

Objects

[OG](#) : object

Functions

[override\(origclass, overrides\)](#)

Adds a list of functions to the prototype of an existing class, overwriting any existing methods with the same name. Usage:

```
Ext.override(MyClass, {  
  newMethod1: function(){  
    // etc.  
  },  
  newMethod2: function(foo){  
    // etc.  
  }  
});
```

OG : object

Kind: global namespace

- [OG](#) : object
 - [.common](#) : object
 - [.Constants](#)
 - [new OG.common.Constants\(\)](#)
 - [.GEOM_TYPE](#)
 - [.GEOM_NAME](#)
 - [.NUM_PRECISION](#)
 - [.NODE_TYPE](#)
 - [.SHAPE_TYPE](#)
 - [.EDGE_TYPE](#)
 - [.LABEL_SUFFIX](#)
 - [.LABEL_EDITOR_SUFFIX](#)
 - [.FROM_LABEL_SUFFIX](#)
 - [.TO_LABEL_SUFFIX](#)
 - [.RUBBER_BAND_ID](#)
 - [.RUBBER_BAND_TOLERANCE](#)
 - [.GUIDE_SUFFIX](#)
 - [.COLLAPSE_SUFFIX](#)
 - [.LOOPTYPE_SUFFIX](#)
 - [.TASKTYPE_SUFFIX](#)
 - [.INCLUSION_SUFFIX](#)
 - [.STATUS_SUFFIX](#)

- [.EXCEPTIONTYPE SUFFIX](#)
- [.MOVE SNAP SIZE](#)
- [.DROP OVER BBOX SUFFIX](#)
- [.TERMINAL SUFFIX](#)
- [.TERMINAL](#)
- [.MARKER TEMP NODE](#)
- [.PATTERN TEMP NODE](#)
- [.MARKER DEFS SUFFIX](#)
- [.ORIGINAL NODE](#)
- [.CONNECT GUIDE EVENT AREA](#)
- [.CONNECT GUIDE SUFFIX](#)
- [.CurveUtil](#)
 - [new OG.common.CurveUtil\(\)](#)
 - [.CatmullRomSpline\(points\)](#) ⇒ Object
 - [.Bezier\(points\)](#) ⇒ Object
 - [.BSpline\(points, order\)](#) ⇒ Object
- [.HashMap](#)
 - [new OG.common.HashMap\(jsonObject\)](#)
 - [.map](#) : Object
 - [.put\(key, value\)](#)
 - [.get\(key\)](#) ⇒ Object
 - [.containsKey\(key\)](#) ⇒ Boolean
 - [.containsValue\(value\)](#) ⇒ Boolean
 - [.isEmpty\(\)](#) ⇒ Boolean
 - [.clear\(\)](#)
 - [.remove\(key\)](#)
 - [.keys\(\)](#) ⇒ Array.<String>
 - [.values\(\)](#) ⇒ Array.<Object>
 - [.size\(\)](#) ⇒ Number
 - [.toString\(\)](#) ⇒ String
- [.JSON](#)
 - [new OG.common.JSON\(\)](#)
 - [.encode](#) ⇒ String
 - [.decode](#) ⇒ Object
 - [.encodeDate\(d\)](#) ⇒ String
- [.Util](#)
 - [new OG.common.Util\(\)](#)
 - [.extend](#) ⇒ function
 - [.clone\(obj\)](#) ⇒ Object
 - [.round\(val\)](#) ⇒ Number
 - [.roundPrecision\(val, precision\)](#) ⇒ Number
 - [.roundGrid\(val, snapSize\)](#) ⇒ Number
 - [.apply\(obj, config, defaults\)](#) ⇒ Object
- [.geometry](#) : object

- [.BezierCurve](#) \Leftarrow [PolyLine](#)
 - [new OG.geometry.BezierCurve\(controlPoints\)](#)
 - [.controlPoints](#) : [Array.<Coordinate>](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getControlPoints\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.getVertices\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) \Rightarrow [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) \Rightarrow [Geometry](#)
 - [.rotate\(angle, origin\)](#) \Rightarrow [Geometry](#)
 - [.toString\(\)](#) \Rightarrow String
 - [.angleBetweenPoints\(prev, next\)](#) \Rightarrow Number
 - [.isRightAngleBetweenPoints\(prev, next\)](#) \Rightarrow Object
 - [.angleBetweenThreePoints\(prev, next\)](#) \Rightarrow Number
 - [.isEqual\(geometry\)](#) \Rightarrow Boolean
 - [.isContains\(geometry\)](#) \Rightarrow Boolean
 - [.isWithin\(geometry\)](#) \Rightarrow Boolean
 - [.getBoundary\(\)](#) \Rightarrow [Envelope](#)
 - [.getCentroid\(\)](#) \Rightarrow [Coordinate](#)
 - [.minDistance\(coordinate\)](#) \Rightarrow Number
 - [.distance\(geometry\)](#) \Rightarrow Number
 - [.getLength\(\)](#) \Rightarrow Number
 - [.moveCentroid\(중심\)](#)
 - [.resizeBox\(width, height\)](#) \Rightarrow [Geometry](#)
 - [.fitToBoundary\(envelope\)](#) \Rightarrow [Geometry](#)
 - [.convertCoordinate\(coordinate\)](#) \Rightarrow [Coordinate](#)
 - [.distanceToLine\(p, line\)](#) \Rightarrow Number
 - [.distanceLineToLine\(line1, line2\)](#) \Rightarrow Number
 - [.intersectToLine\(line\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.shortestIntersectToLine\(line\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.intersectLineToLine\(line1, line2, extension\)](#) \Rightarrow [Coordinate](#)
 - [.intersectCircleToLine\(center, radius, from, to\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.intersectPointToLine\(p, line\)](#) \Rightarrow [Coordinate](#)
 - [.getPercentageDistanceFromPoint\(coordinate\)](#) \Rightarrow Object
 - [.isContainsPoint\(coordinate\)](#) \Rightarrow boolean
 - [.getPointFromPercentageDistance\(pXpY\)](#) \Rightarrow [Coordinate](#)
 - [.getParallelLine\(from, to, distance\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.getParallelPath\(line, distance\)](#)
 - [.reset\(\)](#)
- [.Circle](#) \Leftarrow [Ellipse](#)
 - [new OG.geometry.Circle\(center, radius\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)

- [.TYPE](#) : Number
- [.IS_CLOSED](#) : Boolean
- [.style](#) : [Style](#)
- [.boundary](#) : [Envelope](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.getControlPoints\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.getLength\(\)](#) ⇒ Number
- [.toString\(\)](#) ⇒ String
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(geometry\)](#) ⇒ Boolean
- [.isContains\(geometry\)](#) ⇒ Boolean
- [.isWithin\(geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(coordinate\)](#) ⇒ Number
- [.distance\(geometry\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(coordinate\)](#) ⇒ Object
- [.isContainsPoint\(coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)
- [.Coordinate](#)
 - [new OG.geometry.Coordinate\(x, y\)](#)
 - [.x](#) : Number
 - [.y](#) : Number
 - [.distance\(coordinate\)](#) ⇒ Number
 - [.move\(offsetX, offsetY\)](#) ⇒ [Coordinate](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Coordinate](#)
 - [.isEqual\(coordinate\)](#) ⇒ Boolean

- [.toString\(\)](#) ⇒ String
- [.Curve](#) ⇐ [PolyLine](#)
 - [new OG.geometry.Curve\(controlPoints\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getControlPoints\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
 - [.toString\(\)](#) ⇒ String
 - [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
 - [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
 - [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
 - [.isEqual\(geometry\)](#) ⇒ Boolean
 - [.isContains\(geometry\)](#) ⇒ Boolean
 - [.isWithin\(geometry\)](#) ⇒ Boolean
 - [.getBoundary\(\)](#) ⇒ [Envelope](#)
 - [.getCentroid\(\)](#) ⇒ [Coordinate](#)
 - [.minDistance\(coordinate\)](#) ⇒ Number
 - [.distance\(geometry\)](#) ⇒ Number
 - [.getLength\(\)](#) ⇒ Number
 - [.moveCentroid\(중심\)](#)
 - [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
 - [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
 - [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
 - [.distanceToLine\(p, line\)](#) ⇒ Number
 - [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
 - [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
 - [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
 - [.getPercentageDistanceFromPoint\(coordinate\)](#) ⇒ Object
 - [.isContainsPoint\(coordinate\)](#) ⇒ boolean
 - [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
 - [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getParallelPath\(line, distance\)](#)
 - [.reset\(\)](#)
- [.Ellipse](#) ⇐ [Curve](#)
 - [new OG.geometry.Ellipse\(center, radiusX, radiusY, angle\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)

- [.TYPE](#) : Number
- [.IS_CLOSED](#) : Boolean
- [.style](#) : [Style](#)
- [.boundary](#) : [Envelope](#)
- [.getControlPoints\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.toString\(\)](#) ⇒ String
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(geometry\)](#) ⇒ Boolean
- [.isContains\(geometry\)](#) ⇒ Boolean
- [.isWithin\(geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(coordinate\)](#) ⇒ Number
- [.distance\(geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(coordinate\)](#) ⇒ Object
- [.isContainsPoint\(coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)
- [.Envelope](#)
 - [new OG.geometry.Envelope\(upperLeft, width, height\)](#)
 - [.getUpperLeft\(\)](#) ⇒ [Coordinate](#)
 - [.setUpperLeft\(upperLeft\)](#)
 - [.getUpperRight\(\)](#) ⇒ [Coordinate](#)
 - [.getLowerRight\(\)](#) ⇒ [Coordinate](#)
 - [.getLowerLeft\(\)](#) ⇒ [Coordinate](#)
 - [.getLeftCenter\(\)](#) ⇒ [Coordinate](#)

- [.getUpperCenter\(\)](#) ⇒ [Coordinate](#)
- [.getRightCenter\(\)](#) ⇒ [Coordinate](#)
- [.getLowerCenter\(\)](#) ⇒ [Coordinate](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.setCentroid\(centroid\)](#)
- [.getWidth\(\)](#) ⇒ Number
- [.setWidth\(width\)](#)
- [.getHeight\(\)](#) ⇒ Number
- [.setHeight\(height\)](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.isContains\(coordinate\)](#) ⇒ Boolean
- [.isContainsAll\(coordinateArray\)](#) ⇒ Boolean
- [.getHowManyContains\(coordinateArray\)](#) ⇒ Boolean
- [.isContainsOnce\(coordinateArray\)](#) ⇒ Boolean
- [.move\(offsetX, offsetY\)](#) ⇒ [Envelope](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Envelope](#)
- [.isEqual\(Envelope\)](#) ⇒ Boolean
- [.toString\(\)](#) ⇒ String

- [.Geometry](#)
 - [new OG.geometry.Geometry\(\)](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.isEqual\(geometry\)](#) ⇒ Boolean
 - [.isContains\(geometry\)](#) ⇒ Boolean
 - [.isWithin\(geometry\)](#) ⇒ Boolean
 - [.getBoundary\(\)](#) ⇒ [Envelope](#)
 - [.getCentroid\(\)](#) ⇒ [Coordinate](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.minDistance\(coordinate\)](#) ⇒ Number
 - [.distance\(geometry\)](#) ⇒ Number
 - [.getLength\(\)](#) ⇒ Number
 - [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
 - [.moveCentroid\(중심\)](#)
 - [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
 - [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
 - [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
 - [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
 - [.distanceToLine\(p, line\)](#) ⇒ Number
 - [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
 - [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
 - [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)

- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(coordinate\)](#) ⇒ [Object](#)
- [.isContainsPoint\(coordinate\)](#) ⇒ [boolean](#)
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)
- [.GeometryCollection](#) ⇐ [Geometry](#)
 - [new OG.geometry.GeometryCollection\(geometries\)](#)
 - [.geometries](#) : [Array.<Geometry>](#)
 - [.TYPE](#) : [Number](#)
 - [.IS_CLOSED](#) : [Boolean](#)
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.toString\(\)](#) ⇒ [String](#)
 - [.isEqual\(geometry\)](#) ⇒ [Boolean](#)
 - [.isContains\(geometry\)](#) ⇒ [Boolean](#)
 - [.isWithin\(geometry\)](#) ⇒ [Boolean](#)
 - [.getBoundary\(\)](#) ⇒ [Envelope](#)
 - [.getCentroid\(\)](#) ⇒ [Coordinate](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.minDistance\(coordinate\)](#) ⇒ [Number](#)
 - [.distance\(geometry\)](#) ⇒ [Number](#)
 - [.getLength\(\)](#) ⇒ [Number](#)
 - [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
 - [.moveCentroid\(중심\)](#)
 - [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
 - [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
 - [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
 - [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
 - [.distanceToLine\(p, line\)](#) ⇒ [Number](#)
 - [.distanceLineToLine\(line1, line2\)](#) ⇒ [Number](#)
 - [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
 - [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
 - [.getPercentageDistanceFromPoint\(coordinate\)](#) ⇒ [Object](#)
 - [.isContainsPoint\(coordinate\)](#) ⇒ [boolean](#)
 - [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
 - [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getParallelPath\(line, distance\)](#)
 - [.reset\(\)](#)
- [.Line](#) ⇐ [PolyLine](#)

- [new OG.geometry.Line\(from, to\)](#)
- [.vertices](#) : [Array.<Coordinate>](#)
- [.TYPE](#) : Number
- [.IS_CLOSED](#) : Boolean
- [.style](#) : [Style](#)
- [.boundary](#) : [Envelope](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.toString\(\)](#) ⇒ String
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(geometry\)](#) ⇒ Boolean
- [.contains\(geometry\)](#) ⇒ Boolean
- [.isWithin\(geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(coordinate\)](#) ⇒ Number
- [.distance\(geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(coordinate\)](#) ⇒ Object
- [.isContainsPoint\(coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)
- [.Point](#) ⇐ [Geometry](#)
 - [new OG.geometry.Point\(coordinate\)](#)
 - [.coordinate](#) : [Coordinate](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)

- [.boundary](#) : [Envelope](#)
- [.toString\(\)](#) ⇒ [String](#)
- [.isEqual\(geometry\)](#) ⇒ [Boolean](#)
- [.isContains\(geometry\)](#) ⇒ [Boolean](#)
- [.isWithin\(geometry\)](#) ⇒ [Boolean](#)
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.minDistance\(coordinate\)](#) ⇒ [Number](#)
- [.distance\(geometry\)](#) ⇒ [Number](#)
- [.getLength\(\)](#) ⇒ [Number](#)
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.moveCentroid\(중심\)](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ [Number](#)
- [.distanceLineToLine\(line1, line2\)](#) ⇒ [Number](#)
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(coordinate\)](#) ⇒ [Object](#)
- [.isContainsPoint\(coordinate\)](#) ⇒ [boolean](#)
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)
- [.PolyLine](#) ⇐ [Geometry](#)
 - [new OG.geometry.PolyLine\(vertices\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : [Number](#)
 - [.IS_CLOSED](#) : [Boolean](#)
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.toString\(\)](#) ⇒ [String](#)
 - [.angleBetweenPoints\(prev, next\)](#) ⇒ [Number](#)
 - [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ [Object](#)
 - [.angleBetweenThreePoints\(prev, next\)](#) ⇒ [Number](#)
 - [.isEqual\(geometry\)](#) ⇒ [Boolean](#)
 - [.isContains\(geometry\)](#) ⇒ [Boolean](#)
 - [.isWithin\(geometry\)](#) ⇒ [Boolean](#)
 - [.getBoundary\(\)](#) ⇒ [Envelope](#)

- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.minDistance\(coordinate\)](#) ⇒ Number
- [.distance\(geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.moveCentroid\(중심\)](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(coordinate\)](#) ⇒ Object
- [.isContainsPoint\(coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)
- [.Polygon](#) ⇐ [PolyLine](#)
 - [new OG.geometry.Polygon\(vertices\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
 - [.toString\(\)](#) ⇒ String
 - [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
 - [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
 - [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
 - [.isEqual\(geometry\)](#) ⇒ Boolean
 - [.isContains\(geometry\)](#) ⇒ Boolean
 - [.isWithin\(geometry\)](#) ⇒ Boolean
 - [.getBoundary\(\)](#) ⇒ [Envelope](#)
 - [.getCentroid\(\)](#) ⇒ [Coordinate](#)
 - [.minDistance\(coordinate\)](#) ⇒ Number

- [.distance\(geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(coordinate\)](#) ⇒ Object
- [.isContainsPoint\(coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)
- [.Rectangle](#) ⇐ [Polygon](#)
 - [new OG.geometry.Rectangle\(upperLeft, width, height\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
 - [.toString\(\)](#) ⇒ String
 - [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
 - [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
 - [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
 - [.isEqual\(geometry\)](#) ⇒ Boolean
 - [.isContains\(geometry\)](#) ⇒ Boolean
 - [.isWithin\(geometry\)](#) ⇒ Boolean
 - [.getBoundary\(\)](#) ⇒ [Envelope](#)
 - [.getCentroid\(\)](#) ⇒ [Coordinate](#)
 - [.minDistance\(coordinate\)](#) ⇒ Number
 - [.distance\(geometry\)](#) ⇒ Number
 - [.getLength\(\)](#) ⇒ Number
 - [.moveCentroid\(중심\)](#)
 - [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
 - [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
 - [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)

- [.distanceToLine\(p, line\)](#) ⇒ Number
 - [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
 - [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
 - [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
 - [.getPercentageDistanceFromPoint\(coordinate\)](#) ⇒ Object
 - [.isContainsPoint\(coordinate\)](#) ⇒ boolean
 - [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
 - [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getParallelPath\(line, distance\)](#)
 - [.reset\(\)](#)
- [.Style](#) ⇐ [HashMap](#)
 - [new OG.geometry.Style\(style\)](#)
 - [.map](#) : Object
 - [.put\(key, value\)](#)
 - [.get\(key\)](#) ⇒ Object
 - [.containsKey\(key\)](#) ⇒ Boolean
 - [.containsValue\(value\)](#) ⇒ Boolean
 - [.isEmpty\(\)](#) ⇒ Boolean
 - [.clear\(\)](#)
 - [.remove\(key\)](#)
 - [.keys\(\)](#) ⇒ [Array.<String>](#)
 - [.values\(\)](#) ⇒ [Array.<Object>](#)
 - [.size\(\)](#) ⇒ Number
 - [.toString\(\)](#) ⇒ String
- [.graph](#) : object
 - [.Canvas](#)
 - [new OG.graph.Canvas\(container, containerSize, backgroundColor, backgroundImage\)](#)
 - [.initConfig\(config\)](#)
 - [.getRenderer\(\)](#) ⇒ [OG.RaphaelRenderer](#)
 - [.getContainer\(\)](#) ⇒ [HTMLElement](#)
 - [.getEventHandler\(\)](#) ⇒ [OG.EventHandler](#)
 - [.addSlider\(\)](#)
 - [.removeSlider\(\)](#)
 - [.drawShape\(position, shape, size, style, id, parentId, preventEvent\)](#) ⇒ [Element](#)
 - [.drawTransformer\(position, label, inputs, outputs, id\)](#) ⇒ [Element](#)
 - [.setShapeStyle\(shapeElement, style\)](#)
 - [.setTextListInController\(shapeElement, textList\)](#)
 - [.getTextListInController\(shapeElement\)](#)
 - [.drawLabel\(shapeElement, text, style\)](#) ⇒ [Element](#)
 - [.redrawConnectedEdge\(element\)](#)

- [.reconnect\(edge\)](#) ⇒ Element
- [.connect\(fromElement, toElement, style, label, fromP, toP, preventTrigger, id, edgeShape\)](#) ⇒ * | Element
- [.connectWithTerminalId\(fromTerminal, toTerminal, style, label\)](#) ⇒ String | String | [geometry](#)
- [.disconnect\(element\)](#)
- [.group\(elements\)](#) ⇒ Element
- [.ungroup\(groupElements\)](#) ⇒ Array.<Element>
- [.addToGroup\(groupElement, elements\)](#)
- [.collapse\(element\)](#)
- [.expand\(element\)](#)
- [.clear\(\)](#)
- [.removeShape\(element\)](#)
- [.removeChild\(element\)](#)
- [.removeGuide\(element\)](#)
- [.removeAllGuide\(\)](#)
- [.getRootElement\(\)](#) ⇒ Element
- [.getRootGroup\(\)](#) ⇒ Element
- [.getElementByPoint\(position\)](#) ⇒ Element
- [.getElementsByBBox\(envelope\)](#) ⇒ Array.<Element>
- [.setAttr\(element, attribute\)](#)
- [.getAttr\(element, attrName\)](#) ⇒ Object
- [.toFront\(element\)](#)
- [.toBack\(element\)](#)
- [.bringForward\(element\)](#)
- [.sendBackward\(element\)](#)
- [.getCanvasSize\(\)](#) ⇒ Array.<Number>
- [.setCanvasSize\(size\)](#)
- [.fitCanvasSize\(minSize, fitScale\)](#)
- [.setViewBox\(position, size, isFit\)](#)
- [.getScale\(\)](#) ⇒ Number
- [.setScale\(scale\)](#)
- [.show\(element\)](#)
- [.hide\(element\)](#)
- [.appendChild\(srcElement, targetElement\)](#) ⇒ Element
- [.insertAfter\(srcElement, targetElement\)](#) ⇒ Element
- [.insertBefore\(srcElement, targetElement\)](#) ⇒ Element
- [.move\(element, offset\)](#) ⇒ Element
- [.moveCentroid\(element, position\)](#) ⇒ Element
- [.rotate\(element, angle\)](#) ⇒ Element
- [.resize\(element, offset\)](#) ⇒ Element
- [.resizeBox\(element, size\)](#) ⇒ Element
- [.clone\(element\)](#) ⇒ Element
- [.getBoundary\(element\)](#) ⇒ [Envelope](#)
- [.getElementById\(id\)](#) ⇒ Element
- [.getElementsByType\(shapeType, excludeType\)](#) ⇒ Array.<Element>
- [.getElementsByShapeld\(shapeld\)](#) ⇒ Array.<Element>
- [.getRelatedElementsFromEdge\(edgeElement\)](#) ⇒ Object
- [.getParent\(Element\)](#) ⇒ Element

