**Sketchpad.js** cheatsheet

=======================

SKETCHPAD

**new** Sketchpad(canvasId)

Sketch.update()

Sketch.visible()

Sketch.invisible()

Sketch.currentStyle(style)

Sketch.findElementNames()

Sketch.width {Measure}

Sketch.height {Measure}

ELEMENT

Element.addParents(element0, element1...)

Element.isChildOf(element0, element1...)

Element.updateChildren()

Element.checkStates()

Element.setVisible(**boolean**)

Element.setStyle(style)

Element.findName()

POINT

**new** Point(number, number)

Point.fromMeasures(measure, measure)

Point.along(line, number)

Point.along(circle, number)

Point.along(line, measure)

Point.along(circle, measure)

Point.along(line, numericSet)

Point.along(circle, numericSet)

Point.projection(point, line)

Point.projection(point, circle)

Point.intersection(line, line)

Point.intersection(line, circle)

Point.intersection(circle, circle)

Point.offset(point, number/measure, numer/measure)

Point.offsetPolar(point, number/measure, number/measure)

Point.setPosition(number, number)

Point.move(number, number)

NODE

**new** Node(number, number)

Node.along(line, number)

Node.along(circle, number)

Node.horizontal(number, number)

Node.horizontal(number, measure)

Node.vertical(number, number)

Node.vertical(measure, number)

LINE

**new** Line(number, number, number, number) *// x0, y0, x1, y1*

Line.between(point, point)

Line.between(point, pointSet)

Line.between(pointSet, pointSet)

Line.polar(point, number, number) *// startPoint, length, angle*

Line.polar(point, measure, number)

Line.polar(point, number, measure)

Line.polar(point, measure, measure)

CIRCLE

**new** Circle(number, number, number) *// x, y, radius*

Circle.centerRadius(point, number)

Circle.centerRadius(point, measure)

Circle.setRadius(number)

MEASURE

Measure.distance(point, point)

Measure.angle(point, point, point)

Measure.compose(object, [object...], **function**)

SET

Set.range(number, number, number) *// start, end, steps*

Set.sequence(number, number, number) *// start, stepSize, count*

Set.random(number, number, number) *// start, end, count*

Set.setItems(JSArray, number) *// objects, subtype as pad.C.FOO constant*

STYLE

**new** Style({

stroke: '',

strokeWidth: '',

fill: '',

fontFamily: '',

fontSize: '',

fontStyle: '',

fontCSS: '',

textFill: '',

textVAlign: '',

textHAlign: '',

textOffsetX: 0,

textOffsetY: 0

})

Style.applyTo(element0, element1...)

TAG

**new** Text(string, number, number) *// text, xpos, ypos*

Text.on(point, [text])

LABEL

**new** Label(string) *// text*

Label.compose(object, [object...], **function**)

Label.from(object)