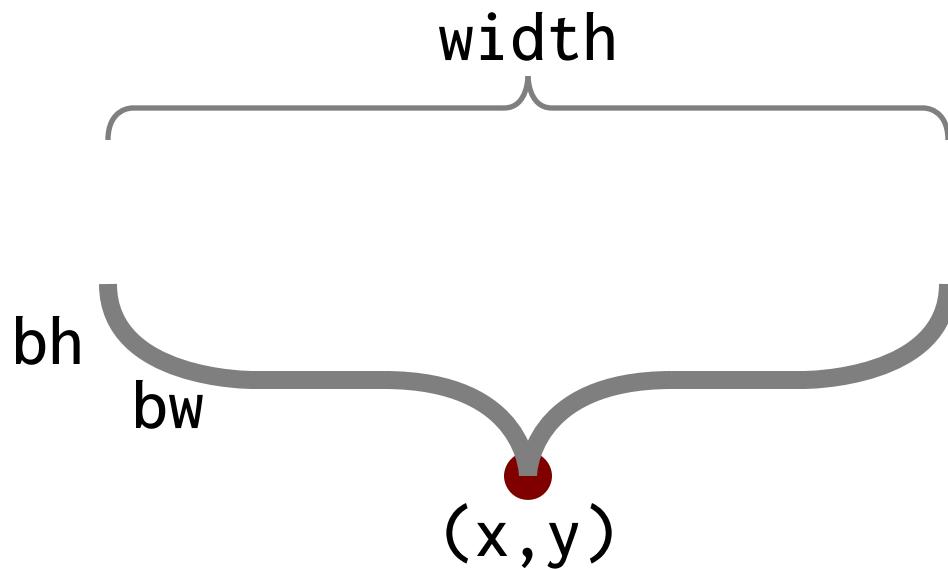
ubrake 20 25 20 2 4



ubrake 50 25 20 4 8 1



ubrake 80 25 20 4 8 1 "maroon"



dbrace x y width bw bh lw color op



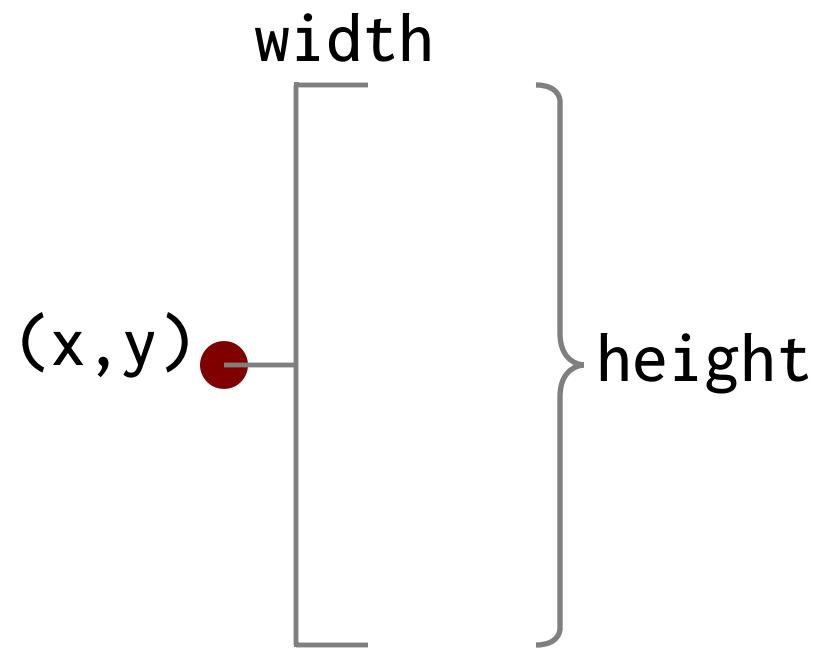
dbrace 20 25 20 2 4



dbrace 50 25 20 4 8 1



dbrace 80 25 20 4 8 1 "maroon"



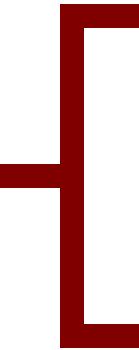
lbracket x y width height lw color op



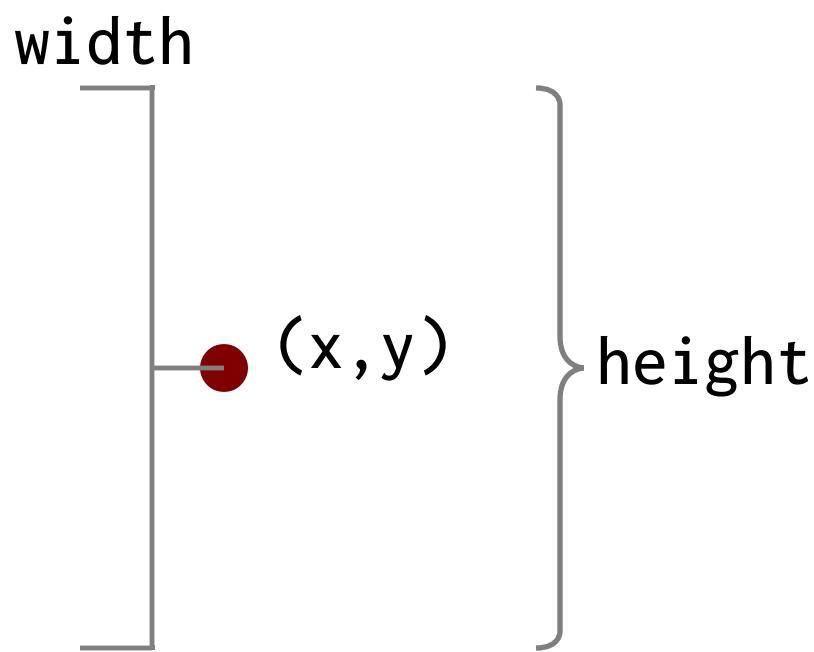
lbrace 20 25 2 20



lbracket 50 25 4 20 1



lbracket 80 25 6 20 1 "maroon"

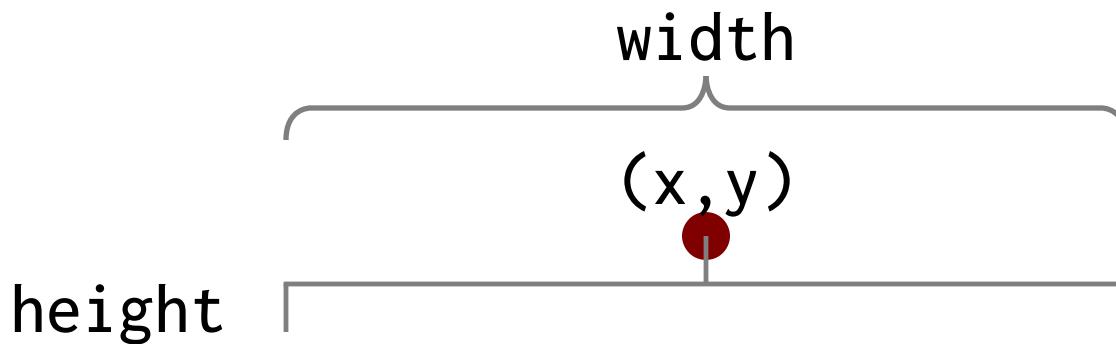


rbracket x y width height lw color op

lbrace 20 25 2 20

rbracket 50 25 4 20 1

rbracket 80 25 6 20 1 "maroon"



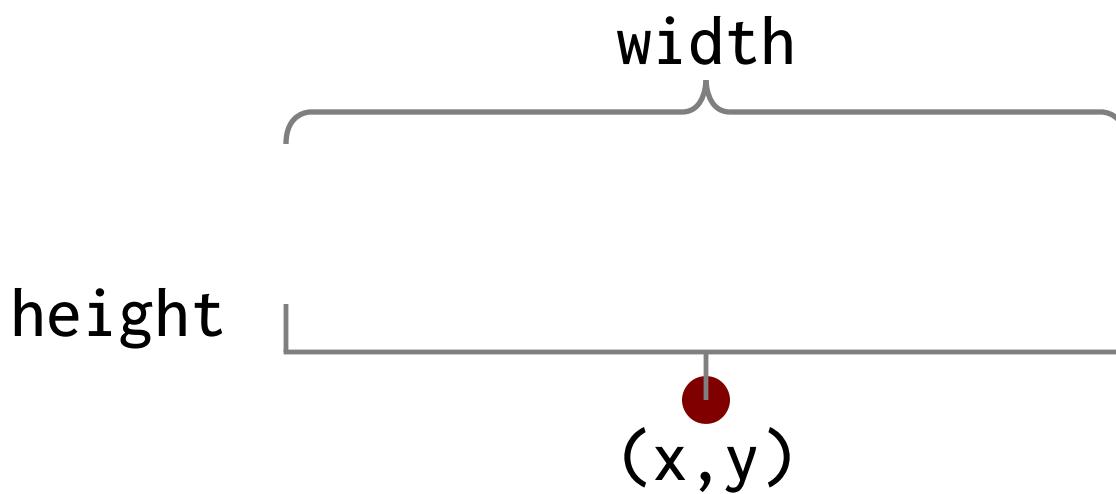
ubracket x y width height lw color op



ubracket 20 25 20 4

ubracket 50 25 20 4 0.3

ubracket 80 25 20 4 1 "maroon"



dbracket x y width height lw color op



dbracket 20 25 20 4



dbracket 50 25 20 4 0.3



dbracket 80 25 20 4 1 "maroon"