- [.getChilds\(element\)](#) ⇒ Array
- [.getAllShapes\(\)](#) ⇒ Array
- [.getAllEdges\(\)](#) ⇒ Array
- [.getBBox\(element\)](#) ⇒ Object
- [.getRootBBox\(\)](#) ⇒ Object
- [.getRealRootBBox\(\)](#) ⇒ Object
- [.isSVG\(\)](#) ⇒ Boolean
- [.isVML\(\)](#) ⇒ Boolean
- [.setCustomData\(shapeElement, data\)](#)
- [.getCustomData\(shapeElement\)](#) ⇒ Object
- [.setExtCustomData\(shapeElement, data\)](#)
- [.getExtCustomData\(shapeElement\)](#) ⇒ Object
- [.toXML\(\)](#) ⇒ String
- [.toJSON\(\)](#) ⇒ Object
- [.loadXML\(xml\)](#) ⇒ Object
- [.loadJSON\(json\)](#) ⇒ Object
- [.undo\(\)](#)
- [.redo\(\)](#)
- [.getPrevEdges\(element\)](#) ⇒ Array.<Element>
- [.getNextEdges\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapes\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapeIds\(element\)](#) ⇒ Array.<String>
- [.getNextShapes\(element\)](#) ⇒ Array.<Element>
- [.getNextShapeIds\(element\)](#) ⇒ Array.<String>
- [.onDrawShape\(callbackFunc\)](#)
- [.onUndo\(callbackFunc\)](#)
- [.onRedo\(callbackFunc\)](#)
- [.onDivideLane\(callbackFunc\)](#)
- [.onDrawLabel\(callbackFunc\)](#)
- [.onLabelChanged\(callbackFunc\)](#)
- [.onBeforeLabelChange\(callbackFunc\)](#)
- [.onRedrawShape\(callbackFunc\)](#)
- [.onRemoveShape\(callbackFunc\)](#)
- [.onRotateShape\(callbackFunc\)](#)
- [.onMoveShape\(callbackFunc\)](#)
- [.onResizeShape\(callbackFunc\)](#)
- [.onBeforeConnectShape\(callbackFunc\)](#)
- [.onBeforeRemoveShape\(callbackFunc\)](#)
- [.onConnectShape\(callbackFunc\)](#)
- [.onDisconnectShape\(callbackFunc\)](#)
- [.onGroup\(callbackFunc\)](#)
- [.onUnGroup\(callbackFunc\)](#)
- [.onCollapsed\(callbackFunc\)](#)
- [.onExpanded\(callbackFunc\)](#)

◦ [.handler](#) : object

- [.EventHandler](#)
 - [new OG.handler.EventHandler\(renderer, config\)](#)
 - [.enableEditLabel\(element\)](#)

- [.setMovable\(element, isMovable\)](#)
- [.setConnectable\(element, guide, isConnectable\)](#)
- [.setResizable\(element, guide, isResizable\)](#)
- [.setClickSelectable\(element, isSelectable\)](#)
- [.setGroupDropable\(element\)](#)
- [.setDragSelectable\(isSelectable\)](#)
- [.setEnableHotKey\(isEnableHotKey\)](#)
- [.enableRootContextMenu\(\)](#)
- [.enableShapeContextMenu\(\)](#)
- [.selectShape\(element\)](#)
- [.selectShapes\(element\)](#)
- [.bringToFront\(\)](#)
- [.sendToBack\(\)](#)
- [.bringForward\(\)](#)
- [.sendBackward\(\)](#)
- [.deleteSelectedShape\(\)](#)
- [.changeShape\(\)](#)
- [.showProperty\(\)](#)
- [.selectAll\(\)](#)
- [.copySelectedShape\(\)](#)
- [.cutSelectedShape\(\)](#)
- [.pasteSelectedShape\(\)](#)
- [.duplicateSelectedShape\(\)](#)
- [.groupSelectedShape\(\)](#)
- [.ungroupSelectedShape\(\)](#)
- [.rotateSelectedShape\(angle\)](#)
- [.setLineWidthSelectedShape\(lineWidth\)](#)
- [.setLineColorSelectedShape\(lineColor\)](#)
- [.setLoopTypeSelectedShape\(lineType\)](#)
- [.setLineStyleSelectedShape\(lineStyle\)](#)
- [.setArrowStartSelectedShape\(arrowType\)](#)
- [.setArrowEndSelectedShape\(arrowType\)](#)
- [.setFillColorSelectedShape\(fillColor\)](#)
- [.setFontFamilySelectedShape\(fontFamily\)](#)
- [.setFontSizeSelectedShape\(fontSize\)](#)
- [.setFontColorSelectedShape\(fontColor\)](#)
- [.setFontWeightSelectedShape\(fontWeight\)](#)
- [.setFontStyleSelectedShape\(fontStyle\)](#)
- [.setTextDecorationSelectedShape\(textDecoration\)](#)
- [.setLabelDirectionSelectedShape\(labelDirection\)](#)
- [.setLabelAngleSelectedShape\(labelAngle\)](#)
- [.setLabelPositionSelectedShape\(labelPosition\)](#)
- [.setLabelVerticalSelectedShape\(verticalAlign\)](#)
- [.setLabelHorizontalSelectedShape\(horizontalAlign\)](#)
- [.setLabelSelectedShape\(label\)](#)
- [.setEdgeFromLabelSelectedShape\(label\)](#)
- [.setEdgeToLabelSelectedShape\(label\)](#)
- [.zoomIn\(\)](#)
- [.zoomOut\(\)](#)
- [.fitWindow\(\)](#)
- [.setConnectGuide\(element, isConnectable\)](#)

- [.layout](#) : object
- [.renderer](#) : object
 - [.IRenderer](#)
 - [new OG.renderer.IRenderer\(container, containerSize, backgroundColor, backgroundImage, config\)](#)
 - [.drawShape\(position, shape, size, style, id\)](#) ⇒ Element
 - [.drawGeom\(geometry, style\)](#) ⇒ Element
 - [.drawText\(position, text, size, style, id\)](#) ⇒ Element
 - [.drawImage\(position, imgSrc, size, style, id\)](#) ⇒ Element
 - [.drawEdge\(line, style, id, isSelf\)](#) ⇒ Element
 - [.drawLabel\(shapeElement, text, style\)](#) ⇒ Element
 - [.drawEdgeLabel\(shapeElement, text, type\)](#) ⇒ Element
 - [.redrawShape\(element, excludeEdgeId\)](#)
 - [.redrawConnectedEdge\(element\)](#)
 - [.connect\(fromTerminal, toTerminal, edge, style, label, preventTrigger\)](#) ⇒ Element
 - [.disconnect\(element\)](#)
 - [.drawDropOverGuide\(element\)](#)
 - [.drawGuide\(element\)](#) ⇒ Object
 - [.removeGuide\(element\)](#)
 - [.removeAllGuide\(\)](#)
 - [.removeConnectGuide\(element\)](#)
 - [.removeAllConnectGuide\(\)](#)
 - [.removeOtherConnectGuide\(element\)](#)
 - [.removeAllVirtualEdge\(\)](#)
 - [.drawEdgeGuide\(element\)](#) ⇒ Object
 - [.drawRubberBand\(position, size, style\)](#) ⇒ Element
 - [.removeRubberBand\(root\)](#)
 - [.drawDraggableGuide\(element\)](#) ⇒ Element
 - [.drawCollapseGuide\(element\)](#) ⇒ Element
 - [.removeCollapseGuide\(element\)](#)
 - [.group\(elements\)](#) ⇒ Element
 - [.ungroup\(groupElements\)](#) ⇒ Array.<Element>
 - [.addToGroup\(groupElement, elements\)](#)
 - [.collapse\(element\)](#)
 - [.expand\(element\)](#)
 - [.clear\(\)](#)
 - [.removeShape\(element\)](#)
 - [.remove\(element\)](#)
 - [.removeChild\(element\)](#)
 - [.getRootElement\(\)](#) ⇒ Element
 - [.getRootGroup\(\)](#) ⇒ Element
 - [.getElementByPoint\(position\)](#) ⇒ Element
 - [.getElementsByBBox\(envelope\)](#) ⇒ Array.<Element>
 - [.setAttr\(element, attribute\)](#)
 - [.getAttr\(element, attrName\)](#) ⇒ Object
 - [.setShapeStyle\(element, style\)](#)
 - [.toFront\(element\)](#)
 - [.toBack\(element\)](#)

- [.bringForward\(element\)](#)
- [.sendBackward\(element\)](#)
- [.getCanvasSize\(\)](#) ⇒ Array.<Number>
- [.setCanvasSize\(size\)](#)
- [.fitCanvasSize\(minSize, fitScale\)](#)
- [.setViewBox\(position, size, isFit\)](#)
- [.getScale\(\)](#) ⇒ Number
- [.setScale\(scale\)](#)
- [.show\(element\)](#)
- [.hide\(element\)](#)
- [.appendChild\(srcElement, targetElement\)](#) ⇒ Element
- [.insertAfter\(srcElement, targetElement\)](#) ⇒ Element
- [.insertBefore\(srcElement, targetElement\)](#) ⇒ Element
- [.move\(element, offset\)](#) ⇒ Element
- [.moveCentroid\(element, position\)](#) ⇒ Element
- [.rotate\(element, angle\)](#) ⇒ Element
- [.resize\(element, offset\)](#) ⇒ Element
- [.resizeBox\(element, size\)](#) ⇒ Element
- [.clone\(element\)](#) ⇒ Element
- [.getElementById\(id\)](#) ⇒ Element
- [.getElementsByType\(shapeType, excludeType\)](#) ⇒ Array.
<Element>
- [.getBBBox\(element\)](#) ⇒ Object
- [.getRootBBBox\(\)](#) ⇒ Object
- [.getRealRootBBBox\(\)](#) ⇒ Object
- [.getContainer\(\)](#) ⇒ Element
- [.isSVG\(\)](#) ⇒ Boolean
- [.isVML\(\)](#) ⇒ Boolean
- [.getPrevEdges\(element\)](#) ⇒ Array.<Element>
- [.getNextEdges\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapes\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapeIds\(element\)](#) ⇒ Array.<String>
- [.getNextShapes\(element\)](#) ⇒ Array.<Element>
- [.getNextShapeIds\(element\)](#) ⇒ Array.<String>
- [.getConnectGuideElements\(Element\)](#) ⇒ Array
- [.isTopGroup\(Element\)](#) ⇒ boolean
- [.getParent\(element\)](#) ⇒ Element
- [.getChilts\(element\)](#) ⇒ Array
- [.isGroup\(element\)](#) ⇒ boolean
- [.getAllShapes\(\)](#) ⇒ Array
- [.getAllEdges\(\)](#) ⇒ Array
- [.getAllNotEdges\(\)](#) ⇒ Array
- [.isEdge\(\)](#) ⇒ boolean
- [.isShape\(\)](#) ⇒ boolean
- [.initHistory\(\)](#)
- [.addHistory\(\)](#)
- [.undo\(\)](#)
- [.redo\(\)](#)

- [.RaphaelRenderer](#) \Leftarrow [IRenderer](#)
 - [new OG.renderer.RaphaelRenderer\(container, containerSize, backgroundColor, backgroundImage, config\)](#)
 - [.drawHtml\(position, html, size, style, id\)](#) \Rightarrow Element
 - [.getPointOfInflectionFromEdge\(\)](#)
 - [.reconnect\(edge\)](#) \Rightarrow Element
 - [.disconnectOneWay\(element, connectDirection\)](#)
 - [.drawStickGuide\(element, position\)](#)
 - [.setTextListInController\(element, textList\)](#)
 - [.getTextListInController\(element\)](#)
 - [.getConnectGuideElements\(Element\)](#) \Rightarrow Array
 - [.getNotConnectGuideElements\(Element\)](#) \Rightarrow Array
 - [.removeConnectGuide\(element\)](#)
 - [.removeAllConnectGuide\(\)](#)
 - [.removeOtherConnectGuide\(element\)](#)
 - [.getSpots\(element\)](#) \Rightarrow Array
 - [.getCircleSpots\(element\)](#) \Rightarrow Array
 - [.createVirtualSpot\(x, x, element\)](#) \Rightarrow Element
 - [.getVirtualSpot\(element\)](#) \Rightarrow Element
 - [.removeVirtualSpot\(element\)](#) \Rightarrow Element
 - [.selectSpot\(선택한\)](#)
 - [.getChildNodes\(element\)](#) \Rightarrow Array
 - [.trimEdge\(element\)](#)
 - [.trimConnectInnerVertice\(element\)](#) \Rightarrow Element
 - [.trimConnectIntersection\(element\)](#) \Rightarrow Element
 - [.getBoundary\(element\)](#) \Rightarrow [Envelope](#)
 - [.setHighlight\(element, highlight\)](#)
 - [.removeHighlight\(element, highlight\)](#)
 - [.createTerminalString\(Element, point\)](#) \Rightarrow String
 - [.createDefaultTerminalString\(Element\)](#) \Rightarrow String
 - [.toFrontEdges\(\)](#)
 - [.removeAllEdgeGuide\(\)](#)
 - [.createVirtualEdge\(x, x, targetEle\)](#) \Rightarrow Element
 - [.updateVirtualEdge\(x, x\)](#)
 - [.getTargetfromVirtualEdge\(x, x\)](#)
 - [.removeAllVirtualEdge\(\)](#)
 - [.isLane\(Element\)](#) \Rightarrow boolean
 - [.isPool\(Element\)](#) \Rightarrow boolean
 - [.isScopeActivity\(Element\)](#) \Rightarrow boolean
 - [.isHorizontalLane\(Element\)](#) \Rightarrow boolean
 - [.isVerticalLane\(Element\)](#) \Rightarrow boolean
 - [.isHorizontalPool\(Element\)](#) \Rightarrow boolean
 - [.isVerticalPool\(Element\)](#) \Rightarrow boolean
 - [.getChildLane\(Element\)](#) \Rightarrow Array
 - [.enableDivideCount\(Element\)](#) \Rightarrow Number
 - [.getExceptTitleLaneArea\(Element, boundary\)](#)
 - [.divideLane\(Element, quarterOrder\)](#)
 - [.getBaseLanes\(Element\)](#) \Rightarrow Array
 - [.getRootLane\(Element\)](#) \Rightarrow Element

- [.getIndexOfLane\(Element\)](#) ⇒ Number
- [.getDepthOfLane\(Element\)](#) ⇒ Number
- [.reEstablishLane\(Element\)](#)
- [.getBoundaryOfElements\(elements\)](#) ⇒ [Envelope](#)
- [.getNearestBaseLaneIndexAsDirection\(Element, direction\)](#) ⇒ Number
- [.getBoundaryOfInnerShapesGroup\(Element\)](#) ⇒ [Envelope](#)
- [.getSmallestBaseLane\(Element, baseLane\)](#)
- [.resizeLane\(Element, offset\)](#)
- [.removeLaneShape\(Element\)](#)
- [.getInnerShapesOfLane\(Element\)](#)
- [.fitLaneOrder\(Element\)](#)
- [.getRootGroupOfShape\(Element\)](#) ⇒ Element
- [.checkBridgeEdge\(Element\)](#)
- [.checkAllBridgeEdge\(\)](#)
- [.getInnerShapesOfGroup\(Element\)](#)
- [.getFrontForCoordinate\(point\)](#) ⇒ Element
- [.getFrontForBoundary\(boundary\)](#) ⇒ Element
- [.trimEdgeDirection\(Edge, FromShape, ToShape\)](#) ⇒ Element
- [.putInnerShapeToPool\(Element\)](#) ⇒ Element
- [.setDropablePool\(Element\)](#) ⇒ Element
- [.offDropablePool\(\)](#)
- [.drawShape\(position, shape, size, style, id\)](#) ⇒ Element
- [.drawGeom\(geometry, style\)](#) ⇒ Element
- [.drawText\(position, text, size, style, id\)](#) ⇒ Element
- [.drawImage\(position, imgSrc, size, style, id\)](#) ⇒ Element
- [.drawEdge\(line, style, id, isSelf\)](#) ⇒ Element
- [.drawLabel\(shapeElement, text, style\)](#) ⇒ Element
 - [~getCenterOfEdge\(element\)](#) ⇒ OG.Coordinate
- [.drawEdgeLabel\(shapeElement, text, type\)](#) ⇒ Element
- [.redrawShape\(element, excludeEdgeld\)](#)
- [.redrawConnectedEdge\(element\)](#)
- [.connect\(fromTerminal, toTerminal, edge, style, label, preventTrigger\)](#) ⇒ Element
- [.disconnect\(element\)](#)
- [.drawDropOverGuide\(element\)](#)
- [.drawGuide\(element\)](#) ⇒ Object
- [.removeGuide\(element\)](#)
- [.removeAllGuide\(\)](#)
- [.drawEdgeGuide\(element\)](#) ⇒ Object
- [.drawRubberBand\(position, size, style\)](#) ⇒ Element
- [.removeRubberBand\(root\)](#)
- [.drawDraggableGuide\(element\)](#) ⇒ Element
- [.drawCollapseGuide\(element\)](#) ⇒ Element
- [.removeCollapseGuide\(element\)](#)
- [.group\(elements\)](#) ⇒ Element
- [.ungroup\(groupElements\)](#) ⇒ Array.<Element>
- [.addToGroup\(groupElement, elements\)](#)

- [.collapse\(element\)](#)
- [.expand\(element\)](#)
- [.clear\(\)](#)
- [.removeShape\(element\)](#)
- [.remove\(element\)](#)
- [.removeChild\(element\)](#)
- [.getRootElement\(\)](#) ⇒ Element
- [.getRootGroup\(\)](#) ⇒ Element
- [.getElementByPoint\(position\)](#) ⇒ Element
- [.getElementsByBBox\(envelope\)](#) ⇒ Array.<Element>
- [.setAttr\(element, attribute\)](#)
- [.getAttr\(element, attrName\)](#) ⇒ Object
- [.setShapeStyle\(element, style\)](#)
- [.toFront\(element\)](#)
- [.toBack\(element\)](#)
- [.bringForward\(element\)](#)
- [.sendBackward\(element\)](#)
- [.getCanvasSize\(\)](#) ⇒ Array.<Number>
- [.setCanvasSize\(size\)](#)
- [.fitCanvasSize\(minSize, fitScale\)](#)
- [.setViewBox\(position, size, isFit\)](#)
- [.getScale\(\)](#) ⇒ Number
- [.setScale\(scale\)](#)
- [.show\(element\)](#)
- [.hide\(element\)](#)
- [.appendChild\(srcElement, targetElement\)](#) ⇒ Element
- [.insertAfter\(srcElement, targetElement\)](#) ⇒ Element
- [.insertBefore\(srcElement, targetElement\)](#) ⇒ Element
- [.move\(element, offset\)](#) ⇒ Element
- [.moveCentroid\(element, position\)](#) ⇒ Element
- [.rotate\(element, angle\)](#) ⇒ Element
- [.resize\(element, offset\)](#) ⇒ Element
- [.resizeBox\(element, size\)](#) ⇒ Element
- [.clone\(element\)](#) ⇒ Element
- [.getElementById\(id\)](#) ⇒ Element
- [.getElementsByType\(shapeType, excludeType\)](#) ⇒ Array.<Element>
- [.getBBox\(element\)](#) ⇒ Object
- [.getRootBBox\(\)](#) ⇒ Object
- [.getRealRootBBox\(\)](#) ⇒ Object
- [.getContainer\(\)](#) ⇒ Element
- [.isSVG\(\)](#) ⇒ Boolean
- [.isVML\(\)](#) ⇒ Boolean
- [.getPrevEdges\(element\)](#) ⇒ Array.<Element>
- [.getNextEdges\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapes\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapeIds\(element\)](#) ⇒ Array.<String>
- [.getNextShapes\(element\)](#) ⇒ Array.<Element>
- [.getNextShapeIds\(element\)](#) ⇒ Array.<String>

- [.isTopGroup\(Element\)](#) ⇒ boolean
- [.getParent\(element\)](#) ⇒ Element
- [.getChilDs\(element\)](#) ⇒ Array
- [.isGroup\(element\)](#) ⇒ boolean
- [.getAllShapes\(\)](#) ⇒ Array
- [.getAllEdges\(\)](#) ⇒ Array
- [.getAllNotEdges\(\)](#) ⇒ Array
- [.isEdge\(\)](#) ⇒ boolean
- [.isShape\(\)](#) ⇒ boolean
- [.initHistory\(\)](#)
- [.addHistory\(\)](#)
- [.undo\(\)](#)
- [.redo\(\)](#)

○ [.marker](#) : object

- [.ArrowMarker](#) ⇐ [IMarker](#)
 - [new OG.marker.ArrowMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *
- [.CircleMarker](#) ⇐ [IMarker](#)
 - [new OG.marker.CircleMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *
- [.IMarker](#)
 - [new OG.marker.IMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *
- [.RectangleMarker](#) ⇐ [IMarker](#)
 - [new OG.marker.RectangleMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *
- [.SwitchLMarker](#) ⇐ [IMarker](#)
 - [new OG.marker.SwitchLMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *
- [.SwitchRMarker](#) ⇐ [IMarker](#)

- [new OG.marker.SwitchRMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *
 - [.SwitchXMarker](#) ⇐ [IMarker](#)
 - [new OG.marker.SwitchXMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *
- [.pattern](#) : object
 - [.HatchedPattern](#) ⇐ [IPattern](#)
 - [new OG.pattern.HatchedPattern\(\)](#)
 - [.PATTERN_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createPattern\(\)](#) ⇒ *
 - [.IPattern](#)
 - [new OG.pattern.IPattern\(\)](#)
 - [.PATTERN_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createPattern\(\)](#) ⇒ *
 - [.RectPattern](#) ⇐ [IPattern](#)
 - [new OG.pattern.RectPattern\(\)](#)
 - [.PATTERN_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createPattern\(\)](#) ⇒ *
- [.shape](#) : object
 - [.CircleShape](#) ⇐ [GeomShape](#)
 - [new OG.shape.CircleShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean

