



# Figure Cheat Sheet

RASCAL  
[www.rascal-mpl.org](http://www.rascal-mpl.org)

Font and Text Properties	
font(str fontName)	Named font
fontSize(int s)	Font size
fontColor(Color c)	Font color
fontColor(str name)	Font color
textAngle(real a)	Text angle

Wedge Properties	
fromAngle(real a)	Begin angle
toAngle(real a)	End angle
innerRadius(real r)	Inner radius

Other Properties	
id(str name)	Figure name

Property (Prop)

Size Properties	
width(real w)	Horizontal (H) size
height(real h)	Vertical (V) size
size(real s)	H and V size
size(real w, real h)	H and V size
gap(real g)	H and V gap
gap(real w, real h)	H and V gap

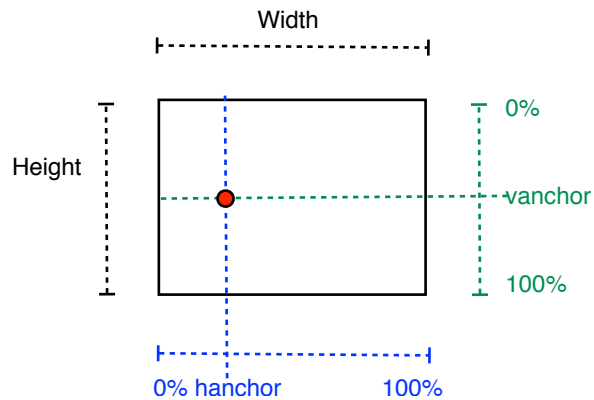
Alignment Properties	
anchor(real h, real v)	Anchor point
hanchor(real h)	H anchor point
vanchor(real v)	V anchor point

Line and Border Properties	
lineWidth(int w)	Line width
lineColor(Color c)	Line color
fillColor(Color c)	Fill color
fillColor(str name)	Fill color

Shape Properties	
connected()	Connect vertices
closed()	Close shape
curved()	Interpolate vertices

Color	
Color gray(int gray, real alpha)	Gray value (0-255) with transparency (0-100%)
Color gray(real perc)	Gray value (0-100%)
Color gray(real gray, real perc)	Gray value (0-100%) with transparency (0-100%)
Color color(str colorName)	Named color (from SVG standard) <a href="http://www.w3.org/TR/SVG/types.html#ColorKeywords">http://www.w3.org/TR/SVG/types.html#ColorKeywords</a>
Color color(str colorName, real alpha)	Named color with transparency (0-100%)
Color rgb(int r, int g, int b)	Color based on red (r), green (g) and blue (b)
Color rgb(int r, int g, int b, real alpha)	RGB color with transparency
list[Color] interpolateColor(Color from, Color to, real perc)	Interpolate between two colors (in RGB space)
list[Color] colorSteps(Color from, Color to, int steps)	List of interpolated colors
Color(int) colorScale(list[int] values, Color from, Color to)	Color scale from list of integers
Color(real) colorScale(list[real] values, Color from, Color to)	Color scale from list of reals

## Bounding Box Model



Aliases
alias Color = int;
alias Props = list[Prop]
alias Figures = list[Figure]
alias Edges = list[Edge]
alias Nodes = Figures

Primitives	
box(Props p)	box
box(Props p, Figure f)	box with inner figure
ellipse(Props p)	ellipse
ellipse(Props p, Figure f)	ellipse with inner figure
wedge(Props p)	wedge
wedge(Props p, Figure f)	wedge with inner figure
space(Props p)	invisible box
space(Props p, Figure f)	invisible box with inner figure

Transformation	
rotate(real a, Figure f)	Rotated figure
scale(real perc, Figure f)	Scaled figure
scale(real xperc,real yperc, Figure f)	Scaled figure

Composition	
hcat(Figures fs)	Horizontal concatenation
hcat(Props p, Figures fs)	Horizontal concatenation
vcat(Figures fs)	Vertical concatenation
vcat(Props p, Figures fs)	Vertical concatenation
align(Figures fs)	Horizontal and vertical composition
overlay(Figure fs)	Overlay of figures
overlay(Props p, Figure fs)	Overlay of figures
grid(Figures fs)	Fixed grid layout
grid(Props p, Figures fs)	Fixed grid layout
pack(Figures fs)	2D packing of figures
pack(Props p, Figures fs)	2D packing of figures
graph(Nodes n, Edges E)	Graph layout
graph(Props p, Nodes n, Edges E)	Graph layout
tree(Nodes n, Edges e)	Tree layout
tree(Props p, Nodes n, Edges e)	Tree layout

# Figure (Figure)

Vertex	
vertex(real x, real y)	(x,y) point
vertex(real x, real y, Figure F)	(x,y) point with figure

Edge	
edge(str from, str to)	edge between named figures
edge(Props p, str from, str to)	edge with properties

Example
<pre> module Example import viz::Figure::Core; import viz::Figure::Render;  // Display a red box of 100x200  public void main(){     Figure b = box([size(100,200), fillColor("red")]);     render(b); } </pre>