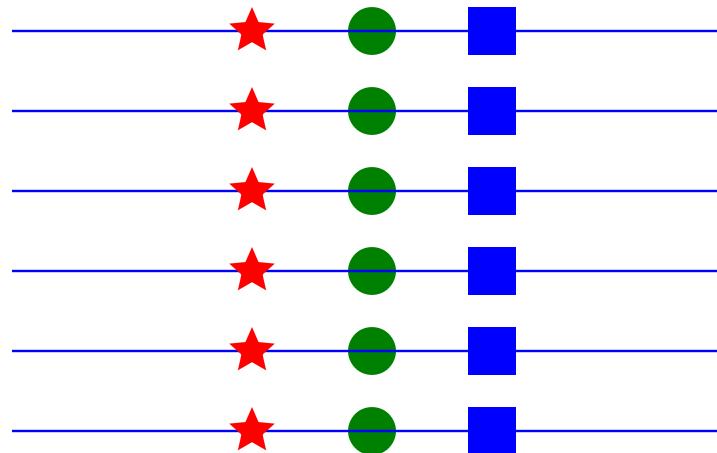
Loop, If, Built-ins

description	keyword	mandatory
Loop	<code>for v=</code>	<code>begin end [increment] ... efor</code>
Conditional	<code>if</code>	<code>condition ... [else] ... eif</code>
Polar coordinate (x)	<code>x=polarx</code>	<code>x y radius angle</code>
Polar coordinate (y)	<code>y=polary</code>	<code>x y radius angle</code>
Polar coordinates	<code>value=polar</code>	<code>x y radius angle</code>
Area	<code>value=area</code>	<code>expression</code>
Formatted text	<code>value=format</code>	<code>fmt expression or up to 5 args</code>
Substring	<code>value=substr</code>	<code>string begin end</code>
Random number	<code>value=random</code>	<code>min max</code>
Value mapping	<code>value=vmap</code>	<code>data min1 max1 min2 max2</code>
Define function	<code>def</code>	<code>name arg1 ... argn ... edef</code>
Import function	<code>import</code>	<code>"file"</code>
In-line data	<code>data</code>	<code>"file" ... edata</code>
Objects on a grid	<code>grid</code>	<code>"file" x y hspace vspace edge</code>
Rulers	<code>ruler</code>	<code>increment [color]</code>

```
for v=begin end [increment]  
...items to repeat using v  
efor
```

for v=begin end increment ...efor

```
for v=10 35 5  
    hline 50 v 30 0.1 "blue"  
    star 60 v 5 1 0.4 "red"  
    circle 65 v 2 "green"  
    square 70 v 2 "blue"  
efor
```



if condition

... statements when true

else

... statements when false

eif

} else block
may be omitted

if condition ... else ... eif

Condition

if a == b
if a != b
if a > b
if a < b
if a >= b
if a <= b
if a >< b c

Description

if a equals b
if a not equal to b
if a less than b
if a greater than b
if a greater than or equal to b
if a less than or equal to b
if a is between b and c

a=60

b=3

if a > b

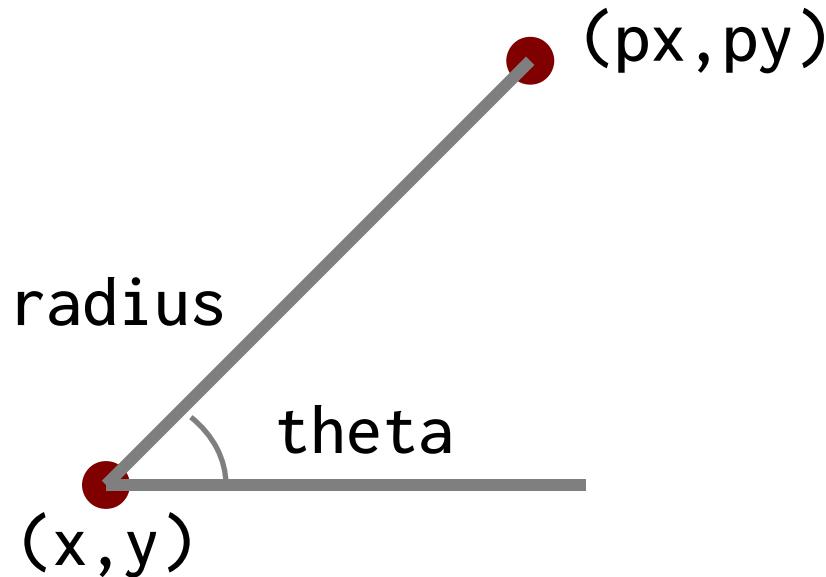
text "hello" a b 2.5 "sans" "red"

else

text "bye" a b 2.5 "sans" "blue"

eif

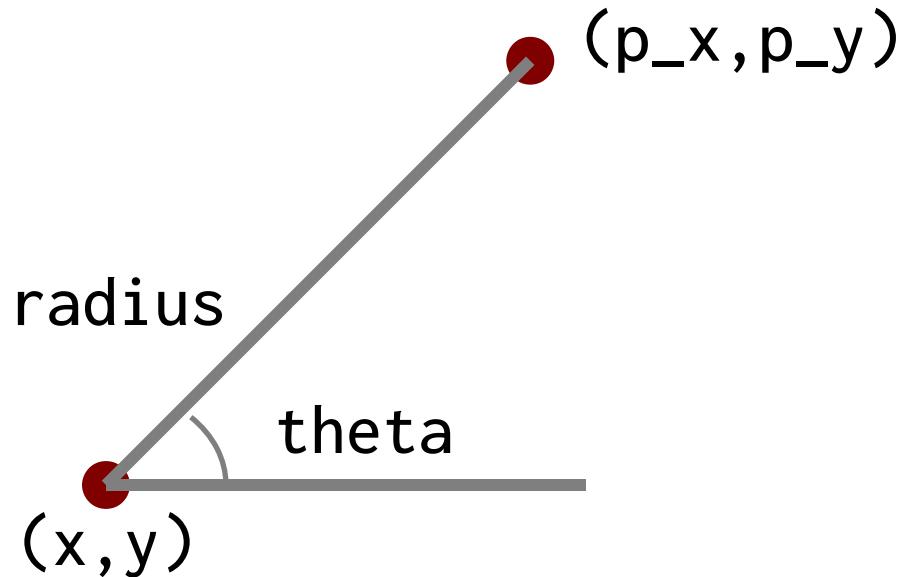
hello



```
px=polarx x y radius theta  
py=polary x y radius theta
```

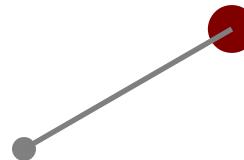
```
cpx=60  
cpy=20  
px1=polarx cpx cpy 10 30  
py1=polary cpx cpy 10 30  
line cpx cpy px1 py1  
circle cpx cpy 1 "gray"  
circle px1 py1 2 "maroon"
```





p=polar x y radius theta

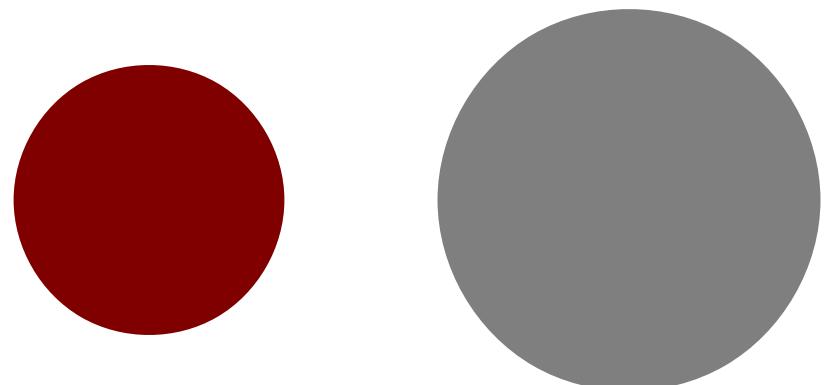
```
cpx=60  
cpy=20  
point=polar cpx cpy 10 30  
line cpx cpy point_x point_y  
circle cpx cpy 1 "gray"  
circle point_x point_y 2 "maroon"
```



v=123.45
a=area v
↑ ↑
area original value

value=area expression

```
m1=100
m2=200
a1=area m1
a2=area m2
circle 60 20 a1 "maroon"
circle 80 20 a2
```



```
x=3.14159  
y=2.0  
title=format "Value=%.2f" x*y  
Value=6.28      format string      expression
```

value=format fmt expression or up to 5 args

v1=100.3
v2=200.234
pi=3.1415926
title=format "%,.2f Million (USD)" v

```
title=format "%.2f Million (USD)" v1
```

```
subtitle=format "Total value: %.2f" v1+v2
```

```
args=format "Multiple args (%v,%v,%.3f)" 80 10 pi
```

ctext title 80 30 4 "sans" "maroon"

ctext subtitle 80 20 3 "sans" "gray"

ctext args 80 10 2 "mono"

100.30 Million (USD)

Total value: 300.53

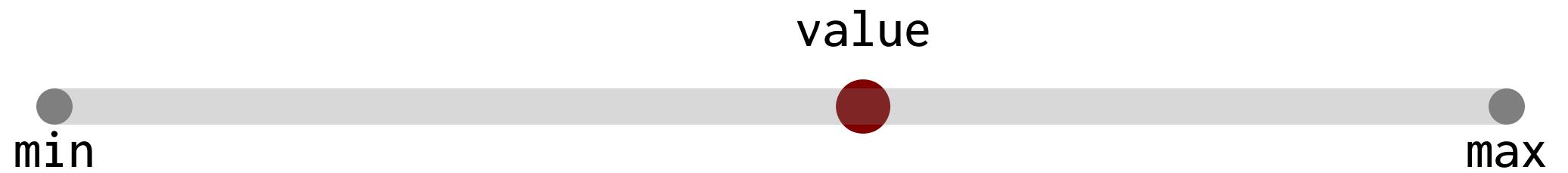
Multiple args (80,10,3.142)

```
s="hello, world"  
h=substr s 3 8  
      ↑ ↑  
 "lo, wo"    begin   end
```

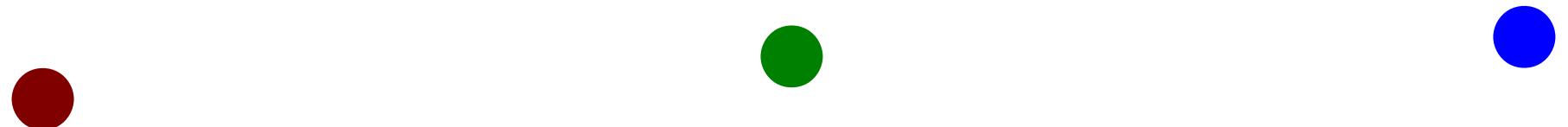
value=**substr** string begin end

```
now="Now is the time for all good men"  
s1=substr now 0 14  
s2=substr now 16 18  
s3=substr now 24 end  
ctext s1 70 34 3  
ctext s2 70 24 3  
ctext s3 70 15 3
```