- [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.clone\(\)](#) ⇒ [IShape](#)
 - [.createShape\(\)](#) ⇒ *
- [.EdgeShape](#) ⇐ [IShape](#)
 - [new OG.shape.EdgeShape\(from, to, label, fromLabel, toLabel\)](#)
 - [.from](#) : Array.<Number>
 - [.to](#) : Array.<Number>
 - [.fromLabel](#) : String
 - [.toLabel](#) : String
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.EllipseShape](#) ⇐ [GeomShape](#)
 - [new OG.shape.EllipseShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean

- [.ENABLE FROM](#) : Boolean
- [.ENABLE TO](#) : Boolean
- [.SELF CONNECTABLE](#) : Boolean
- [.CONNECT CLONEABLE](#) : Boolean
- [.CONNECT REQUIRED](#) : Boolean
- [.CONNECT STYLE CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.clone\(\)](#) ⇒ [IShape](#)
- [.createShape\(\)](#) ⇒ *
- [.GeomShape](#) ⇐ [IShape](#)
 - [new OG.shape.GeomShape\(\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE FROM](#) : Boolean
 - [.ENABLE TO](#) : Boolean
 - [.SELF CONNECTABLE](#) : Boolean
 - [.CONNECT CLONEABLE](#) : Boolean
 - [.CONNECT REQUIRED](#) : Boolean
 - [.CONNECT STYLE CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.GroupShape](#) ⇐ [IShape](#)
 - [new OG.shape.GroupShape\(label\)](#)
 - [.GROUP DROPPABLE](#) : Boolean
 - [.GROUP COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean

- [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE FROM](#) : Boolean
 - [.ENABLE TO](#) : Boolean
 - [.SELF CONNECTABLE](#) : Boolean
 - [.CONNECT CLONEABLE](#) : Boolean
 - [.CONNECT REQUIRED](#) : Boolean
 - [.CONNECT STYLE CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.HorizontalLaneShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.HorizontalLaneShape\(label\)](#)
 - [.GROUP DROPABLE](#) : Boolean
 - [.GROUP COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE FROM](#) : Boolean
 - [.ENABLE TO](#) : Boolean
 - [.SELF CONNECTABLE](#) : Boolean
 - [.CONNECT CLONEABLE](#) : Boolean
 - [.CONNECT REQUIRED](#) : Boolean
 - [.CONNECT STYLE CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.HorizontalPoolShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.HorizontalPoolShape\(label\)](#)
 - [.GROUP DROPABLE](#) : Boolean
 - [.GROUP COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE ID](#) : String
 - [.geom](#) : [Geometry](#)

- [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE FROM](#) : Boolean
 - [.ENABLE TO](#) : Boolean
 - [.SELF CONNECTABLE](#) : Boolean
 - [.CONNECT CLONEABLE](#) : Boolean
 - [.CONNECT REQUIRED](#) : Boolean
 - [.CONNECT STYLE CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.HtmlShape](#) ⇐ [IShape](#)
 - [new OG.shape.HtmlShape\(html, label\)](#)
 - [.html](#) : String
 - [.angle](#) : Number
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE FROM](#) : Boolean
 - [.ENABLE TO](#) : Boolean
 - [.SELF CONNECTABLE](#) : Boolean
 - [.CONNECT CLONEABLE](#) : Boolean
 - [.CONNECT REQUIRED](#) : Boolean
 - [.CONNECT STYLE CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.IShape](#)
 - [new OG.shape.IShape\(\)](#)
 - [.TYPE](#) : String

- [.SHAPE_ID](#) : String
- [.geom](#) : [Geometry](#)
- [.label](#) : String
- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE FROM](#) : Boolean
- [.ENABLE TO](#) : Boolean
- [.SELF CONNECTABLE](#) : Boolean
- [.CONNECT CLONEABLE](#) : Boolean
- [.CONNECT REQUIRED](#) : Boolean
- [.CONNECT STYLE CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)
- [.ImageShape](#) ⇐ [IShape](#)
 - [new OG.shape.ImageShape\(image, label\)](#)
 - [.image](#) : String
 - [.angle](#) : Number
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE FROM](#) : Boolean
 - [.ENABLE TO](#) : Boolean
 - [.SELF CONNECTABLE](#) : Boolean
 - [.CONNECT CLONEABLE](#) : Boolean
 - [.CONNECT REQUIRED](#) : Boolean
 - [.CONNECT STYLE CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.RectangleShape](#) ⇐ [GeomShape](#)

- [new OG.shape.RectangleShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.clone\(\)](#) ⇒ [IShape](#)
 - [.createShape\(\)](#) ⇒ *
- [.SpotShape](#) ⇐ [GeomShape](#)
 - [new OG.shape.SpotShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.clone\(\)](#) ⇒ [IShape](#)
 - [.createShape\(\)](#) ⇒ *
- [.TextShape](#) ⇐ [IShape](#)

- [new OG.shape.TextShape\(text\)](#)
- [.text](#) : String
- [.angle](#) : Number
- [.TYPE](#) : String
- [.SHAPE_ID](#) : String
- [.geom](#) : [Geometry](#)
- [.label](#) : String
- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

- [.VerticalLaneShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.VerticalLaneShape\(label\)](#)
 - [.GROUP_DROPABLE](#) : Boolean
 - [.GROUP_COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array

- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)
- [.VerticalPoolShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.VerticalPoolShape\(label\)](#)
 - [.GROUP_DROPPABLE](#) : Boolean
 - [.GROUP_COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.bpmn](#) : object
- [.elec](#) : object

OG.common : object

Kind: static namespace of [OG](#)

- [.common](#) : object
 - [.Constants](#)
 - [new OG.common.Constants\(\)](#)
 - [.GEOM_TYPE](#)
 - [.GEOM_NAME](#)
 - [.NUM_PRECISION](#)
 - [.NODE_TYPE](#)
 - [.SHAPE_TYPE](#)
 - [.EDGE_TYPE](#)
 - [.LABEL_SUFFIX](#)
 - [.LABEL_EDITOR_SUFFIX](#)
 - [.FROM_LABEL_SUFFIX](#)

- [.TO LABEL SUFFIX](#)
- [.RUBBER BAND ID](#)
- [.RUBBER BAND TOLERANCE](#)
- [.GUIDE SUFFIX](#)
- [.COLLAPSE SUFFIX](#)
- [.LOOPTYPE SUFFIX](#)
- [.TASKTYPE SUFFIX](#)
- [.INCLUSION SUFFIX](#)
- [.STATUS SUFFIX](#)
- [.EXCEPTIONTYPE SUFFIX](#)
- [.MOVE SNAP SIZE](#)
- [.DROP OVER BBOX SUFFIX](#)
- [.TERMINAL SUFFIX](#)
- [.TERMINAL](#)
- [.MARKER TEMP NODE](#)
- [.PATTERN TEMP NODE](#)
- [.MARKER DEFS SUFFIX](#)
- [.ORIGINAL NODE](#)
- [.CONNECT GUIDE EVENT AREA](#)
- [.CONNECT GUIDE SUFFIX](#)
- [.CurveUtil](#)
 - [new OG.common.CurveUtil\(\)](#)
 - [.CatmullRomSpline\(points\)](#) ⇒ Object
 - [.Bezier\(points\)](#) ⇒ Object
 - [.BSpline\(points, order\)](#) ⇒ Object
- [.HashMap](#)
 - [new OG.common.HashMap\(jsonObject\)](#)
 - [.map](#) : Object
 - [.put\(key, value\)](#)
 - [.get\(key\)](#) ⇒ Object
 - [.containsKey\(key\)](#) ⇒ Boolean
 - [.containsValue\(value\)](#) ⇒ Boolean
 - [.isEmpty\(\)](#) ⇒ Boolean
 - [.clear\(\)](#)
 - [.remove\(key\)](#)
 - [.keys\(\)](#) ⇒ Array.<String>
 - [.values\(\)](#) ⇒ Array.<Object>
 - [.size\(\)](#) ⇒ Number
 - [.toString\(\)](#) ⇒ String
- [.JSON](#)
 - [new OG.common.JSON\(\)](#)
 - [.encode](#) ⇒ String
 - [.decode](#) ⇒ Object
 - [.encodeDate\(d\)](#) ⇒ String
- [.Util](#)

- [new OG.common.Util\(\)](#)
- [.extend](#) ⇒ function
- [.clone\(obj\)](#) ⇒ Object
- [.round\(val\)](#) ⇒ Number
- [.roundPrecision\(val, precision\)](#) ⇒ Number
- [.roundGrid\(val, snapSize\)](#) ⇒ Number
- [.apply\(obj, config, defaults\)](#) ⇒ Object

common.Constants

Kind: static class of [common](#)

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Constants](#)
 - [new OG.common.Constants\(\)](#)
 - [.GEOM_TYPE](#)
 - [.GEOM_NAME](#)
 - [.NUM_PRECISION](#)
 - [.NODE_TYPE](#)
 - [.SHAPE_TYPE](#)
 - [.EDGE_TYPE](#)
 - [.LABEL_SUFFIX](#)
 - [.LABEL_EDITOR_SUFFIX](#)
 - [.FROM_LABEL_SUFFIX](#)
 - [.TO_LABEL_SUFFIX](#)
 - [.RUBBER_BAND_ID](#)
 - [.RUBBER_BAND_TOLERANCE](#)
 - [.GUIDE_SUFFIX](#)
 - [.COLLAPSE_SUFFIX](#)
 - [.LOOPTYPE_SUFFIX](#)
 - [.TASKTYPE_SUFFIX](#)
 - [.INCLUSION_SUFFIX](#)
 - [.STATUS_SUFFIX](#)
 - [.EXCEPTIONTYPE_SUFFIX](#)
 - [.MOVE_SNAP_SIZE](#)
 - [.DROP_OVER_BBOX_SUFFIX](#)
 - [.TERMINAL_SUFFIX](#)
 - [.TERMINAL](#)
 - [.MARKER_TEMP_NODE](#)
 - [.PATTERN_TEMP_NODE](#)
 - [.MARKER_DEFS_SUFFIX](#)
 - [.ORIGINAL_NODE](#)
 - [.CONNECT_GUIDE_EVENT_AREA](#)
 - [.CONNECT_GUIDE_SUFFIX](#)

`new OG.common.Constants()`

공통 상수 정의 Javascript 클래스

`Constants.GEOM_TYPE`

공간 기하 객체 타입 정의

Kind: static property of [Constants](#)

Constants.GEOM_NAME

공간 기하 객체 타입-이름 매핑

Kind: static property of [Constants](#)

Constants.NUM_PRECISION

숫자 반올림 소숫점 자리수

Kind: static property of [Constants](#)

Constants.NODE_TYPE

캔버스 노드 타입 정의

Kind: static property of [Constants](#)

Constants.SHAPE_TYPE

Shape 타입 정의

Kind: static property of [Constants](#)

Constants.EDGE_TYPE

Edge 타입 정의

Kind: static property of [Constants](#)

Constants.LABEL_SUFFIX

라벨 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.LABEL_EDITOR_SUFFIX

라벨 에디터 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.FROM_LABEL_SUFFIX

시작점 라벨 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.TO_LABEL_SUFFIX

끝점 라벨 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.RUBBER_BAND_ID

Rectangle 모양의 마우스 드래그 선택 박스 영역

Kind: static property of [Constants](#)

Constants.RUBBER_BAND_TOLERANCE

Rubber Band 허용 오차

Kind: static property of [Constants](#)

Constants.GUIDE_SUFFIX

Move & Resize 용 가이드 ID 의 suffix 정의

Kind: static property of [Constants](#)

Constants.COLLAPSE_SUFFIX

Collapse & Expand 용 가이드 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.LOOPTYPE_SUFFIX

LoopType 용 가이드 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.TASKTYPE_SUFFIX

TaskType 용 가이드 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.INCLUSION_SUFFIX

TaskType 용 가이드 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.STATUS_SUFFIX

STATUS 용 가이드 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.EXCEPTIONTYPE_SUFFIX

EXCEPTIONTYPE 용 가이드 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.MOVE_SNAP_SIZE

Shape Move & Resize 시 이동 간격

Kind: static property of [Constants](#)

Constants.DROP_OVER_BBOX_SUFFIX

Edge 연결할때 Drop Over 가이드 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.TERMINAL_SUFFIX

Shape – Edge 와의 연결 포인트 터미널 ID의 suffix 정의

Kind: static property of [Constants](#)

Constants.TERMINAL

Shape – Edge 와의 연결 포인트

Kind: static property of [Constants](#)

Constants.MARKER_TEMP_NODE

마커 등록을 위한 임시 노드 아이디

Kind: static property of [Constants](#)

Constants.PATTERN_TEMP_NODE

패턴 등록을 위한 임시 노드 아이디

Kind: static property of [Constants](#)

Constants.MARKER_DEFS_SUFFIX

캔버스의 마커 데피니션 suffix 정의

Kind: static property of [Constants](#)

Constants.ORIGINAL_NODE

Shape 에서 마커가 그려질 경우 원본 노드 suffix 정의

Kind: static property of [Constants](#)

Constants.CONNECT_GUIDE_EVENT_AREA

Element 의 커넥트 가이드 이벤트 보장영역의 정의

Kind: static property of [Constants](#)

Constants.CONNECT_GUIDE_SUFFIX

Element 의 커넥트 가이드 suffix 정의

Kind: static property of [Constants](#)

common.CurveUtil

Kind: static class of [common](#)

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.CurveUtil](#)
 - [new OG.common.CurveUtil\(\)](#)
 - [.CatmullRomSpline\(points\)](#) ⇒ Object
 - [.Bezier\(points\)](#) ⇒ Object
 - [.BSpline\(points, order\)](#) ⇒ Object

`new OG.common.CurveUtil()`

곡선(Curve) 알고리즘을 구현한 Javascript 클래스

`CurveUtil.CatmullRomSpline(points)` ⇒ Object

주어진 좌표 Array 에 대해 Cubic Catmull-Rom spline Curve 좌표를 계산하는 함수를 반환한다.
모든 좌표를 지나는 커브를 계산한다.

Kind: static method of [CurveUtil](#)

Returns: Object - t 값에 의해 X, Y 좌표를 구하는 함수와 maxT 값을 반환

Param Type

Description

points Array 좌표 Array (예, [[x1,y1], [x2,y2], [x3,y3], [x4,y4]])

Example

```
var points = [[2, 2], [2, -2], [-2, 2], [-2, -2]],
    cmRomSpline = OG.CurveUtil.CatmullRomSpline(points), t, curve = [];

// t 는 0 ~ maxT 의 값으로, t 값의 증분값이 작을수록 세밀한 Curve 를 그린다.
for(t = 0; t <= cmRomSpline.maxT; t += 0.1) {
    curve.push([cmRomSpline.getX(t), cmRomSpline.getY(t)]);
}
```

`CurveUtil.Bezier(points)` ⇒ Object

주어진 좌표 Array (좌표1, 컨트롤포인트1, 컨트롤포인트2, 좌표2 ...) 에 대해 Cubic Bezier Curve 좌표를 계산하는 함수를 반환한다.

Array 갯수는 $3 * K + 1$ 이어야 한다.

예) 좌표1, 컨트롤포인트1, 컨트롤포인트2, 좌표2, 컨트롤포인트1, 컨트롤포인트2, 좌표3 ...

Kind: static method of [CurveUtil](#)

Returns: Object - t 값에 의해 X, Y 좌표를 구하는 함수와 maxT 값을 반환

Param Type

Description

points Array 좌표 Array (예, [[x1,y1], [cp_x1,cp_y1], [cp_x2,cp_y2], [x2,y4]])

Example

```
var points = [[2, 1], [1, 3], [-1, -1], [-2, 1]],
    bezier = OG.CurveUtil.Bezier(points), t, curve = [];

// t 는 0 ~ maxT 의 값으로, t 값의 증분값이 작을수록 세밀한 Curve 를 그린다.
for(t = 0; t <= bezier.maxT; t += 0.1) {
    curve.push([bezier.getX(t), bezier.getY(t)]);
}
```

CurveUtil.BSpline(points, order) ⇒ Object

주어진 좌표 Array (시작좌표, 컨트롤포인트1, 컨트롤포인트2, ..., 끝좌표) 에 대해 B-Spline Curve 좌표를 계산하는 함수를 반환한다.

Kind: static method of [CurveUtil](#)

Returns: Object - t 값에 의해 X, Y 좌표를 구하는 함수와 maxT 값을 반환

| Param | Type | Description |
|--------|--------|--|
| points | Array | 좌표 Array (예, [[x1,y1], [x2,y2], [x3,y3], [x4,y4]]) |
| order | Number | Order of the B-spline curve. |

Example

```
var points = [[2, 1], [1, 3], [-1, -1], [-2, 1]],
    bspline = OG.CurveUtil.BSpline(points), t, curve = [];

// t 는 0 ~ maxT 의 값으로, t 값의 증분값이 작을수록 세밀한 Curve 를 그린다.
for(t = 0; t <= bspline.maxT; t += 0.1) {
    curve.push([bspline.getX(t), bspline.getY(t)]);
}
```

common.HashMap

Kind: static class of [common](#)

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.HashMap](#)
 - [new OG.common.HashMap\(jsonObject\)](#)
 - [.map](#) : Object
 - [.put\(key, value\)](#)
 - [.get\(key\)](#) ⇒ Object
 - [.containsKey\(key\)](#) ⇒ Boolean
 - [.containsValue\(value\)](#) ⇒ Boolean
 - [.isEmpty\(\)](#) ⇒ Boolean
 - [.clear\(\)](#)
 - [.remove\(key\)](#)
 - [.keys\(\)](#) ⇒ Array.<String>
 - [.values\(\)](#) ⇒ Array.<Object>
 - [.size\(\)](#) ⇒ Number
 - [.toString\(\)](#) ⇒ String

`new OG.common.HashMap(jsonObject)`

HashMap 구현 Javascript 클래스

| Param | Type | Description |
|------------|--------|------------------------|
| jsonObject | Object | key:value 매핑 JSON 오브젝트 |

Example

```
var map1 = new OG.common.HashMap({
  'key1': 'value1',
  'key2': 'value2'
});

console.log(map1.get('key1'));

var map2 = new OG.common.HashMap();
map2.put('key1', 'value1');
map2.put('key2', 'value2');

console.log(map2.get('key1'));
```

`hashMap.map` : Object

key:value 매핑 JSON 오브젝트

Kind: instance property of [HashMap](#)

`hashMap.put(key, value)`

key : value 를 매핑한다.

Kind: instance method of [HashMap](#)

| Param | Type | Description |
|-------|--------|-------------|
| key | String | 키 |
| value | Object | 값 |

`hashMap.get(key)` ⇒ Object

key 에 대한 value 를 반환한다.

Kind: instance method of [HashMap](#)

Returns: Object – 값

| Param | Type | Description |
|-------|--------|-------------|
| key | String | 키 |

`hashMap.containsKey(key)` ⇒ Boolean

주어진 key 를 포함하는지 여부를 반환한다.

Kind: instance method of [HashMap](#)

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|-----|--------|---|
| key | String | 키 |
|-----|--------|---|

`hashMap.containsValue(value)` ⇒ Boolean

주어진 value 를 포함하는지 여부를 반환한다.

Kind: instance method of [HashMap](#)

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|-------|--------|---|
| value | Object | 값 |
|-------|--------|---|

`hashMap.isEmpty()` ⇒ Boolean

Empty 여부를 반환한다.

Kind: instance method of [HashMap](#)

`hashMap.clear()`

매핑정보를 클리어한다.

Kind: instance method of [HashMap](#)

`hashMap.remove(key)`

주어진 key 의 매핑정보를 삭제한다.

Kind: instance method of [HashMap](#)

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|-----|--------|---|
| key | String | 키 |
|-----|--------|---|

`hashMap.keys()` ⇒ Array.<String>

key 목록을 반환한다.

Kind: instance method of [HashMap](#)

Returns: Array.<String> - 키목록

`hashMap.values()` ⇒ Array.<Object>

value 목록을 반환한다.

Kind: instance method of [HashMap](#)

Returns: Array.<Object> - 값목록

`hashMap.size()` ⇒ Number

매핑된 key:value 갯수를 반환한다.

Kind: instance method of [HashMap](#)

`hashMap.toString()` ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [HashMap](#)

Returns: String – 프라퍼티 정보

common.JSON

Kind: static class of [common](#)

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.JSON](#)
 - [new OG.common.JSON\(\)](#)
 - [.encode](#) ⇒ String
 - [.decode](#) ⇒ Object
 - [.encodeDate\(d\)](#) ⇒ String

`new OG.common.JSON()`

Modified version of Douglas Crockford's json.js that doesn't mess with the Object prototype
<http://www.json.org/js.html>

`json.encode` ⇒ String

Encodes an Object, Array or other value

Kind: instance property of [JSON](#)

Returns: String – The JSON string

| Param | Type | Description |
|-------|-------|------------------------|
| o | Mixed | The variable to encode |

`json.decode` ⇒ Object

Decodes (parses) a JSON string to an object. If the JSON is invalid, this function throws a `SyntaxError` unless the `safe` option is set.

Kind: instance property of [JSON](#)

Returns: Object – The resulting object

| Param | Type | Description |
|-------|--------|-----------------|
| json | String | The JSON string |

`json.encodeDate(d)` ⇒ String

Encodes a Date. This returns the actual string which is inserted into the JSON string as the literal expression. **The returned value includes enclosing double quotation marks.**

The default return format is "yyyy-mm-ddThh:mm:ss".

To override this:

```
OG.common.JSON.encodeDate = function(d) {  
  return d.format('"Y-m-d"');  
};
```

Kind: instance method of [JSON](#)

Returns: String – The string literal to use in a JSON string.

| Param Type | Description |
|------------|-------------------------|
| d | Date The Date to encode |

common.Util

Kind: static class of [common](#)

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Util](#)
 - [new OG.common.Util\(\)](#)
 - [.extend](#) ⇒ function
 - [.clone\(obj\)](#) ⇒ Object
 - [.round\(val\)](#) ⇒ Number
 - [.roundPrecision\(val, precision\)](#) ⇒ Number
 - [.roundGrid\(val, snapSize\)](#) ⇒ Number
 - [.apply\(obj, config, defaults\)](#) ⇒ Object

new OG.common.Util()

공통 유틸리티 Javascript 클래스

Util.extend ⇒ function

Extends one class to create a subclass and optionally overrides members with the passed literal. This method also adds the function "override()" to the subclass that can be used to override members of the class.

For example, to create a subclass of Ext GridPanel:

```
MyGridPanel = Ext.extend(Ext.grid.GridPanel, {  
  constructor: function(config) {
```

```
//      Create configuration for this Grid.
var store = new Ext.data.Store({...});
var colModel = new Ext.grid.ColumnModel({...});

//      Create a new config object containing our computed properties
//      *plus* whatever was in the config parameter.
config = Ext.apply({
store: store,
colModel: colModel
}, config);

MyGridPanel.superclass.constructor.call(this, config);

//      Your postprocessing here
},

yourMethod: function() {
// etc.
}
});
</code></pre>
```

This function also supports a 3-argument call in which the subclass's constructor is passed as an argument. In this form, the parameters are as follows:

- **subclass** : Function
The subclass constructor.
- **superclass** : Function
The constructor of class being extended
- **overrides** : Object
A literal with members which are copied into the subclass's prototype, and are therefore shared among all instances of the new class.

Kind: static property of [Util](#)

Returns: function - The subclass constructor from the overrides parameter, or a generated one if not provided.

| Param | Type | Description |
|------------|----------|--|
| superclass | function | The constructor of class being extended. |

A literal with members which are copied into the subclass's prototype, and are therefore shared between all instances of the new class.

overrides Object This may contain a special member named `constructor`. This is used to define the constructor of the new class, and is returned. If this property is *not* specified, a constructor is generated and returned which just calls the superclass's constructor passing on its parameters.

It is essential that you call the superclass constructor in any provided constructor. See example code.

`Util.clone(obj) ⇒ Object`

Object 를 복사한다.

Kind: static method of [Util](#)

Returns: Object - 복사된 Object

| Param | Type | Description |
|-------|--------|-------------|
| obj | Object | 복사할 Object |

`Util.round(val) ⇒ Number`

디폴트로 지정된 소숫점 자리수로 Round 한 값을 반환한다.

Kind: static method of [Util](#)

Returns: Number - 지정한 소숫점 자리수에 따른 반올림 값

| Param | Type | Description |
|-------|--------|-------------|
| val | Number | 반올림할 값 |

Util.roundPrecision(val, precision) ⇒ Number

입력된 숫자값을 지정된 소숫점 자리수로 Round해서 값을 리턴한다.

Kind: static method of [Util](#)

Returns: Number - 지정한 소숫점 자리수에 따른 반올림 값

| Param | Type | Description |
|-----------|--------|-------------|
| val | Number | 반올림할 값 |
| precision | Number | 소숫점 자리수 |

Example

```
OG.Util.roundPrecision(300.12345678, 3);  
Result ) 300.123
```

Util.roundGrid(val, snapSize) ⇒ Number

Shape Move & Resize 이동 간격으로 Round 한 값을 반환한다.

Kind: static method of [Util](#)

Returns: Number - 지정한 간격으로 반올림 값

| Param | Type | Description |
|----------|--------|-------------|
| val | Number | 반올림할 값 |
| snapSize | Number | 이동간격 |

Util.apply(obj, config, defaults) ⇒ Object

Copies all the properties of config to obj.

Kind: static method of [Util](#)

Returns: Object - returns obj

| Param | Type | Description |
|----------|--------|---|
| obj | Object | The receiver of the properties |
| config | Object | The source of the properties |
| defaults | Object | A different object that will also be applied for default values |

OG.geometry : object

Kind: static namespace of [OG](#)

- [.geometry](#) : object
 - [.BezierCurve](#) \Leftarrow [PolyLine](#)
 - [new OG.geometry.BezierCurve\(controlPoints\)](#)
 - [.controlPoints](#) : [Array.<Coordinate>](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getControlPoints\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.getVertices\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) \Rightarrow [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) \Rightarrow [Geometry](#)

- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.toString\(\)](#) ⇒ String
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(_geometry\)](#) ⇒ Boolean
- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean

- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
 - [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getParallelPath\(line, distance\)](#)
 - [.reset\(\)](#)
- [.Circle](#) ⇐ [Ellipse](#)
- [new OG.geometry.Circle\(center, radius\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getControlPoints\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getLength\(\)](#) ⇒ Number
 - [.toString\(\)](#) ⇒ String
 - [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
 - [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
 - [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
 - [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
 - [.isEqual\(_geometry\)](#) ⇒ Boolean
 - [.isContains\(_geometry\)](#) ⇒ Boolean
 - [.isWithin\(_geometry\)](#) ⇒ Boolean

- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

- [.Coordinate](#)

- [new OG.geometry.Coordinate\(x, y\)](#)
- [.x](#) : Number

- [.y](#) : Number
- [.distance\(coordinate\)](#) ⇒ Number
- [.move\(offsetX, offsetY\)](#) ⇒ [Coordinate](#)
- [.rotate\(angle, origin\)](#) ⇒ [Coordinate](#)
- [.isEqual\(coordinate\)](#) ⇒ Boolean
- [.toString\(\)](#) ⇒ String
- [.Curve](#) ⇐ [PolyLine](#)
 - [new OG.geometry.Curve\(controlPoints\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getControlPoints\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
 - [.toString\(\)](#) ⇒ String
 - [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
 - [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
 - [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
 - [.isEqual\(_geometry\)](#) ⇒ Boolean
 - [.isContains\(_geometry\)](#) ⇒ Boolean
 - [.isWithin\(_geometry\)](#) ⇒ Boolean

- [.getBoundary\(\)](#) ⇒ [Envelope](#)
 - [.getCentroid\(\)](#) ⇒ [Coordinate](#)
 - [.minDistance\(_coordinate\)](#) ⇒ Number
 - [.distance\(_geometry\)](#) ⇒ Number
 - [.getLength\(\)](#) ⇒ Number
 - [.moveCentroid\(중심\)](#)
 - [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
 - [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
 - [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
 - [.distanceToLine\(p, line\)](#) ⇒ Number
 - [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
 - [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
 - [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
 - [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
 - [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
 - [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
 - [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getParallelPath\(line, distance\)](#)
 - [.reset\(\)](#)
- [.Ellipse](#) ⇐ [Curve](#)
 - [new OG.geometry.Ellipse\(center, radiusX, radiusY, angle\)](#)

- [.vertices](#) : [Array.<Coordinate>](#)
- [.TYPE](#) : Number
- [.IS_CLOSED](#) : Boolean
- [.style](#) : [Style](#)
- [.boundary](#) : [Envelope](#)
- [.getControlPoints\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.toString\(\)](#) ⇒ String
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(_geometry\)](#) ⇒ Boolean
- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)

- [.distanceToLine\(p, line\)](#) ⇒ Number
 - [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
 - [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
 - [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
 - [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
 - [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
 - [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
 - [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getParallelPath\(line, distance\)](#)
 - [.reset\(\)](#)
- [.Envelope](#)
 - [new OG.geometry.Envelope\(upperLeft, width, height\)](#)
 - [.getUpperLeft\(\)](#) ⇒ [Coordinate](#)
 - [.setUpperLeft\(upperLeft\)](#)
 - [.getUpperRight\(\)](#) ⇒ [Coordinate](#)
 - [.getLowerRight\(\)](#) ⇒ [Coordinate](#)
 - [.getLowerLeft\(\)](#) ⇒ [Coordinate](#)
 - [.getLeftCenter\(\)](#) ⇒ [Coordinate](#)
 - [.getUpperCenter\(\)](#) ⇒ [Coordinate](#)
 - [.getRightCenter\(\)](#) ⇒ [Coordinate](#)
 - [.getLowerCenter\(\)](#) ⇒ [Coordinate](#)

- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.setCentroid\(centroid\)](#)
- [.getWidth\(\)](#) ⇒ Number
- [.setWidth\(width\)](#)
- [.getHeight\(\)](#) ⇒ Number
- [.setHeight\(height\)](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.isContains\(coordinate\)](#) ⇒ Boolean
- [.isContainsAll\(coordinateArray\)](#) ⇒ Boolean
- [.getHowManyContains\(coordinateArray\)](#) ⇒ Boolean
- [.isContainsOnce\(coordinateArray\)](#) ⇒ Boolean
- [.move\(offsetX, offsetY\)](#) ⇒ [Envelope](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Envelope](#)
- [.isEqual\(Envelope\)](#) ⇒ Boolean
- [.toString\(\)](#) ⇒ String

- [.Geometry](#)

- [new OG.geometry.Geometry\(\)](#)
- [.TYPE](#) : Number
- [.IS_CLOSED](#) : Boolean
- [.style](#) : [Style](#)
- [.boundary](#) : [Envelope](#)
- [.isEqual\(_geometry\)](#) ⇒ Boolean
- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean

- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.moveCentroid\(중심\)](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

- [.GeometryCollection](#) \leftarrow [Geometry](#)
 - [new OG.geometry.GeometryCollection\(geometries\)](#)
 - [.geometries](#) : [Array.<Geometry>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.toString\(\)](#) \Rightarrow String
 - [.isEqual\(_geometry\)](#) \Rightarrow Boolean
 - [.isContains\(_geometry\)](#) \Rightarrow Boolean
 - [.isWithin\(_geometry\)](#) \Rightarrow Boolean
 - [.getBoundary\(\)](#) \Rightarrow [Envelope](#)
 - [.getCentroid\(\)](#) \Rightarrow [Coordinate](#)
 - [.getVertices\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.minDistance\(_coordinate\)](#) \Rightarrow Number
 - [.distance\(_geometry\)](#) \Rightarrow Number
 - [.getLength\(\)](#) \Rightarrow Number
 - [.move\(offsetX, offsetY\)](#) \Rightarrow [Geometry](#)
 - [.moveCentroid\(중심\)](#)
 - [.resize\(upper, lower, left, right\)](#) \Rightarrow [Geometry](#)
 - [.resizeBox\(width, height\)](#) \Rightarrow [Geometry](#)
 - [.rotate\(angle, origin\)](#) \Rightarrow [Geometry](#)
 - [.fitToBoundary\(envelope\)](#) \Rightarrow [Geometry](#)
 - [.convertCoordinate\(coordinate\)](#) \Rightarrow [Coordinate](#)

- [.distanceToLine\(p, line\)](#) ⇒ Number
 - [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
 - [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
 - [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
 - [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
 - [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
 - [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
 - [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getParallelPath\(line, distance\)](#)
 - [.reset\(\)](#)
- [.Line](#) ⇐ [PolyLine](#)
 - [new OG.geometry.Line\(from, to\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)

- [.toString\(\)](#) ⇒ String
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(_geometry\)](#) ⇒ Boolean
- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)

- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)
- [.Point](#) ⇐ [Geometry](#)
 - [new OG.geometry.Point\(coordinate\)](#)
 - [.coordinate](#) : [Coordinate](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.toString\(\)](#) ⇒ String
 - [.isEqual\(_geometry\)](#) ⇒ Boolean
 - [.isContains\(_geometry\)](#) ⇒ Boolean
 - [.isWithin\(_geometry\)](#) ⇒ Boolean
 - [.getBoundary\(\)](#) ⇒ [Envelope](#)
 - [.getCentroid\(\)](#) ⇒ [Coordinate](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.minDistance\(_coordinate\)](#) ⇒ Number
 - [.distance\(_geometry\)](#) ⇒ Number
 - [.getLength\(\)](#) ⇒ Number
 - [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
 - [.moveCentroid\(중심\)](#)
 - [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)

- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
 - [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
 - [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
 - [.distanceToLine\(p, line\)](#) ⇒ Number
 - [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
 - [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
 - [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
 - [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
 - [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
 - [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
 - [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getParallelPath\(line, distance\)](#)
 - [.reset\(\)](#)
- [.PolyLine](#) ⇐ [Geometry](#)
 - [new OG.geometry.PolyLine\(vertices\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)

- [.toString\(\)](#) ⇒ String
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(_geometry\)](#) ⇒ Boolean
- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.moveCentroid\(중심\)](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)

- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
 - [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
 - [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
 - [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
 - [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getParallelPath\(line, distance\)](#)
 - [.reset\(\)](#)
- [.Polygon](#) ⇐ [PolyLine](#)
 - [new OG.geometry.Polygon\(vertices\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
 - [.toString\(\)](#) ⇒ String
 - [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
 - [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
 - [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
 - [.isEqual\(_geometry\)](#) ⇒ Boolean
 - [.isContains\(_geometry\)](#) ⇒ Boolean

- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

- [.Rectangle](#) ⇐ [Polygon](#)

- [new OG.geometry.Rectangle\(upperLeft, width, height\)](#)
- [.vertices](#) : [Array.<Coordinate>](#)
- [.TYPE](#) : Number
- [.IS_CLOSED](#) : Boolean
- [.style](#) : [Style](#)
- [.boundary](#) : [Envelope](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.toString\(\)](#) ⇒ String
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(_geometry\)](#) ⇒ Boolean
- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)

- [.distanceToLine\(p, line\)](#) ⇒ Number
 - [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
 - [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
 - [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
 - [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
 - [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
 - [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
 - [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
 - [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getParallelPath\(line, distance\)](#)
 - [.reset\(\)](#)
- [.Style](#) ⇐ [HashMap](#)
 - [new OG.geometry.Style\(style\)](#)
 - [.map](#) : Object
 - [.put\(key, value\)](#)
 - [.get\(key\)](#) ⇒ Object
 - [.containsKey\(key\)](#) ⇒ Boolean
 - [.containsValue\(value\)](#) ⇒ Boolean
 - [.isEmpty\(\)](#) ⇒ Boolean
 - [.clear\(\)](#)
 - [.remove\(key\)](#)
 - [.keys\(\)](#) ⇒ [Array.<String>](#)

- [.values\(\)](#) ⇒ Array.<Object>
- [.size\(\)](#) ⇒ Number
- [.toString\(\)](#) ⇒ String

geometry.BezierCurve ⇐ [PolyLine](#)

Kind: static class of [geometry](#)

Extends: [PolyLine](#)

Requires: module:OG.geometry.Coordinate, module:OG.geometry.Envelope,
module:OG.geometry.Geometry, module:OG.common.CurveUtil

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.BezierCurve](#) ⇐ [PolyLine](#)
 - [new OG.geometry.BezierCurve\(controlPoints\)](#)
 - [.controlPoints](#) : [Array.<Coordinate>](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getControlPoints\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)

- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.toString\(\)](#) ⇒ String
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(_geometry\)](#) ⇒ Boolean
- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object

- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

```
new OG.geometry.BezierCurve(controlPoints)
```

Cubic Bezier Curve 공간 기하 객체(Spatial Geometry Object)

컨트롤포인트1, 컨트롤포인트2에 의해 시작좌표, 끝좌표를 지나는 곡선을 나타낸다.

| Param | Type | Description |
|---------------|--|--|
| controlPoints | Array.<Coordinate> | [from, control_point1, control_point2, to] |

Example

```
var geom = new OG.geometry.BezierCurve([[200, 100], [100, 300], [-100, -100], [-200, 100]]);
```

bezierCurve.controlPoints : [Array.<Coordinate>](#)

Bezier Curve 컨트롤 좌표 Array

Kind: instance property of [BezierCurve](#)

bezierCurve.vertices : [Array.<Coordinate>](#)

Line Vertex 좌표 Array

Kind: instance property of [BezierCurve](#)

Overrides: [vertices](#)

bezierCurve.TYPE : Number

공간 기하 객체 타입

Kind: instance property of [BezierCurve](#)

Overrides: [TYPE](#)

bezierCurve.IS_CLOSED : Boolean

닫힌 기하 객체 인지 여부

Kind: instance property of [BezierCurve](#)

bezierCurve.style : [Style](#)

스타일 속성

Kind: instance property of [BezierCurve](#)

Overrides: [style](#)

bezierCurve.boundary : [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역

Kind: instance property of [BezierCurve](#)

Overrides: [boundary](#)

`bezierCurve.getControlPoints()` ⇒ [Array.<Coordinate>](#)

콘트롤 포인트 목록을 반환한다.

Kind: instance method of [BezierCurve](#)

Returns: [Array.<Coordinate>](#) - controlPoints Array

`bezierCurve.getVertices()` ⇒ [Array.<Coordinate>](#)

공간기하객체의 모든 꼭지점을 반환한다.

Kind: instance method of [BezierCurve](#)

Overrides: [getVertices](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array

`bezierCurve.move(offsetX, offsetY)` ⇒ [Geometry](#)

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance method of [BezierCurve](#)

Overrides: [move](#)

Returns: [Geometry](#) - 이동된 공간 기하 객체

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

`bezierCurve.resize(upper, lower, left, right)` ⇒ [Geometry](#)

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance method of [BezierCurve](#)

Overrides: [resize](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|--------|----------------------------|
| upper | Number | 상단 라인 이동 Offset(위 방향으로 +) |
| lower | Number | 하단 라인 이동 Offset(아래 방향으로 +) |
| left | Number | 좌측 라인 이동 Offset(좌측 방향으로 +) |
| right | Number | 우측 라인 이동 Offset(우측 방향으로 +) |

`bezierCurve.rotate(angle, origin)` ⇒ [Geometry](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [BezierCurve](#)

Overrides: [rotate](#)

Returns: [Geometry](#) - 회전된 공간 기하 객체

| Param | Type | Description |
|--------|----------------------------|---------------------|
| angle | Number | 회전 각도 |
| origin | Coordinate | 기준 좌표(default:중심좌표) |

bezierCurve.toString() ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [BezierCurve](#)

Overrides: [toString](#)

Returns: String - 프라퍼티 정보

bezierCurve.angleBetweenPoints(prev, next) ⇒ Number

공간기하객체의 두 꼭지점 사이에 가상의 선을 그렸을때, 그 기울기를 구한다.

Kind: instance method of [BezierCurve](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

bezierCurve.isRightAngleBetweenPoints(prev, next) ⇒ Object

공간기하객체의 두 꼭지점 사이의 기울기가 수평또는 수직인지 판별한다.

Kind: instance method of [BezierCurve](#)

Returns: Object - {flag : true or false, type: horizontal or vertical or none}

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

`bezierCurve.angleBetweenThreePoints(prev, next) ⇒ Number`

공간기하객체의 세 꼭지점 사이의 각도 중 작은 각도를 반환한다.

Kind: instance method of [BezierCurve](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------------------|----------------------------|-------------|
| <code>prev</code> | Coordinate | 꼭지점 1 |
| <code>next</code> | Coordinate | 꼭지점 2 |

`bezierCurve.isEquals(_geometry) ⇒ Boolean`

주어진 Geometry 객체와 같은지 비교한다.

Kind: instance method of [BezierCurve](#)

Returns: Boolean - true:같은, false:다름

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`bezierCurve.isContains(_geometry) ⇒ Boolean`

주어진 공간기하객체를 포함하는지 비교한다.

Kind: instance method of [BezierCurve](#)

Returns: Boolean - 포함하면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`bezierCurve.isWithin(_geometry)` ⇒ Boolean

주어진 공간기하객체에 포함되는지 비교한다.

Kind: instance method of [BezierCurve](#)

Returns: Boolean - 포함되면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`bezierCurve.getBoundary()` ⇒ [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역을 반환한다.

Kind: instance method of [BezierCurve](#)

Returns: [Envelope](#) - Envelope 영역

`bezierCurve.getCentroid()` ⇒ [Coordinate](#)

공간기하객체의 중심좌표를 반환한다.

Kind: instance method of [BezierCurve](#)

Returns: [Coordinate](#) - 중심좌표

`bezierCurve.minDistance(_coordinate)` ⇒ Number

주어진 좌표와의 최단거리를 반환한다.

Kind: instance method of [BezierCurve](#)

Returns: Number - 최단거리

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

bezierCurve.distance(_geometry) ⇒ Number

주어진 공간기하객체와의 중심점 간의 거리를 반환한다.

Kind: instance method of [BezierCurve](#)

Returns: Number - 거리

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | 공간 기하 객체 |

bezierCurve.getLength() ⇒ Number

공간기하객체의 길이를 반환한다.

Kind: instance method of [BezierCurve](#)

Returns: Number - 길이

bezierCurve.moveCentroid(중심)

주어진 중심좌표로 공간기하객체를 이동한다.

Kind: instance method of [BezierCurve](#)

| Param | Type | Description |
|-------|----------------------------|-------------|
| 중심 | Coordinate | 좌표 |

`bezierCurve.resizeBox(width, height)` ⇒ [Geometry](#)

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [BezierCurve](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|--------|--------|-------------|
| width | Number | 너비 |
| height | Number | 높이 |

`bezierCurve.fitToBoundary(envelope)` ⇒ [Geometry](#)

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

Kind: instance method of [BezierCurve](#)

Returns: [Geometry](#) - 적용된 공간 기하 객체

| Param | Type | Description |
|----------|--------------------------|-------------|
| envelope | Envelope | Envelope 영역 |

`bezierCurve.convertCoordinate(coordinate)` ⇒ [Coordinate](#)

파라미터가 [x, y] 형식의 좌표 Array 이면 OG.geometry.Coordinate 인스턴스를 new 하여 반환한다.

Kind: instance method of [BezierCurve](#)

| Param | Type | Description |
|------------|---|--|
| coordinate | Coordinate Array.<Number> | [x, y] 형식의 좌표 Array 또는 OG.geometry.Coordinate 인스턴스 |

bezierCurve.distanceToLine(p, line) ⇒ Number

포인트 P 로부터 라인 AB의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [BezierCurve](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

bezierCurve.distanceLineToLine(line1, line2) ⇒ Number

라인1 로부터 라인2 의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [BezierCurve](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |

bezierCurve.intersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표들을 반환한다.