Now is the time
for
good men



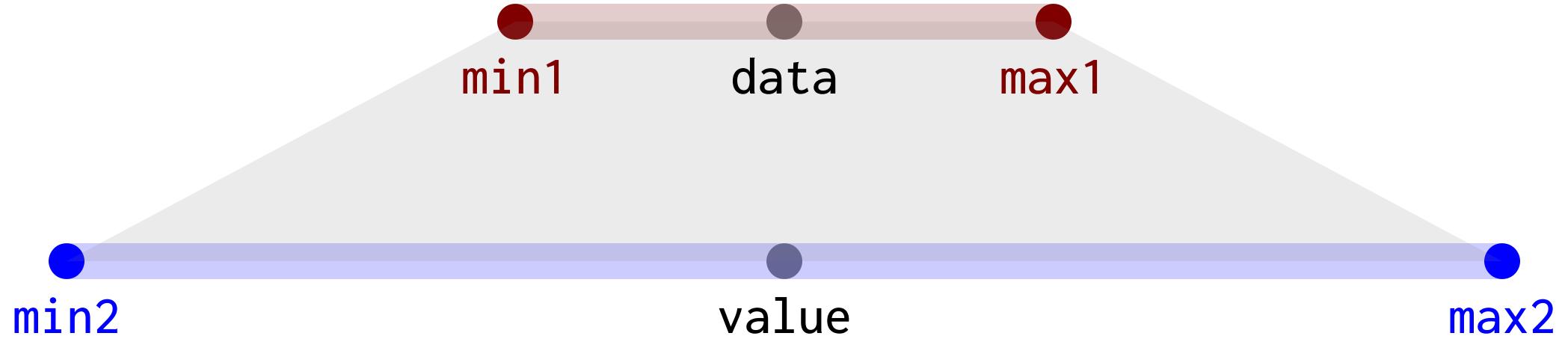
value=random min max



```
rx1=random 5 30  
ry1=random 15 35  
circle rx1 ry1 3 "maroon"
```

```
rx2=random 40 60  
ry2=random 15 35  
circle rx2 ry2 3 "green"
```

```
rx1=random 75 95  
ry1=random 15 35  
circle rx3 ry3 3 "blue"
```



value=vmap data min1 max1 min2 max2

```
yrmin=1776
yrmax=2021
smin=60
smax=90
vp=vmap 1945 yrmin yrmax smin smax
line  smin 20 smax 20 0.5 "gray" 20
circle smin 20 1
circle smax 20 1
circle vp 20 2 "maroon"
```



```
import "doit.dsh"
```

contents of "doit.dsh"

```
def doit fx fy fs ft  
    ctext ft fx fy fs "serif" "purple"  
edef
```

```
doit 50 20 2.5 "hello"
```

fx=50

fs=2.5

ft="hello"

```
    ctext ft fx fy fs "serif" "purple"
```

```
import "file"
```

calling the function

call again

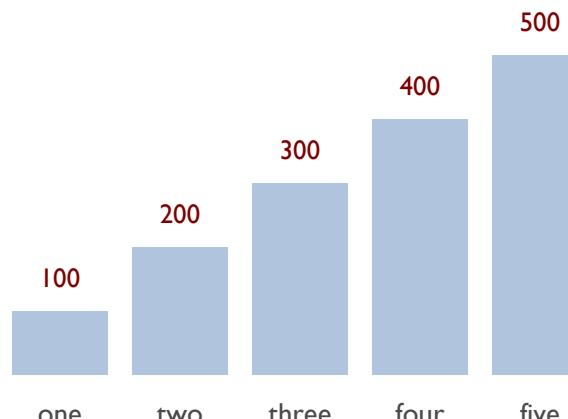
```
doit 50 30 5 "calling the function"  
doit 50 20 4 "call again"
```

```
data "file.d" ← data file  
first 20  
second 100 } data values  
third 200  
edata
```

data "file" ... edata

```
data "test.d"  
one 100  
two 200  
three 300  
four 400  
five 500
```

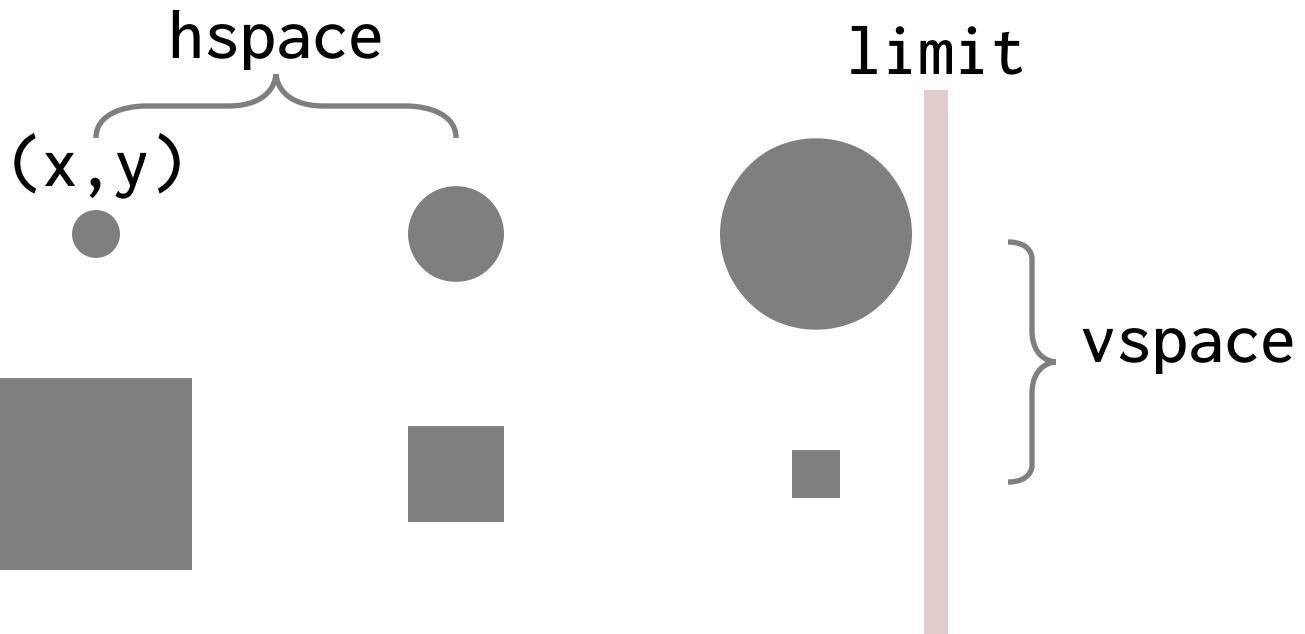
edata



```
dchart -bar -left 50 -bottom 15 -right 70 -top 35 "test.d"
```

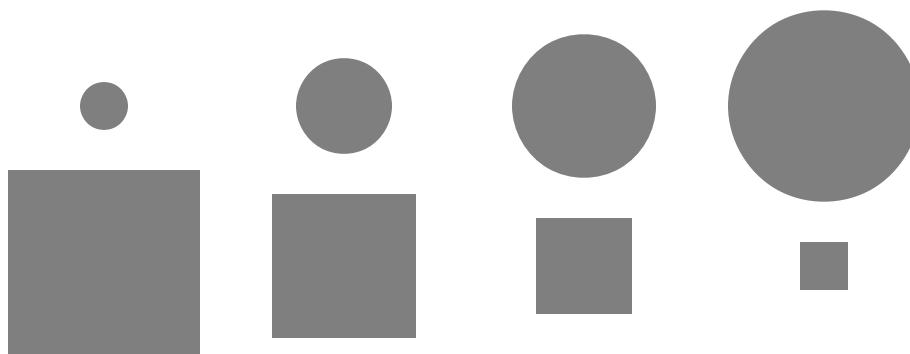
file

```
circle x y 2
circle x y 4
circle x y 8
square x y 8
square x y 4
square x y 2
```

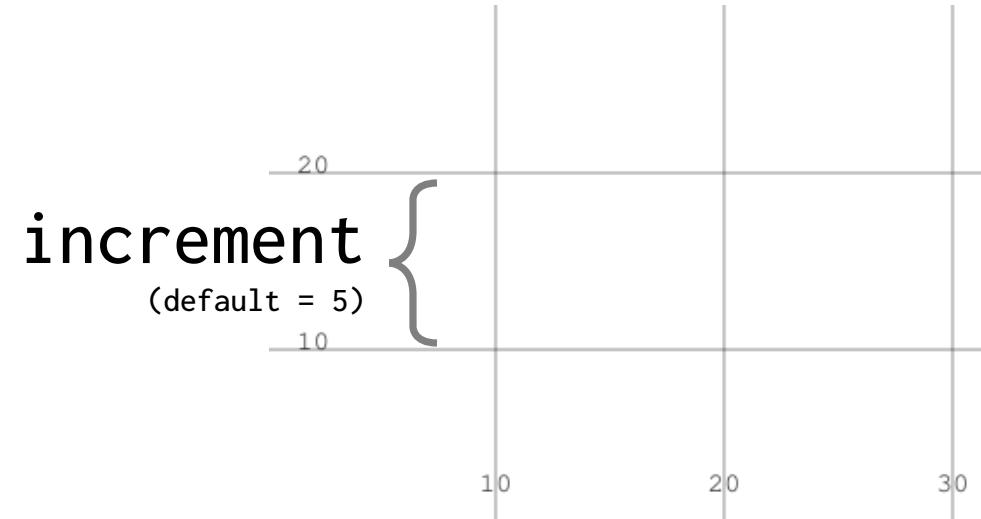


grid "file" x y hspace vspace limit

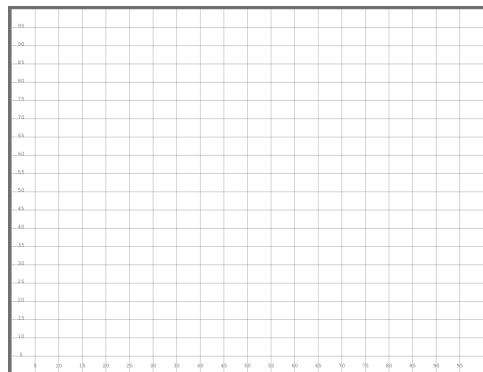
```
circle x y 2
circle x y 4
circle x y 6
circle x y 8
square x y 8
square x y 6
square x y 4
square x y 2
```