Kind: instance method of [BezierCurve](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

bezierCurve.shortestIntersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표중 시작좌표에 가장 가까운 좌표를 반환한다.

Kind: instance method of [BezierCurve](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

bezierCurve.intersectLineToLine(line1, line2, extension) ⇒ [Coordinate](#)

라인1 과 라인2 의 교차점을 계산한다.

Kind: instance method of [BezierCurve](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-----------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |
| extension | boolean | 라인을 연장하여 교차점을 계산하는 여부 |

`bezierCurve.intersectCircleToLine(center, radius, from, to) ⇒ Array.<Coordinate>`

주어진 원과 주어진 라인의 교차점을 계산한다.

Kind: instance method of [BezierCurve](#)

Returns: [Array.<Coordinate>](#) - 교차점

| Param | Type | Description |
|--------|----------------------------|--------------|
| center | Coordinate | 중심점 |
| radius | Number | 반경 |
| from | Coordinate | line 라인 시작좌표 |
| to | Coordinate | line 라인 끝좌표 |

`bezierCurve.intersectPointToLine(p, line) ⇒ Coordinate`

포인트 P 로부터 라인 AB 까지 수직인 가상선을 생각할때, 그 교차점을 계산한다.

Note: NON-ROBUST!

Kind: instance method of [BezierCurve](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-------|---|------------------------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

`bezierCurve.getPercentageDistanceFromPoint(_coordinate) ⇒ Object`

주어진 좌표에 대해 공간기하객체 바운더리 대비 가로,세로 위치 퍼센테이지 비율을 구한다.

Kind: instance method of [BezierCurve](#)

Returns: Object - {px , py}

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

bezierCurve.isContainsPoint(_coordinate) ⇒ boolean

공간기하객체가 주어진 좌표를 포함하는지를 반환한다.

Kind: instance method of [BezierCurve](#)

Returns: boolean - true,false

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

bezierCurve.getPointFromPercentageDistance(pXpY) ⇒ [Coordinate](#)

공간기하객체에 대한 퍼센테이지 좌표의 실제 좌표를 구한다.

Kind: instance method of [BezierCurve](#)

Returns: [Coordinate](#) - 실 좌표

| Param | Type | Description |
|-------|-------|-------------|
| pXpY | Array | 퍼센테이지 좌표 |

bezierCurve.getParallelLine(from, to, distance) ⇒ [Array.<Coordinate>](#)

주어진 선분과 일정 거리에 있는 평행한 선분을 반환한다.

Kind: instance method of [BezierCurve](#)

Returns: [Array.<Coordinate>](#) - 평행선 시작좌표, 끝좌표 Array

| Param | Type | Description |
|----------|----------------------------|-------------|
| from | Coordinate | 라인 시작좌표 |
| to | Coordinate | 라인 끝좌표 |
| distance | | |

`bezierCurve.getParallelPath(line, distance)`

주어진 라인과 일정 거리에 있는 평행한 라인을 반환한다.

Kind: instance method of [BezierCurve](#)

| Param | Type | Description |
|----------|--|-------------|
| line | Array.<Coordinate> | 라인 좌표 Array |
| distance | | |

`bezierCurve.reset()`

저장된 boundary 를 클리어하여 새로 계산하도록 한다.

Kind: instance method of [BezierCurve](#)

Overrides: [reset](#)

`geometry.Circle` \Leftarrow [Ellipse](#)

Kind: static class of [geometry](#)

Extends: [Ellipse](#)

Requires: module:OG.geometry.Coordinate, module:OG.geometry.Envelope,
module:OG.geometry.Geometry

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Circle](#) \Leftarrow [Ellipse](#)
 - [new OG.geometry.Circle\(center, radius\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getVertices\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.getControlPoints\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.getLength\(\)](#) \Rightarrow Number
 - [.toString\(\)](#) \Rightarrow String
 - [.move\(offsetX, offsetY\)](#) \Rightarrow [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) \Rightarrow [Geometry](#)
 - [.rotate\(angle, origin\)](#) \Rightarrow [Geometry](#)
 - [.angleBetweenPoints\(prev, next\)](#) \Rightarrow Number
 - [.isRightAngleBetweenPoints\(prev, next\)](#) \Rightarrow Object
 - [.angleBetweenThreePoints\(prev, next\)](#) \Rightarrow Number
 - [.isEquals\(_geometry\)](#) \Rightarrow Boolean
 - [.isContains\(_geometry\)](#) \Rightarrow Boolean
 - [.isWithin\(_geometry\)](#) \Rightarrow Boolean
 - [.getBoundary\(\)](#) \Rightarrow [Envelope](#)

- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

new OG.geometry.Circle(center, radius)

Circle 공간 기하 객체(Spatial Geometry Object)

| Param | Type | Description |
|--------|----------------------------|--------------|
| center | Coordinate | Circle 중심 좌표 |
| radius | Number | radius 반경 |

Example

```
var geom = new OG.geometry.Circle([10, 10], 5);
```

circle.vertices : [Array.<Coordinate>](#)

Line Vertex 좌표 Array

Kind: instance property of [Circle](#)

circle.TYPE : Number

공간 기하 객체 타입

Kind: instance property of [Circle](#)

Overrides: [TYPE](#)

circle.IS_CLOSED : Boolean

닫힌 기하 객체 인지 여부

Kind: instance property of [Circle](#)

circle.style : [Style](#)

스타일 속성

Kind: instance property of [Circle](#)

Overrides: [style](#)

circle.boundary : [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역

Kind: instance property of [Circle](#)

circle.getVertices() ⇒ [Array.<Coordinate>](#)

공간기하객체의 모든 꼭지점을 반환한다.

Kind: instance method of [Circle](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array

circle.getControlPoints() ⇒ [Array.<Coordinate>](#)

컨트롤 포인트 목록을 반환한다.

Kind: instance method of [Circle](#)

Returns: [Array.<Coordinate>](#) - controlPoints Array

circle.getLength() ⇒ Number

공간기하객체의 길이를 반환한다.

Kind: instance method of [Circle](#)

Overrides: [getLength](#)

Returns: Number - 길이

circle.toString() ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [Circle](#)

Overrides: [toString](#)

Returns: String - 프라퍼티 정보

circle.move(offsetX, offsetY) ⇒ [Geometry](#)

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance method of [Circle](#)

Returns: [Geometry](#) - 이동된 공간 기하 객체

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

circle.resize(upper, lower, left, right) ⇒ [Geometry](#)

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance method of [Circle](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|--------|----------------------------|
| upper | Number | 상단 라인 이동 Offset(위 방향으로 +) |
| lower | Number | 하단 라인 이동 Offset(아래 방향으로 +) |
| left | Number | 좌측 라인 이동 Offset(좌측 방향으로 +) |
| right | Number | 우측 라인 이동 Offset(우측 방향으로 +) |

circle.rotate(angle, origin) ⇒ [Geometry](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [Circle](#)

Returns: [Geometry](#) - 회전된 공간 기하 객체

| Param | Type | Description |
|--------|----------------------------|----------------------|
| angle | Number | 회전 각도 |
| origin | Coordinate | 기준 좌표(default: 중심좌표) |

circle.angleBetweenPoints(prev, next) ⇒ Number

공간기하객체의 두 꼭지점 사이에 가상의 선을 그렸을때, 그 기울기를 구한다.

Kind: instance method of [Circle](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|----------------------------|-------------|
| prev | Coordinate | 꼭지점 1 |

next [Coordinate](#) 꼭지점 2

circle.isRightAngleBetweenPoints(prev, next) ⇒ Object

공간기하객체의 두 꼭지점 사이의 기울기가 수평또는 수직인지 판별한다.

Kind: instance method of [Circle](#)

Returns: Object - {flag : true or false, type: horizontal or vertical or none}

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

circle.angleBetweenThreePoints(prev, next) ⇒ Number

공간기하객체의 세 꼭지점 사이의 각도 중 작은 각도를 반환한다.

Kind: instance method of [Circle](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

circle.isEquals(_geometry) ⇒ Boolean

주어진 Geometry 객체와 같은지 비교한다.

Kind: instance method of [Circle](#)

Returns: Boolean - true:같음, false:다름

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`circle.isContains(_geometry) ⇒ Boolean`

주어진 공간기하객체를 포함하는지 비교한다.

Kind: instance method of [Circle](#)

Returns: Boolean - 포함하면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`circle.isWithin(_geometry) ⇒ Boolean`

주어진 공간기하객체에 포함되는지 비교한다.

Kind: instance method of [Circle](#)

Returns: Boolean - 포함되면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`circle.getBoundary() ⇒ Envelope`

공간기하객체를 포함하는 사각형의 Boundary 영역을 반환한다.

Kind: instance method of [Circle](#)

Returns: [Envelope](#) - Envelope 영역

circle.getCentroid() ⇒ [Coordinate](#)

공간기하객체의 중심좌표를 반환한다.

Kind: instance method of [Circle](#)

Returns: [Coordinate](#) - 중심좌표

circle.minDistance(_coordinate) ⇒ Number

주어진 좌표와의 최단거리를 반환한다.

Kind: instance method of [Circle](#)

Returns: Number - 최단거리

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

circle.distance(_geometry) ⇒ Number

주어진 공간기하객체와의 중심점 간의 거리를 반환한다.

Kind: instance method of [Circle](#)

Returns: Number - 거리

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | 공간 기하 객체 |

circle.moveCentroid([중심](#))

주어진 중심좌표로 공간기하객체를 이동한다.

Kind: instance method of [Circle](#)

| Param | Type | Description |
|-------|----------------------------|-------------|
| 중심 | Coordinate | 좌표 |

circle.resizeBox(width, height) ⇒ [Geometry](#)

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [Circle](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|--------|--------|-------------|
| width | Number | 너비 |
| height | Number | 높이 |

circle.fitToBoundary(envelope) ⇒ [Geometry](#)

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

Kind: instance method of [Circle](#)

Returns: [Geometry](#) - 적용된 공간 기하 객체

| Param | Type | Description |
|----------|--------------------------|-------------|
| envelope | Envelope | Envelope 영역 |

circle.convertCoordinate(coordinate) ⇒ [Coordinate](#)

파라미터가 [x, y] 형식의 좌표 Array 이면 OG.geometry.Coordinate 인스턴스를 new 하여 반환한다.

Kind: instance method of [Circle](#)

| Param | Type | Description |
|------------|---|--|
| coordinate | Coordinate Array.<Number> | [x, y] 형식의 좌표 Array 또는 OG.geometry.Coordinate 인스턴스 |

circle.distanceToLine(p, line) ⇒ Number

포인트 P 로부터 라인 AB의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Circle](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

circle.distanceLineToLine(line1, line2) ⇒ Number

라인1 로부터 라인2 의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Circle](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |

circle.intersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표들을 반환한다.

Kind: instance method of [Circle](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

circle.shortestIntersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표중 시작좌표에 가장 가까운 좌표를 반환한다.

Kind: instance method of [Circle](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

circle.intersectLineToLine(line1, line2, extension) ⇒ [Coordinate](#)

라인1 과 라인2 의 교차점을 계산한다.

Kind: instance method of [Circle](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-----------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |
| extension | boolean | 라인을 연장하여 교차점을 계산하는 여부 |

circle.intersectCircleToLine(center, radius, from, to) ⇒ [Array.<Coordinate>](#)

주어진 원과 주어진 라인의 교차점을 계산한다.

Kind: instance method of [Circle](#)

Returns: [Array.<Coordinate>](#) - 교차점

| Param | Type | Description |
|--------|----------------------------|--------------|
| center | Coordinate | 중심점 |
| radius | Number | 반경 |
| from | Coordinate | line 라인 시작좌표 |
| to | Coordinate | line 라인 끝좌표 |

circle.intersectPointToLine(p, line) ⇒ [Coordinate](#)

포인트 P로부터 라인 AB까지 수직인 가상선을 생각할때, 그 교차점을 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Circle](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-------|--|---------------------|
| p | Coordinate | Array.<Number> 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

circle.getPercentageDistanceFromPoint(_coordinate) ⇒ Object

주어진 좌표에 대해 공간기하객체 바운더리 대비 가로,세로 위치 퍼센테이지 비율을 구한다.

Kind: instance method of [Circle](#)

Returns: Object - {px , py}

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

circle.isContainsPoint(_coordinate) ⇒ boolean

공간기하객체가 주어진 좌표를 포함하는지를 반환한다.

Kind: instance method of [Circle](#)

Returns: boolean - true,false

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

circle.getPointFromPercentageDistance(pXpY) ⇒ [Coordinate](#)

공간기하객체에 대한 퍼센테이지 좌표의 실제 좌표를 구한다.

Kind: instance method of [Circle](#)

Returns: [Coordinate](#) - 실 좌표

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

pXpY Array 퍼센테이지 좌표

circle.getParallelLine(from, to, distance) ⇒ [Array.<Coordinate>](#)

주어진 선분과 일정 거리에 있는 평행한 선분을 반환한다.

Kind: instance method of [Circle](#)

Returns: [Array.<Coordinate>](#) - 평행선 시작좌표, 끝좌표 Array

| Param | Type | Description |
|----------|----------------------------|-------------|
| from | Coordinate | 라인 시작좌표 |
| to | Coordinate | 라인 끝좌표 |
| distance | | |

circle.getParallelPath(line, distance)

주어진 라인과 일정 거리에 있는 평행한 라인을 반환한다.

Kind: instance method of [Circle](#)

| Param | Type | Description |
|----------|--|-------------|
| line | Array.<Coordinate> | 라인 좌표 Array |
| distance | | |

circle.reset()

저장된 boundary 를 클리어하여 새로 계산하도록 한다.

Kind: instance method of [Circle](#)

geometry.Coordinate

Kind: static class of [geometry](#)

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Coordinate](#)
 - [new OG.geometry.Coordinate\(x, y\)](#)
 - [.x](#) : Number
 - [.y](#) : Number
 - [.distance\(coordinate\)](#) ⇒ Number
 - [.move\(offsetX, offsetY\)](#) ⇒ [Coordinate](#)
 - [.rotate\(angle, origin\)](#) ⇒ [Coordinate](#)
 - [.isEqual\(coordinate\)](#) ⇒ Boolean
 - [.toString\(\)](#) ⇒ String

`new OG.geometry.Coordinate(x, y)`

2차원 좌표계에서의 좌표값

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|---|--------|-----|
| x | Number | x좌표 |
|---|--------|-----|

| | | |
|---|--------|-----|
| y | Number | y좌표 |
|---|--------|-----|

Example

```
var coordinate1 = new OG.geometry.Coordinate(10, 10);  
or  
var coordinate2 = new OG.geometry.Coordinate([20, 20]);
```

coordinate.x : Number

x좌표

Kind: instance property of [Coordinate](#)

coordinate.y : Number

y좌표

Kind: instance property of [Coordinate](#)

coordinate.distance(coordinate) ⇒ Number

주어진 좌표와의 거리를 계산한다.

Kind: instance method of [Coordinate](#)

Returns: Number - 좌표간의 거리값

| Param | Type | Description |
|------------|----------------------------|--------------------|
| coordinate | Coordinate | Array.<Number> 좌표값 |

Example

```
coordinate.distance([10, 10]);  
or  
coordinate.distance(new OG.Coordinate(10, 10));
```

coordinate.move(offsetX, offsetY) ⇒ [Coordinate](#)

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance method of [Coordinate](#)

Returns: [Coordinate](#) - 이동된 좌표

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

coordinate.rotate(angle, origin) ⇒ [Coordinate](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [Coordinate](#)

Returns: [Coordinate](#) - 회전된 좌표

| Param | Type | Description |
|--------|---|-------------|
| angle | Number | 회전 각도 |
| origin | Coordinate Array.<Number> | 기준 좌표 |

Example

```
coordinate.rotate(90, [10,10]);  
or  
coordinate.rotate(90, new OG.Coordinate(10, 10));
```

`coordinate.isEquals(coordinate) ⇒ Boolean`

주어진 좌표값과 같은지 비교한다.

Kind: instance method of [Coordinate](#)

Returns: Boolean - true:같은, false:다름

| Param | Type | Description |
|------------|----------------------------|--------------------|
| coordinate | Coordinate | Array.<Number> 좌표값 |

Example

```
coordinate.isEquals([10, 10]);  
or  
coordinate.isEquals(new OG.Coordinate(10, 10));
```

`coordinate.toString() ⇒ String`

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [Coordinate](#)

Returns: String - 프라퍼티 정보

`geometry.Curve ⇐ PolyLine`

Kind: static class of [geometry](#)

Extends: [PolyLine](#)

Requires: module:OG.geometry.Coordinate, module:OG.geometry.Envelope,

module:OG.geometry.Geometry, module:OG.common.CurveUtil

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Curve](#) \Leftarrow [PolyLine](#)
 - [new OG.geometry.Curve\(controlPoints\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getControlPoints\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.getVertices\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) \Rightarrow [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) \Rightarrow [Geometry](#)
 - [.rotate\(angle, origin\)](#) \Rightarrow [Geometry](#)
 - [.toString\(\)](#) \Rightarrow String
 - [.angleBetweenPoints\(prev, next\)](#) \Rightarrow Number
 - [.isRightAngleBetweenPoints\(prev, next\)](#) \Rightarrow Object
 - [.angleBetweenThreePoints\(prev, next\)](#) \Rightarrow Number
 - [.isEqual\(_geometry\)](#) \Rightarrow Boolean
 - [.isContains\(_geometry\)](#) \Rightarrow Boolean
 - [.isWithin\(_geometry\)](#) \Rightarrow Boolean
 - [.getBoundary\(\)](#) \Rightarrow [Envelope](#)
 - [.getCentroid\(\)](#) \Rightarrow [Coordinate](#)
 - [.minDistance\(_coordinate\)](#) \Rightarrow Number

- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

new OG.geometry.Curve(controlPoints)

Catmull-Rom Spline Curve 공간 기하 객체(Spatial Geometry Object)

모든 콘트롤포인트를 지나는 곡선을 나타낸다.

| Param | Type | Description |
|---------------|--|-----------------------|
| controlPoints | Array.<Coordinate> | Curve Vertex 좌표 Array |

Example

```
var geom = new OG.geometry.Curve([[200, 100], [100, 300], [-100, -100], [-200, 100]]);
```

curve.vertices : [Array.<Coordinate>](#)

Line Vertex 좌표 Array

Kind: instance property of [Curve](#)

Overrides: [vertices](#)

curve.TYPE : Number

공간 기하 객체 타입

Kind: instance property of [Curve](#)

Overrides: [TYPE](#)

curve.IS_CLOSED : Boolean

닫힌 기하 객체 인지 여부

Kind: instance property of [Curve](#)

curve.style : [Style](#)

스타일 속성

Kind: instance property of [Curve](#)

Overrides: [style](#)

curve.boundary : [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역

Kind: instance property of [Curve](#)

curve.getControlPoints() ⇒ [Array.<Coordinate>](#)

컨트롤 포인트 목록을 반환한다.

Kind: instance method of [Curve](#)

Returns: [Array.<Coordinate>](#) - controlPoints Array

curve.getVertices() ⇒ [Array.<Coordinate>](#)

공간기하객체의 모든 꼭지점을 반환한다.

Kind: instance method of [Curve](#)

Overrides: [getVertices](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array

`curve.move(offsetX, offsetY)` ⇒ [Geometry](#)

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance method of [Curve](#)

Returns: [Geometry](#) - 이동된 공간 기하 객체

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

`curve.resize(upper, lower, left, right)` ⇒ [Geometry](#)

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance method of [Curve](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|--------|----------------------------|
| upper | Number | 상단 라인 이동 Offset(위 방향으로 +) |
| lower | Number | 하단 라인 이동 Offset(아래 방향으로 +) |
| left | Number | 좌측 라인 이동 Offset(좌측 방향으로 +) |
| right | Number | 우측 라인 이동 Offset(우측 방향으로 +) |

`curve.rotate(angle, origin)` ⇒ [Geometry](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [Curve](#)

Returns: [Geometry](#) - 회전된 공간 기하 객체

| Param | Type | Description |
|--------|----------------------------|----------------------|
| angle | Number | 회전 각도 |
| origin | Coordinate | 기준 좌표(default: 중심좌표) |

curve.toString() ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [Curve](#)

Overrides: [toString](#)

Returns: String - 프라퍼티 정보

curve.angleBetweenPoints(prev, next) ⇒ Number

공간기하객체의 두 꼭지점 사이에 가상의 선을 그렸을때, 그 기울기를 구한다.

Kind: instance method of [Curve](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|----------------------------|-------------|
| prev | Coordinate | 꼭지점 1 |
| next | Coordinate | 꼭지점 2 |

curve.isRightAngleBetweenPoints(prev, next) ⇒ Object

공간기하객체의 두 꼭지점 사이의 기울기가 수평또는 수직인지 판별한다.

Kind: instance method of [Curve](#)

Returns: Object - {flag : true or false, type: horizontal or vertical or none}

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

curve.angleBetweenThreePoints(prev, next) ⇒ Number

공간기하객체의 세 꼭지점 사이의 각도 중 작은 각도를 반환한다.

Kind: instance method of [Curve](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

curve.isEquals(_geometry) ⇒ Boolean

주어진 Geometry 객체와 같은지 비교한다.

Kind: instance method of [Curve](#)

Returns: Boolean - true:같음, false:다름

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |
|-----------|--------------------------|-------------|

curve.isContains(_geometry) ⇒ Boolean

주어진 공간기하객체를 포함하는지 비교한다.

Kind: instance method of [Curve](#)

Returns: Boolean - 포함하면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`curve.isWithin(_geometry)` ⇒ Boolean

주어진 공간기하객체에 포함되는지 비교한다.

Kind: instance method of [Curve](#)

Returns: Boolean - 포함되면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`curve.getBoundary()` ⇒ [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역을 반환한다.

Kind: instance method of [Curve](#)

Returns: [Envelope](#) - Envelope 영역

`curve.getCentroid()` ⇒ [Coordinate](#)

공간기하객체의 중심좌표를 반환한다.

Kind: instance method of [Curve](#)

Returns: [Coordinate](#) - 중심좌표

`curve.minDistance(_coordinate) ⇒ Number`

주어진 좌표와의 최단거리를 반환한다.

Kind: instance method of [Curve](#)

Returns: Number - 최단거리

| Param | Type | Description |
|--------------------------|----------------------------|-------------|
| <code>_coordinate</code> | Coordinate | 좌표 |

`curve.distance(_geometry) ⇒ Number`

주어진 공간기하객체와의 중심점 간의 거리를 반환한다.

Kind: instance method of [Curve](#)

Returns: Number - 거리

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | 공간 기하 객체 |

`curve.getLength() ⇒ Number`

공간기하객체의 길이를 반환한다.

Kind: instance method of [Curve](#)

Returns: Number - 길이

`curve.moveCentroid(중심)`

주어진 중심좌표로 공간기하객체를 이동한다.

Kind: instance method of [Curve](#)

| Param | Type | Description |
|-------|----------------------------|-------------|
| 중심 | Coordinate | 좌표 |

`curve.resizeBox(width, height) ⇒ Geometry`

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [Curve](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|--------|--------|-------------|
| width | Number | 너비 |
| height | Number | 높이 |

`curve.fitToBoundary(envelope) ⇒ Geometry`

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

Kind: instance method of [Curve](#)

Returns: [Geometry](#) - 적용된 공간 기하 객체

| Param | Type | Description |
|----------|--------------------------|-------------|
| envelope | Envelope | Envelope 영역 |

curve.convertCoordinate(coordinate) ⇒ [Coordinate](#)

파라미터가 [x, y] 형식의 좌표 Array 이면 OG.geometry.Coordinate 인스턴스를 new 하여 반환한다.

Kind: instance method of [Curve](#)

| Param | Type | Description |
|------------|---|--|
| coordinate | Coordinate Array.<Number> | [x, y] 형식의 좌표 Array 또는 OG.geometry.Coordinate 인스턴스 |

curve.distanceToLine(p, line) ⇒ Number

포인트 P 로부터 라인 AB의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Curve](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

curve.distanceLineToLine(line1, line2) ⇒ Number

라인1 로부터 라인2 의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Curve](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |

curve.intersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표들을 반환한다.

Kind: instance method of [Curve](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

curve.shortestIntersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표중 시작좌표에 가장 가까운 좌표를 반환한다.

Kind: instance method of [Curve](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

curve.intersectLineToLine(line1, line2, extension) ⇒ [Coordinate](#)

라인1 과 라인2 의 교차점을 계산한다.

Kind: instance method of [Curve](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-----------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |
| extension | boolean | 라인을 연장하여 교차점을 계산하는 여부 |

curve.intersectCircleToLine(center, radius, from, to) ⇒ [Array.<Coordinate>](#)

주어진 원과 주어진 라인의 교차점을 계산한다.

Kind: instance method of [Curve](#)

Returns: [Array.<Coordinate>](#) - 교차점

| Param | Type | Description |
|--------|----------------------------|--------------|
| center | Coordinate | 중심점 |
| radius | Number | 반경 |
| from | Coordinate | line 라인 시작좌표 |
| to | Coordinate | line 라인 끝좌표 |

curve.intersectPointToLine(p, line) ⇒ [Coordinate](#)

포인트 P로부터 라인 AB까지 수직인 가상선을 생각할때, 그 교차점을 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Curve](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-------|--|---------------------|
| p | Coordinate | Array.<Number> 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

curve.getPercentageDistanceFromPoint(_coordinate) ⇒ Object

주어진 좌표에 대해 공간기하객체 바운더리 대비 가로,세로 위치 퍼센테이지 비율을 구한다.

Kind: instance method of [Curve](#)

Returns: Object - {px , py}

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

curve.isContainsPoint(_coordinate) ⇒ boolean

공간기하객체가 주어진 좌표를 포함하는지를 반환한다.

Kind: instance method of [Curve](#)

Returns: boolean - true,false

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

curve.getPointFromPercentageDistance(pXpY) ⇒ [Coordinate](#)

공간기하객체에 대한 퍼센테이지 좌표의 실제 좌표를 구한다.

Kind: instance method of [Curve](#)

Returns: [Coordinate](#) - 실 좌표

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

pXpY Array 퍼센테이지 좌표

curve.getParallelLine(from, to, distance) ⇒ [Array.<Coordinate>](#)

주어진 선분과 일정 거리에 있는 평행한 선분을 반환한다.

Kind: instance method of [Curve](#)

Returns: [Array.<Coordinate>](#) - 평행선 시작좌표, 끝좌표 Array

| Param | Type | Description |
|----------|----------------------------|-------------|
| from | Coordinate | 라인 시작좌표 |
| to | Coordinate | 라인 끝좌표 |
| distance | | |

curve.getParallelPath(line, distance)

주어진 라인과 일정 거리에 있는 평행한 라인을 반환한다.

Kind: instance method of [Curve](#)

| Param | Type | Description |
|----------|--|-------------|
| line | Array.<Coordinate> | 라인 좌표 Array |
| distance | | |

curve.reset()

저장된 boundary 를 클리어하여 새로 계산하도록 한다.

Kind: instance method of [Curve](#)

geometry.Ellipse \Leftarrow [Curve](#)

Kind: static class of [geometry](#)

Extends: [Curve](#)

Requires: module:OG.geometry.Coordinate, module:OG.geometry.Envelope,
module:OG.geometry.Geometry

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Ellipse](#) \Leftarrow [Curve](#)
 - [new OG.geometry.Ellipse\(center, radiusX, radiusY, angle\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getControlPoints\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.getVertices\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.toString\(\)](#) \Rightarrow String
 - [.move\(offsetX, offsetY\)](#) \Rightarrow [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) \Rightarrow [Geometry](#)
 - [.rotate\(angle, origin\)](#) \Rightarrow [Geometry](#)
 - [.angleBetweenPoints\(prev, next\)](#) \Rightarrow Number
 - [.isRightAngleBetweenPoints\(prev, next\)](#) \Rightarrow Object
 - [.angleBetweenThreePoints\(prev, next\)](#) \Rightarrow Number
 - [.isEquals\(_geometry\)](#) \Rightarrow Boolean

- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

```
new OG.geometry.Ellipse(center, radiusX, radiusY, angle)
```

Ellipse 공간 기하 객체(Spatial Geometry Object)

| Param | Type | Description |
|---------|----------------------------|---------------|
| center | Coordinate | Ellipse 중심 좌표 |
| radiusX | Number | X축 반경 |
| radiusY | Number | Y축 반경 |
| angle | Number | X축 기울기 |

Example

```
var geom = new OG.geometry.Ellipse([10, 10], 10, 5);
```

ellipse.vertices : [Array.<Coordinate>](#)

Line Vertex 좌표 Array

Kind: instance property of [Ellipse](#)

ellipse.TYPE : Number

공간 기하 객체 타입

Kind: instance property of [Ellipse](#)

Overrides: [TYPE](#)

ellipse.IS_CLOSED : Boolean

닫힌 기하 객체 인지 여부

Kind: instance property of [Ellipse](#)

Overrides: [IS_CLOSED](#)

ellipse.style : [Style](#)

스타일 속성

Kind: instance property of [Ellipse](#)

Overrides: [style](#)

ellipse.boundary : [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역

Kind: instance property of [Ellipse](#)

ellipse.getControlPoints() ⇒ [Array.<Coordinate>](#)

컨트롤 포인트 목록을 반환한다.

Kind: instance method of [Ellipse](#)

Overrides: [getControlPoints](#)

Returns: [Array.<Coordinate>](#) - controlPoints Array

ellipse.getVertices() ⇒ [Array.<Coordinate>](#)

공간기하객체의 모든 꼭지점을 반환한다.

Kind: instance method of [Ellipse](#)

Overrides: [getVertices](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array

ellipse.toString() ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [Ellipse](#)

Overrides: [toString](#)

Returns: String - 프라퍼티 정보

ellipse.move(offsetX, offsetY) ⇒ [Geometry](#)

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance method of [Ellipse](#)

Returns: [Geometry](#) - 이동된 공간 기하 객체

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

`ellipse.resize(upper, lower, left, right)` ⇒ [Geometry](#)

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance method of [Ellipse](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|--------|----------------------------|
| upper | Number | 상단 라인 이동 Offset(위 방향으로 +) |
| lower | Number | 하단 라인 이동 Offset(아래 방향으로 +) |
| left | Number | 좌측 라인 이동 Offset(좌측 방향으로 +) |
| right | Number | 우측 라인 이동 Offset(우측 방향으로 +) |

`ellipse.rotate(angle, origin)` ⇒ [Geometry](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [Ellipse](#)

Returns: [Geometry](#) - 회전된 공간 기하 객체

| Param | Type | Description |
|--------|----------------------------|---------------------|
| angle | Number | 회전 각도 |
| origin | Coordinate | 기준 좌표(default:중심좌표) |

`ellipse.angleBetweenPoints(prev, next)` ⇒ Number

공간기하객체의 두 꼭지점 사이에 가상의 선을 그렸을때, 그 기울기를 구한다.

Kind: instance method of [Ellipse](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

ellipse.isRightAngleBetweenPoints(prev, next) ⇒ Object

공간기하객체의 두 꼭지점 사이의 기울기가 수평또는 수직인지 판별한다.

Kind: instance method of [Ellipse](#)

Returns: Object - {flag : true or false, type: horizontal or vertical or none}

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

ellipse.angleBetweenThreePoints(prev, next) ⇒ Number

공간기하객체의 세 꼭지점 사이의 각도 중 작은 각도를 반환한다.

Kind: instance method of [Ellipse](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

ellipse.isEquals(_geometry) ⇒ Boolean

주어진 Geometry 객체와 같은지 비교한다.

Kind: instance method of [Ellipse](#)

Returns: Boolean - true:같음, false:다름

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |

ellipse.isContains(_geometry) ⇒ Boolean

주어진 공간기하객체를 포함하는지 비교한다.

Kind: instance method of [Ellipse](#)

Returns: Boolean - 포함하면 true

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |

ellipse.isWithin(_geometry) ⇒ Boolean

주어진 공간기하객체에 포함되는지 비교한다.

Kind: instance method of [Ellipse](#)

Returns: Boolean - 포함되면 true

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |

ellipse.getBoundary() ⇒ [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역을 반환한다.

Kind: instance method of [Ellipse](#)

Returns: [Envelope](#) - Envelope 영역

`ellipse.getCentroid()` ⇒ [Coordinate](#)

공간기하객체의 중심좌표를 반환한다.

Kind: instance method of [Ellipse](#)

Returns: [Coordinate](#) - 중심좌표

`ellipse.minDistance(_coordinate)` ⇒ Number

주어진 좌표와의 최단거리를 반환한다.

Kind: instance method of [Ellipse](#)

Returns: Number - 최단거리

| Param | Type | Description |
|--------------------------|----------------------------|-------------|
| <code>_coordinate</code> | Coordinate | 좌표 |

`ellipse.distance(_geometry)` ⇒ Number

주어진 공간기하객체와의 중심점 간의 거리를 반환한다.

Kind: instance method of [Ellipse](#)

Returns: Number - 거리

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | 공간 기하 객체 |

ellipse.getLength() ⇒ Number

공간기하객체의 길이를 반환한다.

Kind: instance method of [Ellipse](#)

Overrides: [getLength](#)

Returns: Number - 길이

ellipse.moveCentroid(중심)

주어진 중심좌표로 공간기하객체를 이동한다.

Kind: instance method of [Ellipse](#)

| Param | Type | Description |
|-------|----------------------------|-------------|
| 중심 | Coordinate | 좌표 |

ellipse.resizeBox(width, height) ⇒ [Geometry](#)

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [Ellipse](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|--------|-------------|
| width | Number | 너비 |

height Number 높이

ellipse.fitToBoundary(envelope) ⇒ [Geometry](#)

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

Kind: instance method of [Ellipse](#)

Returns: [Geometry](#) - 적용된 공간 기하 객체

| Param | Type | Description |
|----------|--------------------------|-------------|
| envelope | Envelope | Envelope 영역 |

ellipse.convertCoordinate(coordinate) ⇒ [Coordinate](#)

파라미터가 [x, y] 형식의 좌표 Array 이면 OG.geometry.Coordinate 인스턴스를 new 하여 반환한다.

Kind: instance method of [Ellipse](#)

| Param | Type | Description |
|------------|---|---|
| coordinate | Coordinate Array. <Number> | [x, y] 형식의 좌표 Array 또는 OG.geometry.Coordinate 인스턴스 |

ellipse.distanceToLine(p, line) ⇒ Number

포인트 P 로부터 라인 AB의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Ellipse](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

ellipse.distanceLineToLine(line1, line2) ⇒ Number

라인1 로부터 라인2 의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Ellipse](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |

ellipse.intersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표들을 반환한다.

Kind: instance method of [Ellipse](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

ellipse.shortestIntersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표중 시작좌표에 가장 가까운 좌표를 반환한다.

Kind: instance method of [Ellipse](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

ellipse.intersectLineToLine(line1, line2, extension) ⇒ [Coordinate](#)

라인1 과 라인2 의 교차점을 계산한다.

Kind: instance method of [Ellipse](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-----------|--|-----------------------|
| line1 | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |
| extension | boolean | 라인을 연장하여 교차점을 계산하는 여부 |

ellipse.intersectCircleToLine(center, radius, from, to) ⇒ [Array.<Coordinate>](#)

주어진 원과 주어진 라인의 교차점을 계산한다.

Kind: instance method of [Ellipse](#)

Returns: [Array.<Coordinate>](#) - 교차점

| Param | Type | Description |
|--------|----------------------------|-------------|
| center | Coordinate | 중심점 |
| radius | Number | 반경 |
| from | Coordinate | 라인 시작좌표 |
| to | Coordinate | 라인 끝좌표 |

ellipse.intersectPointToLine(p, line) ⇒ [Coordinate](#)

포인트 P 로부터 라인 AB 까지 수직인 가상선을 생각할때, 그 교차점을 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Ellipse](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

ellipse.getPercentageDistanceFromPoint(_coordinate) ⇒ Object

주어진 좌표에 대해 공간기하객체 바운더리 대비 가로,세로 위치 퍼센테이지 비율을 구한다.

Kind: instance method of [Ellipse](#)

Returns: Object - {px , py}

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

ellipse.isContainsPoint(_coordinate) ⇒ boolean

공간기하객체가 주어진 좌표를 포함하는지를 반환한다.

Kind: instance method of [Ellipse](#)

Returns: boolean - true,false

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

ellipse.getPointFromPercentageDistance(pXpY) ⇒ [Coordinate](#)

공간기하객체에 대한 퍼센테이지 좌표의 실제 좌표를 구한다.

Kind: instance method of [Ellipse](#)

Returns: [Coordinate](#) - 실 좌표

Param Type Description

pXpY Array 퍼센테이지 좌표

ellipse.getParallelLine(from, to, distance) ⇒ [Array.<Coordinate>](#)

주어진 선분과 일정 거리에 있는 평행한 선분을 반환한다.

Kind: instance method of [Ellipse](#)

Returns: [Array.<Coordinate>](#) - 평행선 시작좌표, 끝좌표 Array

| Param | Type | Description |
|----------|----------------------------|-------------|
| from | Coordinate | 라인 시작좌표 |
| to | Coordinate | 라인 끝좌표 |
| distance | | |

ellipse.getParallelPath(line, distance)

주어진 라인과 일정 거리에 있는 평행한 라인을 반환한다.

Kind: instance method of [Ellipse](#)

| Param | Type | Description |
|-------|--|-------------|
| line | Array.<Coordinate> | 라인 좌표 Array |

distance

ellipse.reset()

저장된 boundary 를 클리어하여 새로 계산하도록 한다.

Kind: instance method of [Ellipse](#)

geometry.Envelope

Kind: static class of [geometry](#)

Requires: module:OG.geometry.Coordinate

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Envelope](#)
 - [new OG.geometry.Envelope\(upperLeft, width, height\)](#)
 - [.getUpperLeft\(\)](#) ⇒ [Coordinate](#)
 - [.setUpperLeft\(upperLeft\)](#)
 - [.getUpperRight\(\)](#) ⇒ [Coordinate](#)
 - [.getLowerRight\(\)](#) ⇒ [Coordinate](#)
 - [.getLowerLeft\(\)](#) ⇒ [Coordinate](#)
 - [.getLeftCenter\(\)](#) ⇒ [Coordinate](#)
 - [.getUpperCenter\(\)](#) ⇒ [Coordinate](#)
 - [.getRightCenter\(\)](#) ⇒ [Coordinate](#)
 - [.getLowerCenter\(\)](#) ⇒ [Coordinate](#)

- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.setCentroid\(centroid\)](#)
- [.getWidth\(\)](#) ⇒ Number
- [.setWidth\(width\)](#)
- [.getHeight\(\)](#) ⇒ Number
- [.setHeight\(height\)](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.isContains\(coordinate\)](#) ⇒ Boolean
- [.isContainsAll\(coordinateArray\)](#) ⇒ Boolean
- [.getHowManyContains\(coordinateArray\)](#) ⇒ Boolean
- [.isContainsOnce\(coordinateArray\)](#) ⇒ Boolean
- [.move\(offsetX, offsetY\)](#) ⇒ [Envelope](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Envelope](#)
- [.isEqual\(Envelope\)](#) ⇒ Boolean
- [.toString\(\)](#) ⇒ String

```
new OG.geometry.Envelope(upperLeft, width, height)
```

2차원 좌표계에서 Envelope 영역을 정의

| Param | Type | Description |
|-----------|-----------------------------------|-------------|
| upperLeft | <u>Coordinate</u> | 기준 좌상단 좌표 |
| width | Number | 너비 |
| height | Number | 높이 |

Example

```
var boundingBox = new OG.geometry.Envelope([50, 50], 200, 100);
```

envelope.getUpperLeft() ⇒ [Coordinate](#)

기준 좌상단 좌표를 반환한다.

Kind: instance method of [Envelope](#)

Returns: [Coordinate](#) - 좌상단 좌표

envelope.setUpperLeft(upperLeft)

주어진 좌표로 기준 좌상단 좌표를 설정한다. 새로 설정된 값으로 이동된다.

Kind: instance method of [Envelope](#)

| Param | Type | Description |
|-----------|----------------------------|-----------------------|
| upperLeft | Coordinate | Array.<Number> 좌상단 좌표 |

envelope.getUpperRight() ⇒ [Coordinate](#)

우상단 좌표를 반환한다.

Kind: instance method of [Envelope](#)

Returns: [Coordinate](#) - 우상단 좌표

envelope.getLowerRight() ⇒ [Coordinate](#)

우하단 좌표를 반환한다.

Kind: instance method of [Envelope](#)

Returns: [Coordinate](#) - 우하단 좌표

envelope.getLowerLeft() ⇒ [Coordinate](#)

좌하단 좌표를 반환한다.

Kind: instance method of [Envelope](#)

Returns: [Coordinate](#) - 좌하단 좌표

envelope.getLeftCenter() ⇒ [Coordinate](#)

좌중간 좌표를 반환한다.

Kind: instance method of [Envelope](#)

Returns: [Coordinate](#) - 좌중간 좌표

envelope.getUpperCenter() ⇒ [Coordinate](#)

상단중간 좌표를 반환한다.

Kind: instance method of [Envelope](#)

Returns: [Coordinate](#) - 상단중간 좌표

envelope.getRightCenter() ⇒ [Coordinate](#)

우중간 좌표를 반환한다.

Kind: instance method of [Envelope](#)

Returns: [Coordinate](#) - 우중간 좌표

`envelope.getLowerCenter()` ⇒ [Coordinate](#)

하단중간 좌표를 반환한다.

Kind: instance method of [Envelope](#)

Returns: [Coordinate](#) - 하단중간 좌표

`envelope.getCentroid()` ⇒ [Coordinate](#)

Envelope 의 중심좌표를 반환한다.

Kind: instance method of [Envelope](#)

Returns: [Coordinate](#) - 중심좌표

`envelope.setCentroid(centroid)`

주어진 좌표로 중심 좌표를 설정한다. 새로 설정된 값으로 이동된다.

Kind: instance method of [Envelope](#)

| Param | Type | Description |
|----------|---|-------------|
| centroid | Coordinate Array.<Number> | 중심좌표 |

`envelope.getWidth() ⇒ Number`

Envelope 의 가로 사이즈를 반환한다.

Kind: instance method of [Envelope](#)

Returns: Number - 너비

`envelope.setWidth(width)`

주어진 값으로 Envelope 의 가로 사이즈를 설정한다.

Kind: instance method of [Envelope](#)

| Param | Type | Description |
|-------|--------|-------------|
| width | Number | 너비 |

`envelope.getHeight() ⇒ Number`

Envelope 의 세로 사이즈를 반환한다.

Kind: instance method of [Envelope](#)

Returns: Number - 높이

`envelope.setHeight(height)`

주어진 값으로 Envelope 의 세로 사이즈를 설정한다.

Kind: instance method of [Envelope](#)

| Param | Type | Description |
|--------|--------|-------------|
| height | Number | 높이 |

envelope.getVertices() ⇒ [Array.<Coordinate>](#)

Envelope 모든 꼭지점을 반환한다.

좌상단좌표부터 시계방향으로 꼭지점 Array 를 반환한다.

Kind: instance method of [Envelope](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array : [좌상단, 상단중간, 우상단, 우중간, 우하단, 하단중간, 좌하단, 좌중간, 좌상단]

envelope.isContains(coordinate) ⇒ Boolean

주어진 좌표값이 Envelope 영역에 포함되는지 비교한다.