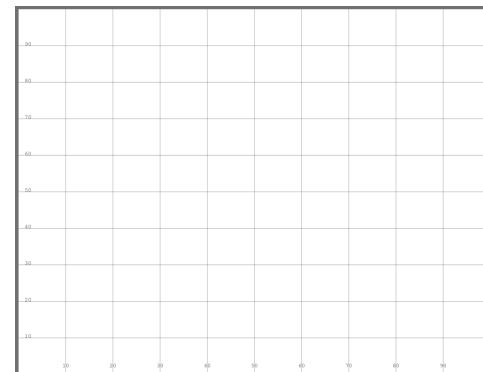
grid "code/grid-ex.dsh" 35 33 10 10 65



ruler increment color



ruler



ruler 10



ruler 2 "red"

Math Functions

description	keyword	mandatory
Cosine	cosine	number or expression
Sine	sine	number or expression
Square Root	sqrt	number or expression
Tangent	tangent	number or expression

a=3.14159265359

b=2*a

x=cosine 4 ← x = -1

y=cosine a ← y = -0.65364

n=cosine a+b ← n = -1

value=**cosine** number or expression

a=10

b=71

x=sin 4 ← x = -0.75680

y=sin a ← y = -0.5440

n=sin a+b ← n = -0.62989

value=sine number or expression

a=10

b=71

x=sqrt 4 ← x = 2

y=sqrt a ← y = 3.1622776

n=sqrt a+b ← n = 9

value=sqrt number or expression

a_squared=10*10

b_squared=20*20

c=sqrt a_squared + b_squared

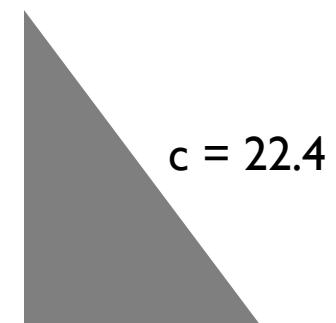
w=70+10

h=15+20

polygon "70 w 70" "15 15 h"

clabel=format "c = %.1f" c

text xlabel 76 25 2



a=10

b=71

x=tangent 4 ← x = 1.1578213

y=tangent a ← y = 0.6483608

n=tangent a+b ← n = -0.8109944

value=tangent number or expression

Charts

description	keyword	mandatory	optional
Charts	dchart	options "file"	
Chart Legends	legend	"text" x y size font color	
Area Chart	areachart	"file"	color
Bar Chart	barchart	"file"	color
Dot Chart	dotchart	"file"	color
Horizontal Bar Chart	hbarchart	"file"	color
Line Chart	linechart	"file"	color
Word Bar Chart	wbarchart	"file"	color

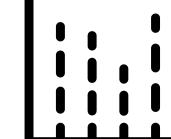
dcharts types



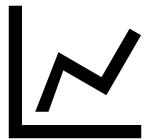
Column



Bar



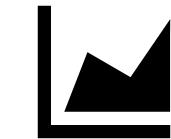
Dot



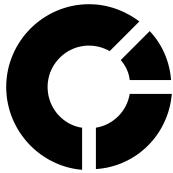
Line



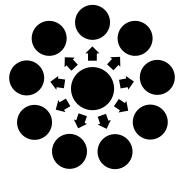
Scatter



Area



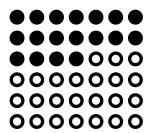
Donut/Pie



Radial



Pmap



Waffle/Lego



Fan



Bowtie

dchart options: chart types

option	default	description
-bar	true	bar chart
-wbar	false	word bar chart
-hbar	false	horizontal bar chart
-donut	false	donut chart
-dot	false	dot chart
-lego	false	lego chart
-line	false	line chart
-pgrid	false	proportional grid
-pmap	false	proportional map
-bowtie	false	bowtie chart
-fan	false	fan chart
-radial	false	radial chart
-scatter	false	scatter chart
-slope	false	slope chart
-vol	false	volume (area) chart

dchart options: elements

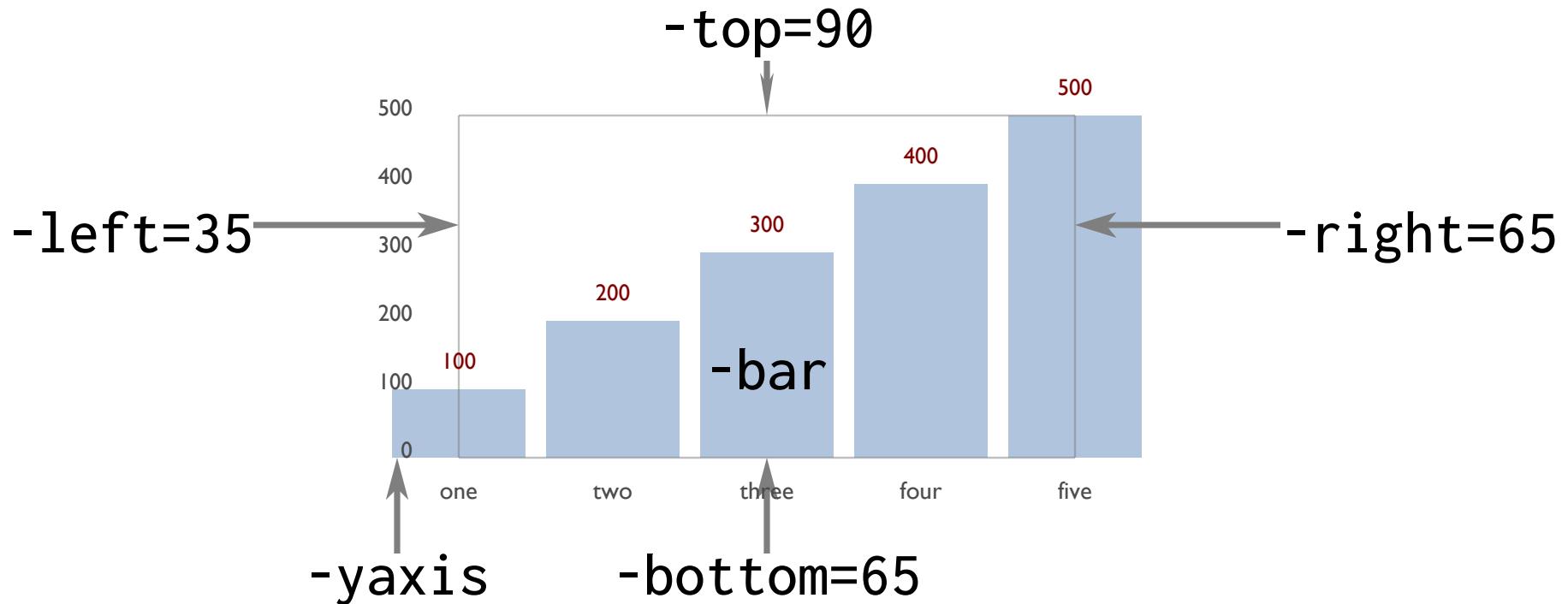
option	default	description
-csv	false	read CSV files
-frame	false	show a colored frame
-fulldeck	true	generate full deck markup
-grid	false	show gridlines on the y axis
-note	true	show annotations
-pct	false	show computed percentage
-rline	false	show a regression line
-solidpmap	false	show solid pmap colors
-spokes	false	show spokes in radial chart
-title	true	show the title
-val	true	show values
-xlast	false	show the last x label
-xstagger	false	stagger x axis labels
-yaxis	false	show a y axis
-charttitle	override title in data	specify the title
-datacond	low,high,color	conditional data colors
-hline	value,label	label horizontal line at value
-valpos	t=top, b=bottom, m=middle	value position
- xlabel	default=1, 0 to suppress	x axis label interval
-yrange	min,max.step	specify the y axis label range

dchart options: measures and attributes

option	default	description
-bgcolor	white	background color
-barwidth	computed from data size	barwidth
-color	lightsteelblue	data color
-csvcol	label1,label2	specify csv columns
-datafmt	%.1f	data format for values
-dmin	false	use data minimum, not zero
-framecolor	rgb(127,127,127)	frame color
-lcolor	rgb(75,75,75)	label color
-linewidth	0.2	linewidth
-ls	2.4	linespacing
-noteloc	c=center, r=right, l=left	annotation location
-pmlen	20	pmap label length
-psize	30	diameter of the donut
-pwidth	3	width of the donut or pmap
-rlcolor	rgb(127,0,0)	regression line color
-textsize	1.5	text size
-xlabrot	0	xlabel rotation (deg.)
-vcolor	rgb(127,0,0)	value color
-volop	50	volume opacity %

dchart options: position and scaling

option	default	description
-top	80	top of the chart
-bottom	30	bottom of the chart
-left	20	left margin
-right	80	right margin
-min	data min	set the minimum data value
-max	data max	set the maximum data value
-bounds	""	set left,right,top,bottom



dchart options "file"



```
dchart -left=10 -right=30 -top=35 -bottom=20 "test.d"
```

```
dchart -left=55 -right=85 -top=35 -bottom=20 -bar=f -line "test.d"
```

Chart Variables

description	variable	default value
Chart Top	chartTop	80
Chart Bottom	chartBottom	30
Chart Left	chartLeft	10
Chart Right	chartRight	90
Show data values	chartVal	1 (0: off, 1: on)
Show Y-Axis Grid	chartGrid	0 (0: off, 1: on)
X-Axis label interval	chartXLabel	1 (1: all, 0: none, every nth label)
Y-Axis range	chartYRange	"" (min,max,interval)



```
legend "text" x y fontsize font color
```

■ Item on the chart

■ Thing

```
legend "Item on the chart" 20 30 3 "sans" "red"
```

```
legend "Thing" 70 30 2 "serif" "blue"
```

Geographic Functions

description	keyword	mandatory	optional
Geographic Regions	georegion	"file"	color op
Geographic Borders	geoborder	"file"	lw color op
Text labels	geolabel	"loc"	size font color op
Dot markers	geomark	"loc"	size color op
Text with markers	geoloc	"loc"	align size font color op
Place images	geoimage	"loc" w h	
Lines between points	geopath	"p1" "p2"	lw color op
Arcs between points	geoarc	"p1" "p2"	lw color op
Lines between points	geopathfile	"file"	lw color op