Kind: instance method of [Envelope](#)

Returns: Boolean - true:포함, false:비포함

| Param | Type | Description |
|------------|----------------------------|-------------|
| coordinate | Coordinate | 좌표값 |

envelope.isContainsAll(coordinateArray) ⇒ Boolean

주어진 모든 좌표값이 Envelope 영역에 포함되는지 비교한다.

Kind: instance method of [Envelope](#)

Returns: Boolean - true:포함, false:비포함

| Param | Type | Description |
|-----------------|--|-------------|
| coordinateArray | Array.<Coordinate> | 좌표값 Array |

envelope.getHowManyContains(coordinateArray) ⇒ Boolean

주어진 모든 좌표값이 Envelope 영역에 포함되는지 비교한다.

Kind: instance method of [Envelope](#)

Returns: Boolean - true:포함, false:비포함

| Param | Type | Description |
|-----------------|--|-------------|
| coordinateArray | Array.<Coordinate> | 좌표값 Array |

envelope.isContainsOnce(coordinateArray) ⇒ Boolean

주어진 모든 좌표값이 Envelope 영역에 포함되는지 비교한다.

Kind: instance method of [Envelope](#)

Returns: Boolean - true:포함, false:비포함

| Param | Type | Description |
|-----------------|--|-------------|
| coordinateArray | Array.<Coordinate> | 좌표값 Array |

envelope.move(offsetX, offsetY) ⇒ [Envelope](#)

크기는 고정된 채 가로, 세로 Offset 만큼 Envelope 을 이동한다.

Kind: instance method of [Envelope](#)

Returns: [Envelope](#) - 이동된 Envelope

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

envelope.resize(upper, lower, left, right) ⇒ [Envelope](#)

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance method of [Envelope](#)

Returns: [Envelope](#) - 리사이즈된 Envelope

| Param | Type | Description |
|-------|--------|----------------------------|
| upper | Number | 상단 라인 이동 Offset(위 방향으로 +) |
| lower | Number | 하단 라인 이동 Offset(아래 방향으로 +) |
| left | Number | 좌측 라인 이동 Offset(좌측 방향으로 +) |
| right | Number | 우측 라인 이동 Offset(우측 방향으로 +) |

envelope.isEquals(Envelope) ⇒ Boolean

주어진 Envelope 영역과 같은지 비교한다.

Kind: instance method of [Envelope](#)

Returns: Boolean - true:같음, false:다름

| Param | Type | Description |
|----------|--------------------------|-------------|
| Envelope | Envelope | 영역 |

`envelope.toString() ⇒ String`

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [Envelope](#)

Returns: String - 프라퍼티 정보

`geometry.Geometry`

Kind: static class of [geometry](#)

Requires: module:OG.geometry.Coordinate, module:OG.geometry.Envelope

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Geometry](#)
 - [new OG.geometry.Geometry\(\)](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.isEqual\(_geometry\)](#) ⇒ Boolean
 - [.isContains\(_geometry\)](#) ⇒ Boolean
 - [.isWithin\(_geometry\)](#) ⇒ Boolean
 - [.getBoundary\(\)](#) ⇒ [Envelope](#)
 - [.getCentroid\(\)](#) ⇒ [Coordinate](#)
 - [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
 - [.minDistance\(_coordinate\)](#) ⇒ Number

- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.moveCentroid\(중심\)](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

`new OG.geometry.Geometry()`

공간 기하 객체(Spatial Geometry Object)의 최상위 추상 클래스

`geometry.TYPE` : Number

공간 기하 객체 타입

Kind: instance property of [Geometry](#)

`geometry.IS_CLOSED` : Boolean

닫힌 기하 객체 인지 여부

Kind: instance property of [Geometry](#)

`geometry.style` : [Style](#)

스타일 속성

Kind: instance property of [Geometry](#)

`geometry.boundary` : [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역

Kind: instance property of [Geometry](#)

`geometry.isEquals(_geometry) ⇒ Boolean`

주어진 Geometry 객체와 같은지 비교한다.

Kind: instance method of [Geometry](#)

Returns: Boolean - true:같음, false:다름

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`geometry.isContains(_geometry) ⇒ Boolean`

주어진 공간기하객체를 포함하는지 비교한다.

Kind: instance method of [Geometry](#)

Returns: Boolean - 포함하면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`geometry.isWithin(_geometry) ⇒ Boolean`

주어진 공간기하객체에 포함되는지 비교한다.

Kind: instance method of [Geometry](#)

Returns: Boolean - 포함되면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`geometry.getBoundary()` ⇒ [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역을 반환한다.

Kind: instance method of [Geometry](#)

Returns: [Envelope](#) - Envelope 영역

`geometry.getCentroid()` ⇒ [Coordinate](#)

공간기하객체의 중심좌표를 반환한다.

Kind: instance method of [Geometry](#)

Returns: [Coordinate](#) - 중심좌표

`geometry.getVertices()` ⇒ [Array.<Coordinate>](#)

공간기하객체의 모든 꼭지점을 반환한다.

Kind: instance abstract method of [Geometry](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array

`geometry.minDistance(_coordinate)` ⇒ Number

주어진 좌표와의 최단거리를 반환한다.

Kind: instance method of [Geometry](#)

Returns: Number - 최단거리

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

`geometry.distance(_geometry) ⇒ Number`

주어진 공간기하객체와의 중심점 간의 거리를 반환한다.

Kind: instance method of [Geometry](#)

Returns: Number - 거리

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | 공간 기하 객체 |

`geometry.getLength() ⇒ Number`

공간기하객체의 길이를 반환한다.

Kind: instance method of [Geometry](#)

Returns: Number - 길이

`geometry.move(offsetX, offsetY) ⇒ Geometry`

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance abstract method of [Geometry](#)

Returns: [Geometry](#) - 이동된 공간 기하 객체

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

`geometry.moveCentroid(중심)`

주어진 중심좌표로 공간기하객체를 이동한다.

Kind: instance method of [Geometry](#)

| Param | Type | Description |
|-------|----------------------------|-------------|
| 중심 | Coordinate | 좌표 |

`geometry.resize(upper, lower, left, right) ⇒ Geometry`

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance abstract method of [Geometry](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|--------|----------------------------|
| upper | Number | 상단 라인 이동 Offset(위 방향으로 +) |
| lower | Number | 하단 라인 이동 Offset(아래 방향으로 +) |
| left | Number | 좌측 라인 이동 Offset(좌측 방향으로 +) |
| right | Number | 우측 라인 이동 Offset(우측 방향으로 +) |

`geometry.resizeBox(width, height) ⇒ Geometry`

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [Geometry](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|--------|--------|-------------|
| width | Number | 너비 |
| height | Number | 높이 |

geometry.rotate(angle, origin) ⇒ [Geometry](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance abstract method of [Geometry](#)

Returns: [Geometry](#) - 회전된 공간 기하 객체

| Param | Type | Description |
|--------|----------------------------|---------------------|
| angle | Number | 회전 각도 |
| origin | Coordinate | 기준 좌표(default:중심좌표) |

geometry.fitToBoundary(envelope) ⇒ [Geometry](#)

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

Kind: instance method of [Geometry](#)

Returns: [Geometry](#) - 적용된 공간 기하 객체

| Param | Type | Description |
|----------|--------------------------|-------------|
| envelope | Envelope | Envelope 영역 |

geometry.convertCoordinate(coordinate) ⇒ [Coordinate](#)

파라미터가 [x, y] 형식의 좌표 Array 이면 OG.geometry.Coordinate 인스턴스를 new 하여 반환한다.

Kind: instance method of [Geometry](#)

| Param | Type | Description |
|------------|---|--|
| coordinate | Coordinate Array.<Number> | [x, y] 형식의 좌표 Array 또는 OG.geometry.Coordinate 인스턴스 |

geometry.distanceToLine(p, line) ⇒ Number

포인트 P 로부터 라인 AB의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Geometry](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

geometry.distanceLineToLine(line1, line2) ⇒ Number

라인1 로부터 라인2 의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Geometry](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |

geometry.intersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표들을 반환한다.

Kind: instance method of [Geometry](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

geometry.shortestIntersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표중 시작좌표에 가장 가까운 좌표를 반환한다.

Kind: instance method of [Geometry](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

geometry.intersectLineToLine(line1, line2, extension) ⇒ [Coordinate](#)

라인1 과 라인2 의 교차점을 계산한다.

Kind: instance method of [Geometry](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-----------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |
| extension | boolean | 라인을 연장하여 교차점을 계산하는 여부 |

geometry.intersectCircleToLine(center, radius, from, to) ⇒ [Array.<Coordinate>](#)

주어진 원과 주어진 라인의 교차점을 계산한다.

Kind: instance method of [Geometry](#)

Returns: [Array.<Coordinate>](#) - 교차점

| Param | Type | Description |
|--------|----------------------------|--------------|
| center | Coordinate | 중심점 |
| radius | Number | 반경 |
| from | Coordinate | line 라인 시작좌표 |
| to | Coordinate | line 라인 끝좌표 |

geometry.intersectPointToLine(p, line) ⇒ [Coordinate](#)

포인트 P 로부터 라인 AB 까지 수직인 가상선을 생각할때, 그 교차점을 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Geometry](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

geometry.getPercentageDistanceFromPoint(_coordinate) ⇒ Object

주어진 좌표에 대해 공간기하객체 바운더리 대비 가로,세로 위치 퍼센테이지 비율을 구한다.

Kind: instance method of [Geometry](#)

Returns: Object - {px , py}

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

geometry.isContainsPoint(_coordinate) ⇒ boolean

공간기하객체가 주어진 좌표를 포함하는지를 반환한다.

Kind: instance method of [Geometry](#)

Returns: boolean - true,false

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

geometry.getPointFromPercentageDistance(pXpY) ⇒ [Coordinate](#)

공간기하객체에 대한 퍼센테이지 좌표의 실제 좌표를 구한다.

Kind: instance method of [Geometry](#)

Returns: [Coordinate](#) - 실 좌표

| Param | Type | Description |
|-------|-------|-------------|
| pXpY | Array | 퍼센테이지 좌표 |

geometry.getParallelLine(from, to, distance) ⇒ [Array.<Coordinate>](#)

주어진 선분과 일정 거리에 있는 평행한 선분을 반환한다.

Kind: instance method of [Geometry](#)

Returns: [Array.<Coordinate>](#) - 평행선 시작좌표, 끝좌표 Array

| Param | Type | Description |
|----------|----------------------------|-------------|
| from | Coordinate | 라인 시작좌표 |
| to | Coordinate | 라인 끝좌표 |
| distance | | |

`geometry.getParallelPath(line, distance)`

주어진 라인과 일정 거리에 있는 평행한 라인을 반환한다.

Kind: instance method of [Geometry](#)

| Param | Type | Description |
|----------|--|-------------|
| line | Array.<Coordinate> | 라인 좌표 Array |
| distance | | |

`geometry.reset()`

저장된 boundary 를 클리어하여 새로 계산하도록 한다.

Kind: instance method of [Geometry](#)

`geometry.GeometryCollection` \Leftarrow [Geometry](#)

Kind: static class of [geometry](#)

Extends: [Geometry](#)

Requires: module:OG.geometry.Coordinate, module:OG.geometry.Envelope,
module:OG.geometry.Geometry

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.GeometryCollection](#) \Leftarrow [Geometry](#)
 - [new OG.geometry.GeometryCollection geometries](#)
 - [.geometries](#) : [Array.<Geometry>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.toString\(\)](#) \Rightarrow String
 - [.isEqual\(_geometry\)](#) \Rightarrow Boolean
 - [.isContains\(_geometry\)](#) \Rightarrow Boolean
 - [.isWithin\(_geometry\)](#) \Rightarrow Boolean
 - [.getBoundary\(\)](#) \Rightarrow [Envelope](#)
 - [.getCentroid\(\)](#) \Rightarrow [Coordinate](#)
 - [.getVertices\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.minDistance\(_coordinate\)](#) \Rightarrow Number
 - [.distance\(_geometry\)](#) \Rightarrow Number
 - [.getLength\(\)](#) \Rightarrow Number
 - [.move\(offsetX, offsetY\)](#) \Rightarrow [Geometry](#)
 - [.moveCentroid\(중심\)](#)
 - [.resize\(upper, lower, left, right\)](#) \Rightarrow [Geometry](#)

- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

new OG.geometry.GeometryCollection geometries)

공간 기하 객체(Spatial Geometry Object) Collection

| Param | Type | Description |
|------------|---|----------------|
| geometries | <u>Array.<Geometry></u> | 공간 기하 객체 Array |

Example

```
var geom1 = new OG.geometry.Point([20, 5]),  
    geom2 = new OG.geometry.Line([20, 5], [30, 15]),  
    geom3 = new OG.geometry.PolyLine([[20, 5], [30, 15], [40, 25], [50, 15]]);  
  
var collection = new OG.geometry.GeometryCollection([geom1, geom2, geom3]);
```

geometryCollection.geometries : [Array.<Geometry>](#)

공간 기하 객체 Array

Kind: instance property of [GeometryCollection](#)

geometryCollection.TYPE : Number

공간 기하 객체 타입

Kind: instance property of [GeometryCollection](#)

Overrides: [TYPE](#)

geometryCollection.IS_CLOSED : Boolean

닫힌 기하 객체 인지 여부

Kind: instance property of [GeometryCollection](#)

geometryCollection.style : [Style](#)

스타일 속성

Kind: instance property of [GeometryCollection](#)

Overrides: [style](#)

`geometryCollection.boundary` : [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역

Kind: instance property of [GeometryCollection](#)

`geometryCollection.toString()` ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [GeometryCollection](#)

Returns: String - 프라퍼티 정보

`geometryCollection.isEquals(_geometry)` ⇒ Boolean

주어진 Geometry 객체와 같은지 비교한다.

Kind: instance method of [GeometryCollection](#)

Returns: Boolean - true:같음, false:다름

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`geometryCollection.isContains(_geometry) ⇒ Boolean`

주어진 공간기하객체를 포함하는지 비교한다.

Kind: instance method of [GeometryCollection](#)

Returns: Boolean - 포함하면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`geometryCollection.isWithin(_geometry) ⇒ Boolean`

주어진 공간기하객체에 포함되는지 비교한다.

Kind: instance method of [GeometryCollection](#)

Returns: Boolean - 포함되면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`geometryCollection.getBoundary() ⇒ Envelope`

공간기하객체를 포함하는 사각형의 Boundary 영역을 반환한다.

Kind: instance method of [GeometryCollection](#)

Returns: [Envelope](#) - Envelope 영역

`geometryCollection.getCentroid()` ⇒ [Coordinate](#)

공간기하객체의 중심좌표를 반환한다.

Kind: instance method of [GeometryCollection](#)

Returns: [Coordinate](#) - 중심좌표

`geometryCollection.getVertices()` ⇒ [Array.<Coordinate>](#)

공간기하객체의 모든 꼭지점을 반환한다.

Kind: instance method of [GeometryCollection](#)

Overrides: [getVertices](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array

`geometryCollection.minDistance(_coordinate)` ⇒ Number

주어진 좌표와의 최단거리를 반환한다.

Kind: instance method of [GeometryCollection](#)

Returns: Number - 최단거리

| Param | Type | Description |
|--------------------------|----------------------------|-------------|
| <code>_coordinate</code> | Coordinate | 좌표 |

`geometryCollection.distance(_geometry)` ⇒ Number

주어진 공간기하객체와의 중심점 간의 거리를 반환한다.

Kind: instance method of [GeometryCollection](#)

Returns: Number - 거리

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | 공간 기하 객체 |

`geometryCollection.getLength()` ⇒ Number

공간기하객체의 길이를 반환한다.

Kind: instance method of [GeometryCollection](#)

Returns: Number - 길이

`geometryCollection.move(offsetX, offsetY)` ⇒ [Geometry](#)

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance method of [GeometryCollection](#)

Overrides: [move](#)

Returns: [Geometry](#) - 이동된 공간 기하 객체

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

`geometryCollection.moveCentroid(중심)`

주어진 중심좌표로 공간기하객체를 이동한다.

Kind: instance method of [GeometryCollection](#)

| Param | Type | Description |
|-------|----------------------------|-------------|
| 중심 | Coordinate | 좌표 |

`geometryCollection.resize(upper, lower, left, right)` ⇒ [Geometry](#)

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance method of [GeometryCollection](#)

Overrides: [resize](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|--------|----------------------------|
| upper | Number | 상단 라인 이동 Offset(위 방향으로 +) |
| lower | Number | 하단 라인 이동 Offset(아래 방향으로 +) |
| left | Number | 좌측 라인 이동 Offset(좌측 방향으로 +) |
| right | Number | 우측 라인 이동 Offset(우측 방향으로 +) |

`geometryCollection.resizeBox(width, height)` ⇒ [Geometry](#)

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [GeometryCollection](#)

Overrides: [resizeBox](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|--------|--------|-------------|
| width | Number | 너비 |
| height | Number | 높이 |

geometryCollection.rotate(angle, origin) ⇒ [Geometry](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [GeometryCollection](#)

Overrides: [rotate](#)

Returns: [Geometry](#) - 회전된 공간 기하 객체

| Param | Type | Description |
|--------|----------------------------|----------------------|
| angle | Number | 회전 각도 |
| origin | Coordinate | 기준 좌표(default: 중심좌표) |

geometryCollection.fitToBoundary(envelope) ⇒ [Geometry](#)

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

Kind: instance method of [GeometryCollection](#)

Overrides: [fitToBoundary](#)

Returns: [Geometry](#) - 적용된 공간 기하 객체

| Param | Type | Description |
|----------|--------------------------|-------------|
| envelope | Envelope | Envelope 영역 |

geometryCollection.convertCoordinate(coordinate) ⇒ [Coordinate](#)

파라미터가 [x, y] 형식의 좌표 Array 이면 OG.geometry.Coordinate 인스턴스를 new 하여 반환한다.

Kind: instance method of [GeometryCollection](#)

| Param | Type | Description |
|------------|---|---|
| coordinate | Coordinate Array. <Number> | [x, y] 형식의 좌표 Array 또는 OG.geometry.Coordinate 인스턴스 |

geometryCollection.distanceToLine(p, line) ⇒ Number

포인트 P 로부터 라인 AB의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [GeometryCollection](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

geometryCollection.distanceLineToLine(line1, line2) ⇒ Number

라인1 로부터 라인2 의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [GeometryCollection](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |

geometryCollection.intersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표들을 반환한다.

Kind: instance method of [GeometryCollection](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

geometryCollection.shortestIntersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표중 시작좌표에 가장 가까운 좌표를 반환한다.

Kind: instance method of [GeometryCollection](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

geometryCollection.intersectLineToLine(line1, line2, extension) ⇒ [Coordinate](#)

라인1 과 라인2 의 교차점을 계산한다.

Kind: instance method of [GeometryCollection](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-----------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |
| extension | boolean | 라인을 연장하여 교차점을 계산하는 여부 |

geometryCollection.intersectCircleToLine(center, radius, from, to) ⇒ [Array.<Coordinate>](#)

주어진 원과 주어진 라인의 교차점을 계산한다.

Kind: instance method of [GeometryCollection](#)

Returns: [Array.<Coordinate>](#) - 교차점

| Param | Type | Description |
|--------|----------------------------|--------------|
| center | Coordinate | 중심점 |
| radius | Number | 반경 |
| from | Coordinate | line 라인 시작좌표 |
| to | Coordinate | line 라인 끝좌표 |

`geometryCollection.intersectPointToLine(p, line) ⇒ Coordinate`

포인트 P로부터 라인 AB까지 수직인 가상선을 생각할때, 그 교차점을 계산한다.

Note: NON-ROBUST!

Kind: instance method of [GeometryCollection](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-------|---|------------------------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

`geometryCollection.getPercentageDistanceFromPoint(_coordinate) ⇒ Object`

주어진 좌표에 대해 공간기하객체 바운더리 대비 가로,세로 위치 퍼센테이지 비율을 구한다.

Kind: instance method of [GeometryCollection](#)

Returns: Object - {px , py}

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

geometryCollection.isContainsPoint(_coordinate) ⇒ boolean

공간기하객체가 주어진 좌표를 포함하는지를 반환한다.

Kind: instance method of [GeometryCollection](#)

Returns: boolean - true,false

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

geometryCollection.getPointFromPercentageDistance(pXpY) ⇒ [Coordinate](#)

공간기하객체에 대한 퍼센테이지 좌표의 실제 좌표를 구한다.

Kind: instance method of [GeometryCollection](#)

Returns: [Coordinate](#) - 실 좌표

| Param | Type | Description |
|-------|-------|-------------|
| pXpY | Array | 퍼센테이지 좌표 |

geometryCollection.getParallelLine(from, to, distance) ⇒ [Array.<Coordinate>](#)

주어진 선분과 일정 거리에 있는 평행한 선분을 반환한다.

Kind: instance method of [GeometryCollection](#)

Returns: [Array.<Coordinate>](#) - 평행선 시작좌표, 끝좌표 Array

| Param | Type | Description |
|----------|----------------------------|-------------|
| from | Coordinate | 라인 시작좌표 |
| to | Coordinate | 라인 끝좌표 |
| distance | | |

`geometryCollection.getParallelPath(line, distance)`

주어진 라인과 일정 거리에 있는 평행한 라인을 반환한다.

Kind: instance method of [GeometryCollection](#)

| Param | Type | Description |
|----------|--|-------------|
| line | Array.<Coordinate> | 라인 좌표 Array |
| distance | | |

`geometryCollection.reset()`

저장된 boundary 를 클리어하여 새로 계산하도록 한다.