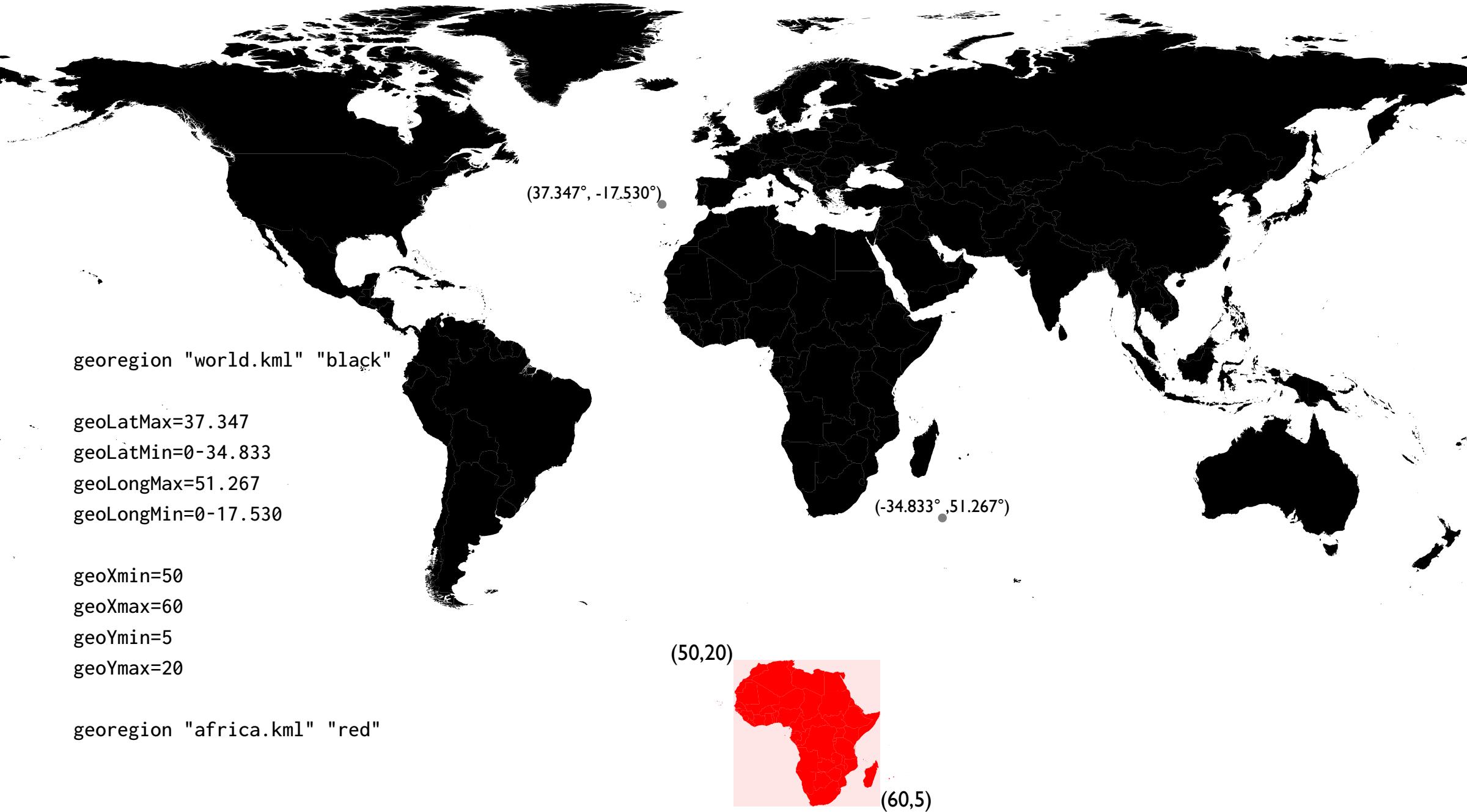
"file" refer to Shapefile (.shp), GeoJSON (.json or .geojson) or Keyhole Markup Language (.kml) files from opendatasoft
geoLatMin, geoLatMax, geoLongMin, geoLongMax variables set the latitude and longitude boundaries

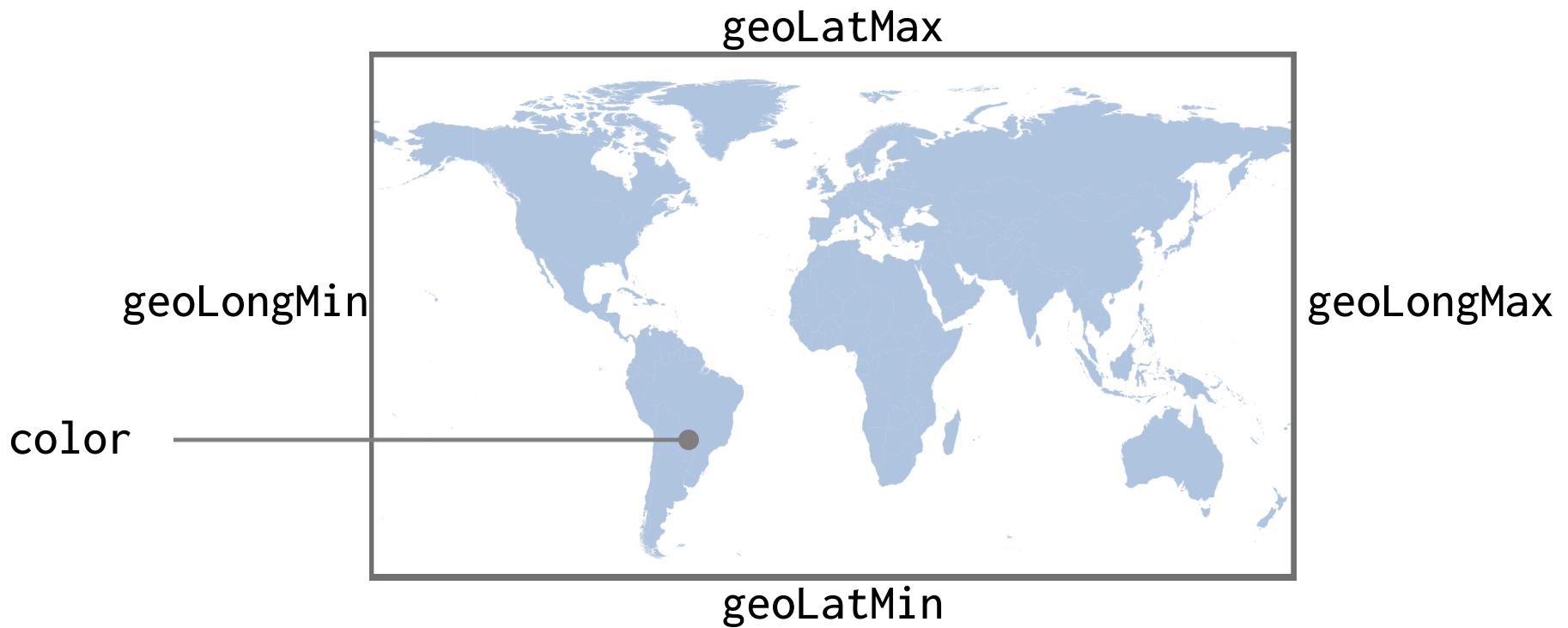
geoXmin, geoXmax, geoYmin, geoYmax variables define the canvas boundaries for geographic functions

"loc" refers to a geo URI string ("geo:lat,long"), or a filename containing multiple locations

"p1" and "p2" are geo URIs

Geographic Scaling





georegion "file" color op

geoLatMin=0-60

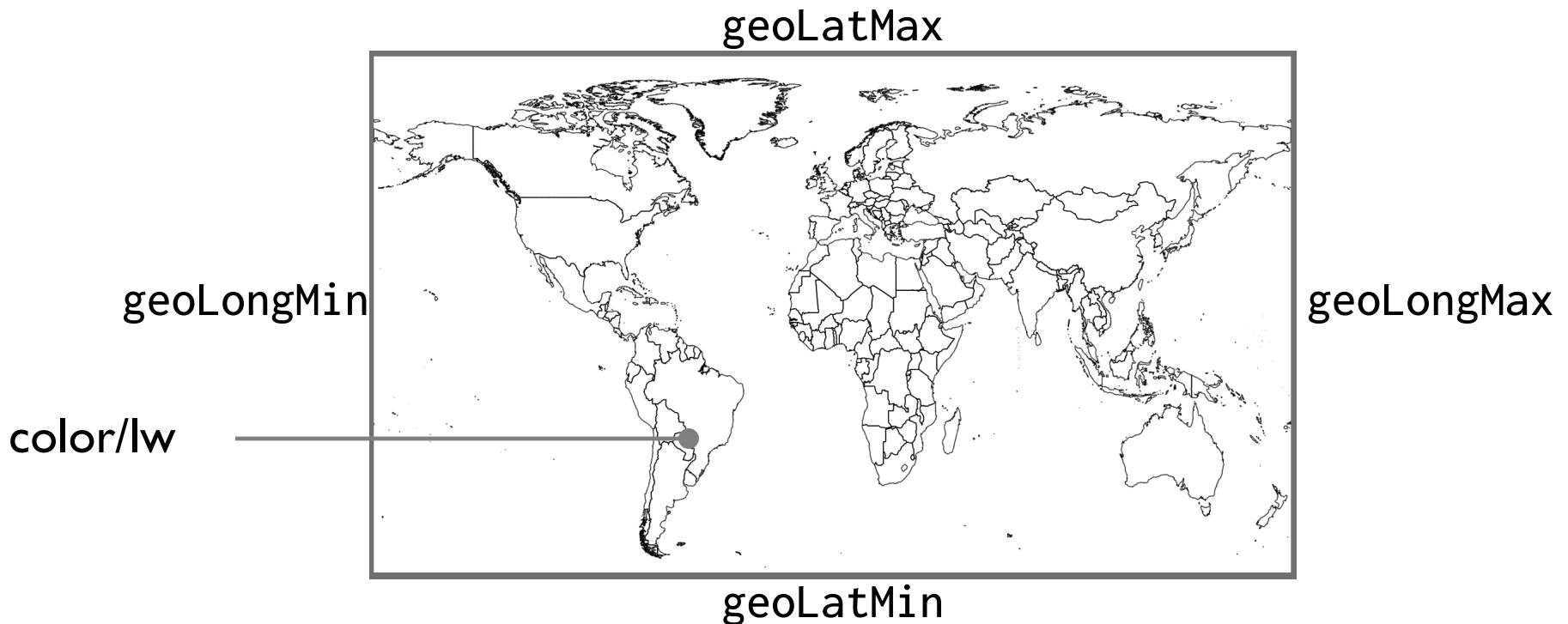
geoLatMax=90

geoLongMin=0-180

geoLongMax=180

georegion "world.kml" "white"





geoborder "file" lw color

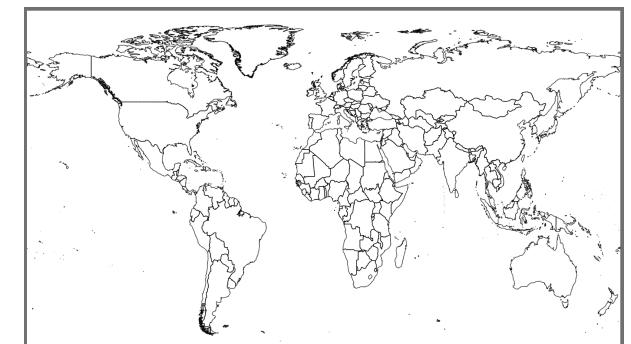
geoLatMin=0-60

geoLatMax=90

geoLongMin=0-180

geoLongMax=180

geoborder "world.kml" 0.1 "black"



geoLatMax

geoLongMin

geo:lat,long<tab>Label

geoLongMax

geoLatMin

geolabel "loc" size color op

geoLatMin=0-60

geoLatMax=90

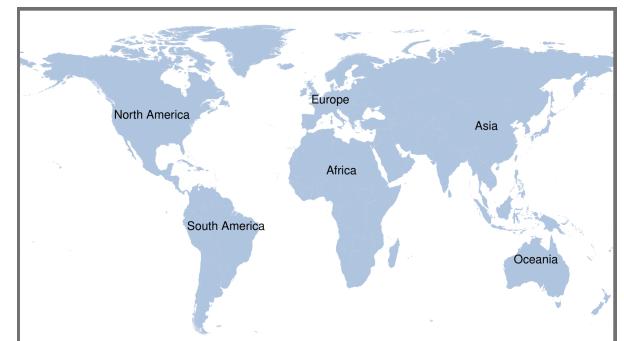
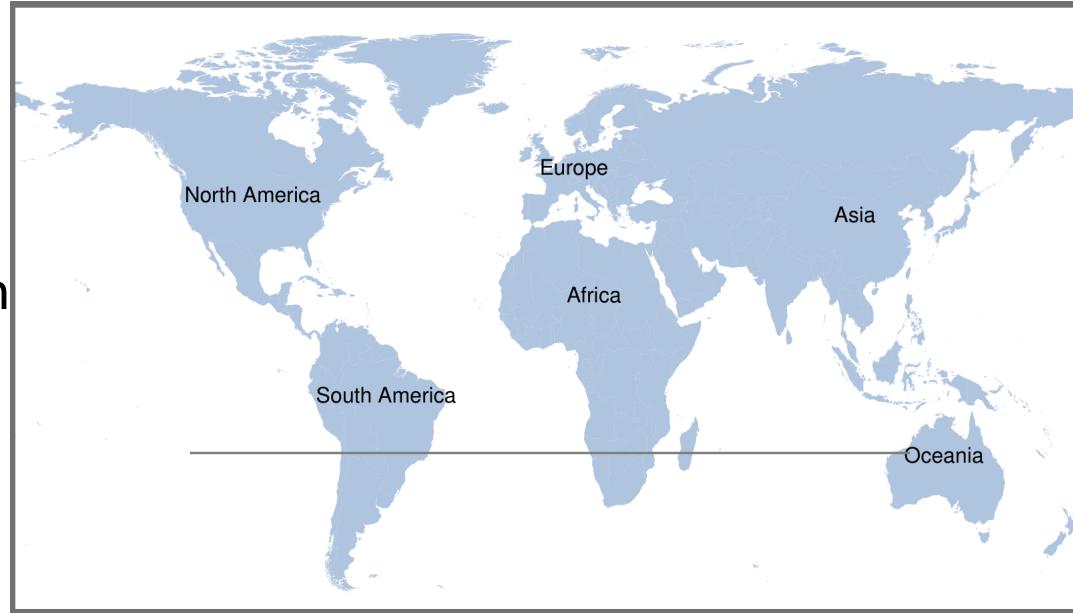
geoLongMin=0-180

geoLongMax=180

geo:-10,-55	South America
geo:40,-100	North America
geo:46.8,8.3	Europe
geo:15,15	Africa
geo:35,103	Asia
geo:-25,133	Oceania

georegion "world.kml" "white"

geolabel "continents.d" 2 "sans" "black"



geoLatMax

geoLongMin

geo:lat, long

geoLongMax

geoLatMin

geomark "loc" size color op

geoLatMin=0-60

geoLatMax=90

geoLongMin=0-180

geoLongMax=180

geo:41.8967,12.4822000	Rome (41.90°, 12.48°)
geo:-18.91368,47.53610	Antananarivo (-18.91°, 47.54°)
geo:-33.8559,151.20670	Sydney (-33.86°, 151.21°)
geo:40.7167,-74.400000	New York (40.72°, -74.40)
...	
geo:34.0500,-118.25000	Los Angeles (34.05°, -118.25°)

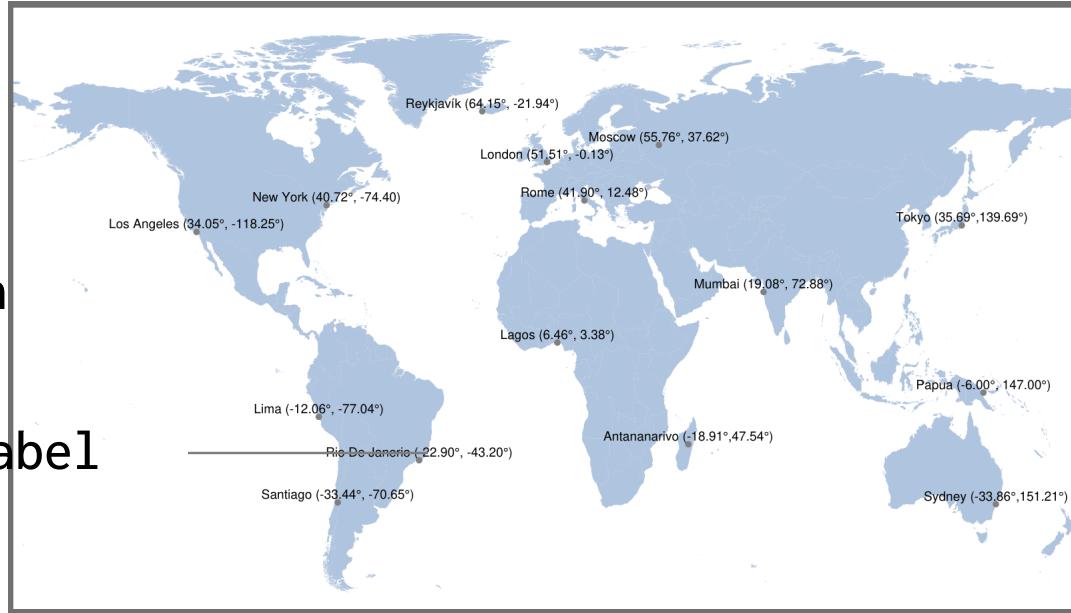
georegion "world.kml" "white"

geolabel "cities.d" 2 "sans" "black"



geoLatMax

geoLongMin
geo:lat, long<tab>Label



geoLongMax

Label Position (pos):
"c" Center above
"b" Begin Aligned
"e" End Aligned
"u" Center below

geoLatMin

geoloc "loc" pos size font color op

geoLatMin=0-60

geoLatMax=90

geoLongMin=0-180

geoLongMax=180

georegion "world.kml" "white"

geoloc "cities.d" "c" ts

Label	Coordinates
Rome	(41.90°, 12.48°)
Antananarivo	(-18.91°, 47.54°)
Sydney	(-33.86°, 151.21°)
New York	(40.72°, -74.40°)
...	
Los Angeles	(34.05°, -118.25°)



geoLatMax

geoLongMin

geoLongMax

geo:lat,long<tab>imgfile

geoLatMin

geoimage "loc" w h

geoLatMin=25

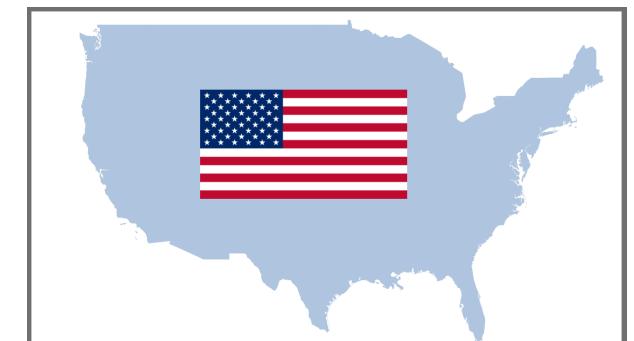
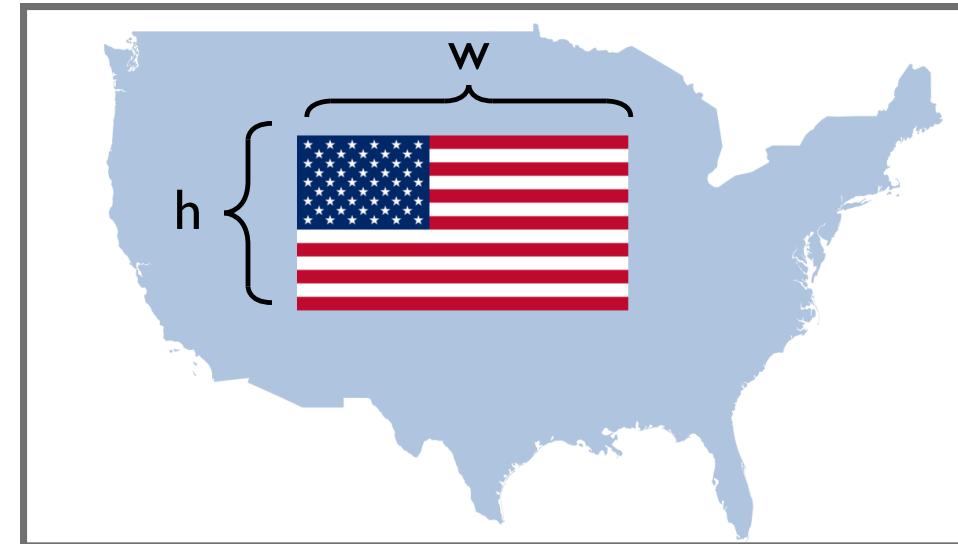
geoLatMax=50

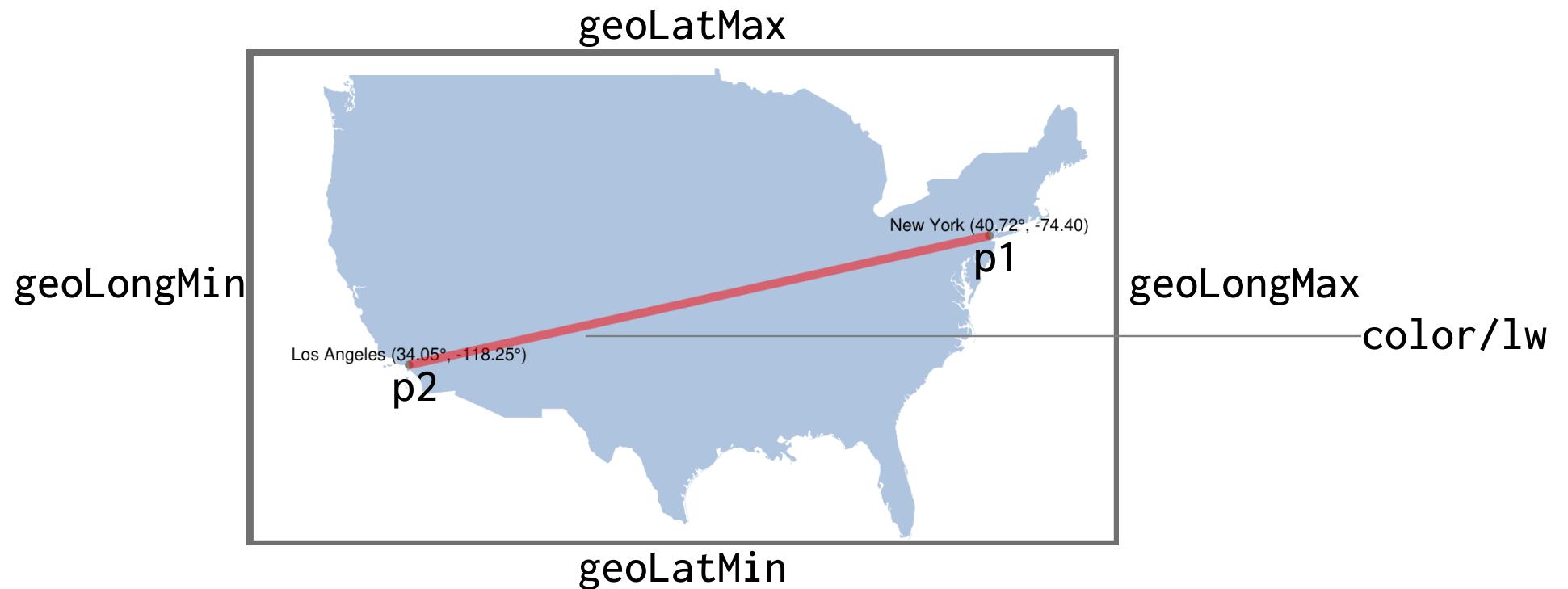
geoLongMin=0-130

geoLongMax=0-65

georegion "usa.kml" "lightsteelblue"

geoimage "geo:40,-100" usa.png" 35 0





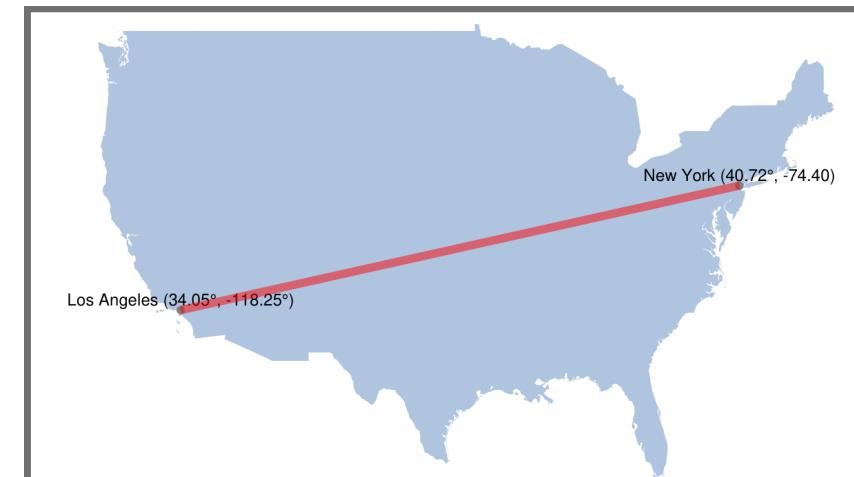
geopath "p1" "p2" lw lw color op

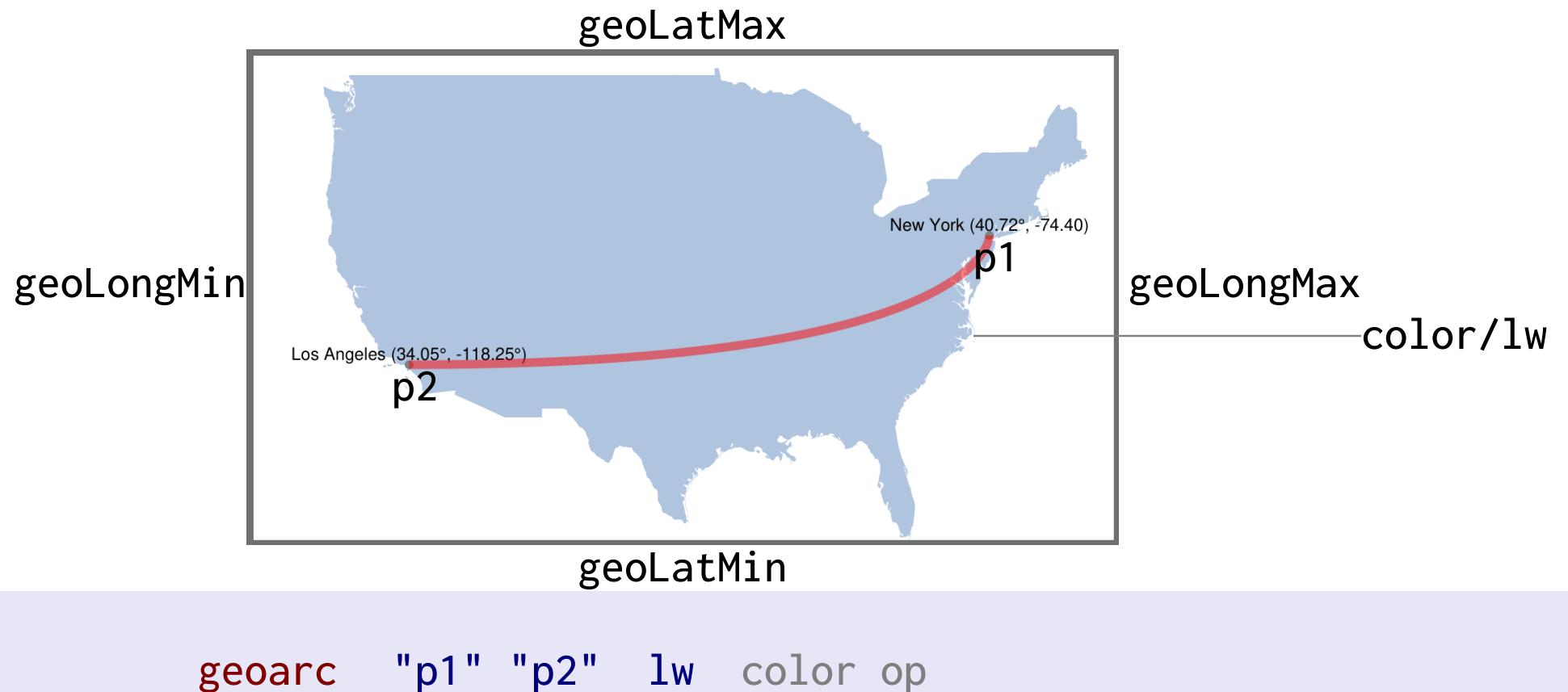
```

geoLatMin=25
geoLatMax=50
geoLongMin=0-130
geoLongMax=0-65
nyc="geo:+40.7167,-74.4000      New York (40.72°, -74.40)"
los="geo:+34.05000,-118.250     Los Angeles (34.05°, -118.25°)"

georegion "usa.kml" "lightsteelblue"
geoloc   nyc "c" 2
geoloc   los "c" 2
geopath   nyc los 1 "red" 50

```



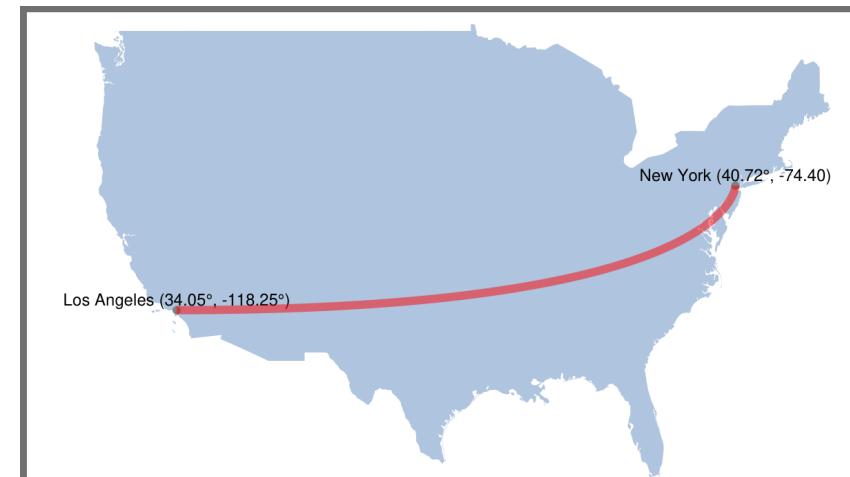


```

geoLatMin=25
geolatmax=50
geoLongMin=0-130
geoLongMax=0-65
nyc="geo:+40.7167,-74.4000      New York (40.72°, -74.40)"
los="geo:+34.05000,-118.250     Los Angeles (34.05°, -118.25°)"

georegion "usa.kml" "lightsteelblue"
geoloc    nyc "c" 2
geoloc    los "c" 2
geoarc    nyc los 1 "red" 50

```



geoLatMax

geoLongMin

geoLongMax

geoLatMin

geopathfile "file" lw color op

geoLatMin=25

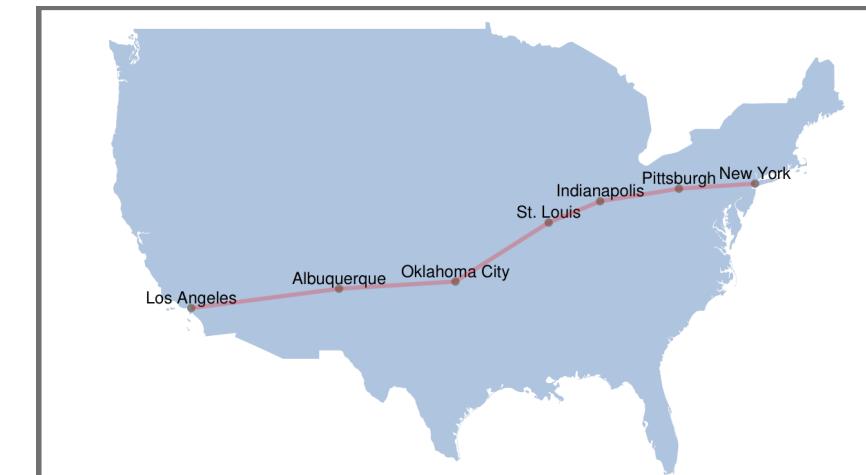
geoLatMax=50

geoLongMin=0-130

geoLongMax=0-65

geo:40.712778,-74.006111	New York
geo:39.952778,-75.163611	Philadelphia
geo:39.768611,-86.158056	Indianapolis
geo:38.627222,-90.197778	St. Louis
geo:35.468611,-97.521389	Oklahoma City
geo:35.084444,-106.650278	Albuquerque
geo:36.167222,-115.148611	Las Vegas
geo:34.050000,-118.250000	Los Angeles

```
georegion "usa.kml" "lightsteelblue"
geoloc "roadtrip.d" "c" 2
geopathfile "roadtrip.d" 0.5 "red" 25
```



Keyword Reference

Keyword	Arguments	Description
Structure	<code>deck</code>	Begin a deck; end with "edeck"
	<code>def</code>	Define a function; end with "edef"
	<code>for</code>	Begin loop; end with "efor"
	<code>if</code>	Conditional; one of: a==b, a!=b, a>b, a<b, a>=b, a<=b, a<>b c
	<code>import</code>	import function found in a file
	<code>include</code>	Include the contents of a file
	<code>slide</code>	Begin a slide; end with "eslide"
	<code>edeck</code>	End the deck
	<code>edef</code>	End the defintion
	<code>efor</code>	End the for loop
Utility	<code>eif</code>	End the conditional
	<code>else</code>	Begin the else clause
	<code>canvas</code>	Define with dimensions of the canvas
	<code>content</code>	Embed content
	<code>dump</code>	Dump variables
Graphics	<code>grid</code>	Define a content grid
	<code>ruler</code>	draw a (x,y) ruler
	<code>acircle</code>	Circle with sized based on area
	<code>arc</code>	Elliptical arc centered at (x,y), dimensions (w,h) between angles a1 and a2
	<code>circle</code>	Circle centered at (x,y), diameter w
	<code>curve</code>	Quadratic Bezier Curve begin (bx,by), control (cx, cy), end (ex,ey)
	<code>ellipse</code>	Ellipse centered at (x,y), dimension (w,h)
	<code>hline</code>	Horizontal line begin at (x,y), length w
	<code>line</code>	Line between (x1,y1) and (x2,y2)
	<code>pill</code>	Pill shape beginning at (x,y), dimensions (w,h)
	<code>polygon</code>	Polygon with specified x, y coordinates
	<code>polyline</code>	Polyline with specified x, y coordinates
	<code>rect</code>	Rectangle centered at (x,y), dimensions (w,h)
	<code>rrect</code>	Rounded rectangle centered at (x,y), dimensions (w,h), corner radius r
	<code>square</code>	Square centered at (x,y), size w
	<code>star</code>	Star centered at (x,y), with sides, innner and outer sizes
	<code>vline</code>	Vertical line beginning at (x,y), h high

	Keyword	Arguments	Description
Text	<code>arctext</code>	"string" x y radius a1 a2 fontsize [font] [color] [opacity] [link]	Text on an arc, at fontsize, center (x,y), radius r, between a1. a2
	<code>btext</code>	"string" x y fontsize [font] [color] [opacity] [link]	Text beginning at (x,y), at fontsize
	<code>ctext</code>	"string" x y fontsize [font] [color] [opacity] [link]	Centered text beginning at (x,y), at fontsize
	<code>etext</code>	"string" x y fontsize [font] [color] [opacity] [link]	End-aligned text at (x,y), at fontsize
	<code>rtext</code>	"string" x y angle fontsize [font] [color] [opacity] [link]	Rotated text centered at (x,y), at angle and fontsize
	<code>text</code>	"string" x y fontsize [font] [color] [opacity] [link]	Text beginning at (x,y), at fontsize
	<code>textblock</code>	"string" x y w fontsize [font] [color] [opacity] [link]	Block of text beginning at (x,y), at fontsize, with width w
	<code>textblockfile</code>	"file" x y w fontsize [font] [color] [opacity] [link]	Block of text read for a file, beginning at (x,y), at fontsize, with width w
	<code>textcode</code>	"file" x y w fontsize [font] [color] [opacity]	Lines of code, read from a file, upper right corner at (x,y), margin at w
	<code>textfile</code>	"file" x y fontsize [font] [color] [opacity] [spacing]	Contents of a text file pper right corner at (x,y)
Lists	<code>blist</code>	x y fontsize [font] [color] [opacity] [spacing]	Bulleted list starting at (x,y), at fontsize
	<code>clist</code>	x y fontsize [font] [color] [opacity] [spacing]	Centered list starting at (x,y), at fontsize
	<code>list</code>	x y fontsize [font] [color] [opacity] [spacing]	List starting at (x,y), at fontsize
	<code>nlist</code>	x y fontsize [font] [color] [opacity] [spacing]	Numbered list starting at (x,y), at fontsize
	<code>li</code>	"item" [font] [color] [opacity]	List item
	<code>elist</code>		End the list
Images	<code>cimage</code>	"file" "caption" x y w h [scale] [link] capsiz	Captioned image; center (x,y), dimensions (w,h) (h=0, w is % of canvas width)
	<code>image</code>	"file" x y w h [scale] [link]	Image center at (x,y), dimensions (w,h) (h=0, w is % of canvas width)
Braces/ Brackets	<code>dbrace</code>	x y w bw bh [lw] [color] [opacity]	Downward pointing brace
	<code>dbracket</code>	x y w h [lw] [color] [opacity]	Downward pointing bracket
	<code>lbrace</code>	x y h bw bh [lw] [color] [opacity]	Left pointing brace
	<code>lbracket</code>	x y w h [lw] [color] [opacity]	Left pointing bracket
	<code>rbrace</code>	x y h bw bh [lw] [color] [opacity]	Right pointing brace
	<code>rbracket</code>	x y w h [lw] [color] [opacity]	Right pointing bracket
	<code>ubrace</code>	x y w bw bh [lw] [color] [opacity]	Upward facing brace
	<code>ubracket</code>	x y w h [lw] [color] [opacity]	Upward facing bracket

	Keyword	Arguments	Description
Arrows	<code>arrow</code>	<code>x1 y1 x2 y2 [lw] [aw] [ah] [color] [opacity]</code>	Arrow starting at (x1,y1), ending at (x2,y2), aw=width, ah=height
	<code>darrow</code>	<code>bx by bx xy ex ey [lw] [aw] [ah] [color] [opacity]</code>	Downward curved arrow; curve specified by (bx,by), (cx,cy), (ex,ey)
	<code>larrow</code>	<code>bx by bx xy ex ey [lw] [aw] [ah] [color] [opacity]</code>	Left curved arrow; curve specified by (bx,by), (cx,cy), (ex,ey)
	<code>rarrow</code>	<code>bx by bx xy ex ey [lw] [aw] [ah] [color] [opacity]</code>	Right curved arrow; curve specified by (bx,by), (cx,cy), (ex,ey)
	<code>uarrow</code>	<code>bx by bx xy ex ey [lw] [aw] [ah] [color] [opacity]</code>	Upward curved arrow; curve specified by (bx,by), (cx,cy), (ex,ey)
Charts	<code>dchart</code>	options...	Chart with specified options
	<code>legend</code>	"string" x y fontsize font color	Chart legend
Built-ins	<code>x=area</code>	expression	Assign an area
	<code>x=format</code>	"fmt" expr... (up to 5)	Assign formatting to expressions
	<code>x=polar</code>	x y radius angle	Assign polar coordinate centered at (x,y) at radius and angle (0-360)
	<code>x=polarx</code>	x y radius angle	Assign X-polar coordinate centered at (x,y) at radius and angle (0-360)
	<code>x=poly</code>	x y radius angle	Assign Y-polar coordinate centered at (x,y) at radius and angle (0-360)
	<code>x=random</code>	min max	Assign a random number between two values
	<code>x=substr</code>	"string" begin end	Assign a substring
	<code>x=vmap</code>	data min1 max1 min2 max2	Assign a value mapped to two ranges
Math	<code>x=cosine</code>	expression	Assign the cosine of expression
	<code>x=sine</code>	expression	Assign the sine of expression
	<code>x=sqrt</code>	expression	Assign the square root of expression
	<code>x=tangent</code>	expression	Assign the tangent of expression
	<code>geoarc</code>	"p1" "p2" [lw] [color] [opacity]	Draw arcs between points
Geographic	<code>geoborder</code>	"file" [lw] [color] [opacity]	Reads KML data from the specified file and renders the map borders
	<code>geoimage</code>	"loc" width height	Place an image at a geographical location
	<code>geolabel</code>	"loc" [size] [font] [color] [opacity]	Reads data from the specified file or location and renders the map labels
	<code>geoloc</code>	"loc" [align] [size] [font] [color] [opacity]	Reads data from the specified file or location and a make map point and labels
	<code>geomark</code>	"loc" [size] [color] [opacity]	Reads data from the specified file or location and renders map points
	<code>geopath</code>	"p1" "p2" [lw] [color] [opacity]	Draw line between points
	<code>geopathfile</code>	"file" [lw] [color] [opacity]	Reads data from the specified file and a make lines between points
	<code>georegion</code>	"file" [color] [opacity]	Reads KML data from the specified file and renders the map regions