Kind: instance method of [GeometryCollection](#)

`geometry.Line` \Leftarrow [PolyLine](#)

Kind: static class of [geometry](#)

Extends: [PolyLine](#)

Requires: module:OG.geometry.Coordinate, module:OG.geometry.Envelope,
module:OG.geometry.Geometry

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Line](#) \Leftarrow [PolyLine](#)
 - [new OG.geometry.Line\(from, to\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getVertices\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) \Rightarrow [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) \Rightarrow [Geometry](#)
 - [.rotate\(angle, origin\)](#) \Rightarrow [Geometry](#)
 - [.toString\(\)](#) \Rightarrow String
 - [.angleBetweenPoints\(prev, next\)](#) \Rightarrow Number
 - [.isRightAngleBetweenPoints\(prev, next\)](#) \Rightarrow Object
 - [.angleBetweenThreePoints\(prev, next\)](#) \Rightarrow Number
 - [.isEquals\(_geometry\)](#) \Rightarrow Boolean
 - [.isContains\(_geometry\)](#) \Rightarrow Boolean
 - [.isWithin\(_geometry\)](#) \Rightarrow Boolean
 - [.getBoundary\(\)](#) \Rightarrow [Envelope](#)
 - [.getCentroid\(\)](#) \Rightarrow [Coordinate](#)
 - [.minDistance\(_coordinate\)](#) \Rightarrow Number
 - [.distance\(_geometry\)](#) \Rightarrow Number
 - [.getLength\(\)](#) \Rightarrow Number
 - [.moveCentroid\(중심\)](#)
 - [.resizeBox\(width, height\)](#) \Rightarrow [Geometry](#)

- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

new OG.geometry.Line(from, to)

Line 공간 기하 객체(Spatial Geometry Object)

| Param | Type | Description |
|-------|-----------------------------------|-------------|
| from | <u>Coordinate</u> | 라인 시작 좌표값 |
| to | <u>Coordinate</u> | 라인 끝 좌표값 |

Example


```
var geom = new OG.geometry.Line([20, 5], [30, 15]);
```

line.vertices : [Array.<Coordinate>](#)

Line Vertex 좌표 Array

Kind: instance property of [Line](#)

line.TYPE : Number

공간 기하 객체 타입

Kind: instance property of [Line](#)

Overrides: [TYPE](#)

line.IS_CLOSED : Boolean

닫힌 기하 객체 인지 여부

Kind: instance property of [Line](#)

line.style : [Style](#)

스타일 속성

Kind: instance property of [Line](#)

Overrides: [style](#)

line.boundary : [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역

Kind: instance property of [Line](#)

line.getVertices() ⇒ [Array.<Coordinate>](#)

공간기하객체의 모든 꼭지점을 반환한다.

Kind: instance method of [Line](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array

line.move(offsetX, offsetY) ⇒ [Geometry](#)

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance method of [Line](#)

Returns: [Geometry](#) - 이동된 공간 기하 객체

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

line.resize(upper, lower, left, right) ⇒ [Geometry](#)

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance method of [Line](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------------|----------------------------|-------------|
| upperNumber | 상단 라인 이동 Offset(위 방향으로 +) | |
| lowerNumber | 하단 라인 이동 Offset(아래 방향으로 +) | |
| left | 좌측 라인 이동 Offset(좌측 방향으로 +) | |
| right | 우측 라인 이동 Offset(우측 방향으로 +) | |

line.rotate(angle, origin) ⇒ [Geometry](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [Line](#)

Returns: [Geometry](#) - 회전된 공간 기하 객체

| Param | Type | Description |
|--------|----------------------------|---------------------|
| angle | Number | 회전 각도 |
| origin | Coordinate | 기준 좌표(default:중심좌표) |

line.toString() ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [Line](#)

Overrides: [toString](#)

Returns: String - 프라퍼티 정보

`line.angleBetweenPoints(prev, next) ⇒ Number`

공간기하객체의 두 꼭지점 사이에 가상의 선을 그렸을때, 그 기울기를 구한다.

Kind: instance method of [Line](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|----------------------------|-------------|
| prev | Coordinate | 꼭지점 1 |
| next | Coordinate | 꼭지점 2 |

`line.isRightAngleBetweenPoints(prev, next) ⇒ Object`

공간기하객체의 두 꼭지점 사이의 기울기가 수평또는 수직인지 판별한다.

Kind: instance method of [Line](#)

Returns: Object - {flag : true or false, type: horizontal or vertical or none}

| Param | Type | Description |
|-------|----------------------------|-------------|
| prev | Coordinate | 꼭지점 1 |
| next | Coordinate | 꼭지점 2 |

`line.angleBetweenThreePoints(prev, next) ⇒ Number`

공간기하객체의 세 꼭지점 사이의 각도 중 작은 각도를 반환한다.

Kind: instance method of [Line](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

line.isEquals(_geometry) ⇒ Boolean

주어진 Geometry 객체와 같은지 비교한다.

Kind: instance method of [Line](#)

Returns: Boolean - true:같은, false:다름

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |

line.isContains(_geometry) ⇒ Boolean

주어진 공간기하객체를 포함하는지 비교한다.

Kind: instance method of [Line](#)

Returns: Boolean - 포함하면 true

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |

line.isWithin(_geometry) ⇒ Boolean

주어진 공간기하객체에 포함되는지 비교한다.

Kind: instance method of [Line](#)

Returns: Boolean - 포함되면 true

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |

line.getBoundary() ⇒ [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역을 반환한다.

Kind: instance method of [Line](#)

Returns: [Envelope](#) - Envelope 영역

line.getCentroid() ⇒ [Coordinate](#)

공간기하객체의 중심좌표를 반환한다.

Kind: instance method of [Line](#)

Returns: [Coordinate](#) - 중심좌표

line.minDistance(_coordinate) ⇒ Number

주어진 좌표와의 최단거리를 반환한다.

Kind: instance method of [Line](#)

Returns: Number - 최단거리

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

line.distance(_geometry) ⇒ Number

주어진 공간기하객체와의 중심점 간의 거리를 반환한다.

Kind: instance method of [Line](#)

Returns: Number - 거리

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | 공간 기하 객체 |

line.getLength() ⇒ Number

공간기하객체의 길이를 반환한다.

Kind: instance method of [Line](#)

Returns: Number - 길이

line.moveCentroid(중심)

주어진 중심좌표로 공간기하객체를 이동한다.

Kind: instance method of [Line](#)

| Param | Type | Description |
|-------|----------------------------|-------------|
| 중심 | Coordinate | 좌표 |

line.resizeBox(width, height) ⇒ [Geometry](#)

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [Line](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|--------|--------|-------------|
| width | Number | 너비 |
| height | Number | 높이 |

line.fitToBoundary(envelope) ⇒ [Geometry](#)

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

Kind: instance method of [Line](#)

Returns: [Geometry](#) - 적용된 공간 기하 객체

| Param | Type | Description |
|----------|--------------------------|-------------|
| envelope | Envelope | Envelope 영역 |

line.convertCoordinate(coordinate) ⇒ [Coordinate](#)

파라미터가 [x, y] 형식의 좌표 Array 이면 OG.geometry.Coordinate 인스턴스를 new 하여 반환한다.

Kind: instance method of [Line](#)

| Param | Type | Description |
|------------|---|---|
| coordinate | Coordinate Array. <Number> | [x, y] 형식의 좌표 Array 또는 OG.geometry.Coordinate 인스턴스 |

line.distanceToLine(p, line) ⇒ Number

포인트 P로부터 라인 AB의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Line](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------|
| p | Coordinate | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

line.distanceLineToLine(line1, line2) ⇒ Number

라인1로부터 라인2의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Line](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------|
| line1 | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

line.intersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표들을 반환한다.

Kind: instance method of [Line](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

line.shortestIntersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표중 시작좌표에 가장 가까운 좌표를 반환한다.

Kind: instance method of [Line](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

line.intersectLineToLine(line1, line2, extension) ⇒ [Coordinate](#)

라인1 과 라인2 의 교차점을 계산한다.

Kind: instance method of [Line](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-----------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |
| extension | boolean | 라인을 연장하여 교차점을 계산하는 여부 |

line.intersectCircleToLine(center, radius, from, to) ⇒ [Array.<Coordinate>](#)

주어진 원과 주어진 라인의 교차점을 계산한다.

Kind: instance method of [Line](#)

Returns: [Array.<Coordinate>](#) - 교차점

| Param | Type | Description |
|--------|----------------------------|--------------|
| center | Coordinate | 중심점 |
| radius | Number | 반경 |
| from | Coordinate | line 라인 시작좌표 |
| to | Coordinate | line 라인 끝좌표 |

line.intersectPointToLine(p, line) ⇒ [Coordinate](#)

포인트 P 로부터 라인 AB 까지 수직인 가상선을 생각할때, 그 교차점을 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Line](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

line.getPercentageDistanceFromPoint(_coordinate) ⇒ Object

주어진 좌표에 대해 공간기하객체 바운더리 대비 가로,세로 위치 퍼센테이지 비율을 구한다.

Kind: instance method of [Line](#)

Returns: Object - {px , py}

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

line.isContainsPoint(_coordinate) ⇒ boolean

공간기하객체가 주어진 좌표를 포함하는지를 반환한다.

Kind: instance method of [Line](#)

Returns: boolean - true,false

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

line.getPointFromPercentageDistance(pXpY) ⇒ [Coordinate](#)

공간기하객체에 대한 퍼센테이지 좌표의 실제 좌표를 구한다.

Kind: instance method of [Line](#)

Returns: [Coordinate](#) - 실 좌표

| Param | Type | Description |
|-------|-------|-------------|
| pXpY | Array | 퍼센테이지 좌표 |

line.getParallelLine(from, to, distance) ⇒ [Array.<Coordinate>](#)

주어진 선분과 일정 거리에 있는 평행한 선분을 반환한다.

Kind: instance method of [Line](#)

Returns: [Array.<Coordinate>](#) - 평행선 시작좌표, 끝좌표 Array

| Param | Type | Description |
|----------|----------------------------|-------------|
| from | Coordinate | 라인 시작좌표 |
| to | Coordinate | 라인 끝좌표 |
| distance | | |

`line.getParallelPath(line, distance)`

주어진 라인과 일정 거리에 있는 평행한 라인을 반환한다.

Kind: instance method of [Line](#)

| Param | Type | Description |
|----------|--|-------------|
| line | Array.<Coordinate> | 라인 좌표 Array |
| distance | | |

`line.reset()`

저장된 boundary 를 클리어하여 새로 계산하도록 한다.

Kind: instance method of [Line](#)

`geometry.Point` \Leftarrow [Geometry](#)

Kind: static class of [geometry](#)

Extends: [Geometry](#)

Requires: module:OG.geometry.Coordinate, module:OG.geometry.Envelope,
module:OG.geometry.Geometry

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Point](#) \Leftarrow [Geometry](#)
 - [new OG.geometry.Point\(coordinate\)](#)
 - [.coordinate](#) : [Coordinate](#)
 - [.vertices](#) : [Array.<Coordinate>](#)

- [.TYPE](#) : Number
- [.IS_CLOSED](#) : Boolean
- [.style](#) : [Style](#)
- [.boundary](#) : [Envelope](#)
- [.toString\(\)](#) ⇒ String
- [.isEqual\(_geometry\)](#) ⇒ Boolean
- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.moveCentroid\(중심\)](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)

- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ [Object](#)
- [.isContainsPoint\(_coordinate\)](#) ⇒ [boolean](#)
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

new OG.geometry.Point(coordinate)

Point 공간 기하 객체(Spatial Geometry Object)

| Param | Type | Description |
|------------|-----------------------------------|-------------|
| coordinate | <u>Coordinate</u> | 좌표값 |

Example

```
var geom = new OG.geometry.Point([20, 5]);
```

point.coordinate : [Coordinate](#)

좌표값

Kind: instance property of [Point](#)

point.vertices : [Array.<Coordinate>](#)

Line Vertex 좌표 Array

Kind: instance property of [Point](#)

point.TYPE : Number

공간 기하 객체 타입

Kind: instance property of [Point](#)

Overrides: [TYPE](#)

point.IS_CLOSED : Boolean

닫힌 기하 객체 인지 여부

Kind: instance property of [Point](#)

point.style : [Style](#)

스타일 속성

Kind: instance property of [Point](#)

Overrides: [style](#)

point.boundary : [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역

Kind: instance property of [Point](#)

Overrides: [boundary](#)

point.toString() ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [Point](#)

Returns: String - 프라퍼티 정보

point.isEquals(_geometry) ⇒ Boolean

주어진 Geometry 객체와 같은지 비교한다.

Kind: instance method of [Point](#)

Returns: Boolean - true:같은, false:다름

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |

point.isContains(_geometry) ⇒ Boolean

주어진 공간기하객체를 포함하는지 비교한다.

Kind: instance method of [Point](#)

Returns: Boolean - 포함하면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`point.isWithin(_geometry) ⇒ Boolean`

주어진 공간기하객체에 포함되는지 비교한다.

Kind: instance method of [Point](#)

Returns: Boolean - 포함되면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`point.getBoundary() ⇒ Envelope`

공간기하객체를 포함하는 사각형의 Boundary 영역을 반환한다.

Kind: instance method of [Point](#)

Returns: [Envelope](#) - Envelope 영역

`point.getCentroid() ⇒ Coordinate`

공간기하객체의 중심좌표를 반환한다.

Kind: instance method of [Point](#)

Returns: [Coordinate](#) - 중심좌표

point.getVertices() ⇒ [Array.<Coordinate>](#)

공간기하객체의 모든 꼭지점을 반환한다.

Kind: instance method of [Point](#)

Overrides: [getVertices](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array

point.minDistance(_coordinate) ⇒ Number

주어진 좌표와의 최단거리를 반환한다.

Kind: instance method of [Point](#)

Returns: Number - 최단거리

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

point.distance(_geometry) ⇒ Number

주어진 공간기하객체와의 중심점 간의 거리를 반환한다.

Kind: instance method of [Point](#)

Returns: Number - 거리

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | 공간 기하 객체 |

`point.getLength()` ⇒ Number

공간기하객체의 길이를 반환한다.

Kind: instance method of [Point](#)

Returns: Number - 길이

`point.move(offsetX, offsetY)` ⇒ [Geometry](#)

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance method of [Point](#)

Overrides: [move](#)

Returns: [Geometry](#) - 이동된 공간 기하 객체

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

`point.moveCentroid(중심)`

주어진 중심좌표로 공간기하객체를 이동한다.

Kind: instance method of [Point](#)

Overrides: [moveCentroid](#)

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|----|----------------------------|----|
| 중심 | Coordinate | 좌표 |
|----|----------------------------|----|

point.resize(upper, lower, left, right) ⇒ [Geometry](#)

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance method of [Point](#)

Overrides: [resize](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|-------|--------|---------------------------|
| upper | Number | 상단 라인 이동 Offset(위 방향으로 +) |
|-------|--------|---------------------------|

| | | |
|-------|--------|----------------------------|
| lower | Number | 하단 라인 이동 Offset(아래 방향으로 +) |
|-------|--------|----------------------------|

| | | |
|------|--------|----------------------------|
| left | Number | 좌측 라인 이동 Offset(좌측 방향으로 +) |
|------|--------|----------------------------|

| | | |
|-------|--------|----------------------------|
| right | Number | 우측 라인 이동 Offset(우측 방향으로 +) |
|-------|--------|----------------------------|

point.resizeBox(width, height) ⇒ [Geometry](#)

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [Point](#)

Overrides: [resizeBox](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|-------|--------|----|
| width | Number | 너비 |
|-------|--------|----|

| | | |
|--------|--------|----|
| height | Number | 높이 |
|--------|--------|----|

point.rotate(angle, origin) ⇒ [Geometry](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [Point](#)

Overrides: [rotate](#)

Returns: [Geometry](#) - 회전된 공간 기하 객체

| Param | Type | Description |
|--------|----------------------------|----------------------|
| angle | Number | 회전 각도 |
| origin | Coordinate | 기준 좌표(default: 중심좌표) |

point.fitToBoundary(envelope) ⇒ [Geometry](#)

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

Kind: instance method of [Point](#)

Overrides: [fitToBoundary](#)

Returns: [Geometry](#) - 적용된 공간 기하 객체

| Param | Type | Description |
|----------|--------------------------|-------------|
| envelope | Envelope | Envelope 영역 |

point.convertCoordinate(coordinate) ⇒ [Coordinate](#)

파라미터가 [x, y] 형식의 좌표 Array 이면 OG.geometry.Coordinate 인스턴스를 new 하여 반환한다.

Kind: instance method of [Point](#)

| Param | Type | Description |
|-------|-------------------------------------|------------------------|
| | Coordinate Array. | [x, y] 형식의 좌표 Array 또는 |

coordinate <Number>

OG.geometry.Coordinate 인스턴스

point.distanceToLine(p, line) ⇒ Number

포인트 P 로부터 라인 AB의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Point](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

point.distanceLineToLine(line1, line2) ⇒ Number

라인1 로부터 라인2 의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Point](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |

point.intersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표들을 반환한다.

Kind: instance method of [Point](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

point.shortestIntersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표중 시작좌표에 가장 가까운 좌표를 반환한다.

Kind: instance method of [Point](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

point.intersectLineToLine(line1, line2, extension) ⇒ [Coordinate](#)

라인1 과 라인2 의 교차점을 계산한다.

Kind: instance method of [Point](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-----------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |
| extension | boolean | 라인을 연장하여 교차점을 계산하는 여부 |

point.intersectCircleToLine(center, radius, from, to) ⇒ [Array.<Coordinate>](#)

주어진 원과 주어진 라인의 교차점을 계산한다.

Kind: instance method of [Point](#)

Returns: [Array.<Coordinate>](#) - 교차점

| Param | Type | Description |
|--------|----------------------------|--------------|
| center | Coordinate | 중심점 |
| radius | Number | 반경 |
| from | Coordinate | line 라인 시작좌표 |
| to | Coordinate | line 라인 끝좌표 |

`point.intersectPointToLine(p, line) ⇒ Coordinate`

포인트 P로부터 라인 AB까지 수직인 가상선을 생각할때, 그 교차점을 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Point](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-------|---|------------------------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

`point.getPercentageDistanceFromPoint(_coordinate) ⇒ Object`

주어진 좌표에 대해 공간기하객체 바운더리 대비 가로,세로 위치 퍼센테이지 비율을 구한다.

Kind: instance method of [Point](#)

Returns: Object - {px , py}

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

`point.isContainsPoint(_coordinate) ⇒ boolean`

공간기하객체가 주어진 좌표를 포함하는지를 반환한다.

Kind: instance method of [Point](#)

Returns: boolean - true,false

| Param | Type | Description |
|--------------------------|----------------------------|-------------|
| <code>_coordinate</code> | Coordinate | 좌표 |

`point.getPointFromPercentageDistance(pXpY) ⇒ Coordinate`

공간기하객체에 대한 퍼센테이지 좌표의 실제 좌표를 구한다.

Kind: instance method of [Point](#)

Returns: [Coordinate](#) - 실 좌표

| Param | Type | Description |
|-------------------|-------|-------------|
| <code>pXpY</code> | Array | 퍼센테이지 좌표 |

`point.getParallelLine(from, to, distance) ⇒ Array.<Coordinate>`

주어진 선분과 일정 거리에 있는 평행한 선분을 반환한다.

Kind: instance method of [Point](#)

Returns: [Array.<Coordinate>](#) - 평행선 시작좌표, 끝좌표 Array

| Param | Type | Description |
|-------------------|----------------------------|-------------|
| <code>from</code> | Coordinate | 라인 시작좌표 |

to [Coordinate](#) 라인 끝좌표
distance

```
point.getParallelPath(line, distance)
```

주어진 라인과 일정 거리에 있는 평행한 라인을 반환한다.

Kind: instance method of [Point](#)

| Param | Type | Description |
|----------|--|-------------|
| line | Array.<Coordinate> | 라인 좌표 Array |
| distance | | |

```
point.reset()
```

저장된 boundary 를 클리어하여 새로 계산하도록 한다.

Kind: instance method of [Point](#)

geometry.PolyLine \Leftarrow [Geometry](#)

Kind: static class of [geometry](#)

Extends: [Geometry](#)

Requires: module:OG.geometry.Coordinate, module:OG.geometry.Envelope,
module:OG.geometry.Geometry

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.PolyLine](#) \Leftarrow [Geometry](#)

- [new OG.geometry.PolyLine\(vertices\)](#)
- [.vertices](#) : [Array.<Coordinate>](#)
- [.TYPE](#) : Number
- [.IS_CLOSED](#) : Boolean
- [.style](#) : [Style](#)
- [.boundary](#) : [Envelope](#)
- [.toString\(\)](#) ⇒ String
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(_geometry\)](#) ⇒ Boolean
- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.moveCentroid\(중심\)](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)

- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

```
new OG.geometry.PolyLine(vertices)
```

PolyLine 공간 기하 객체(Spatial Geometry Object)

| Param | Type | Description |
|----------|---|----------------------|
| vertices | <u>Array.<Coordinate></u> | Line Vertex 좌표 Array |

Example

```
var geom = new OG.geometry.PolyLine([[20, 5], [30, 15], [40, 25], [50, 15]]);
```

polyLine.vertices : [Array.<Coordinate>](#)

Line Vertex 좌표 Array

Kind: instance property of [PolyLine](#)

polyLine.TYPE : Number

공간 기하 객체 타입

Kind: instance property of [PolyLine](#)

Overrides: [TYPE](#)

polyLine.IS_CLOSED : Boolean

닫힌 기하 객체 인지 여부

Kind: instance property of [PolyLine](#)

polyLine.style : [Style](#)

스타일 속성

Kind: instance property of [PolyLine](#)

Overrides: [style](#)

polyLine.boundary : [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역

Kind: instance property of [PolyLine](#)

polyLine.toString() ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [PolyLine](#)

Returns: String - 프라퍼티 정보

polyLine.angleBetweenPoints(prev, next) ⇒ Number

공간기하객체의 두 꼭지점 사이에 가상의 선을 그렸을때, 그 기울기를 구한다.

Kind: instance method of [PolyLine](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

polyLine.isRightAngleBetweenPoints(prev, next) ⇒ Object

공간기하객체의 두 꼭지점 사이의 기울기가 수평또는 수직인지 판별한다.

Kind: instance method of [PolyLine](#)

Returns: Object - {flag : true or false, type: horizontal or vertical or none}

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

`polyLine.angleBetweenThreePoints(prev, next) ⇒ Number`

공간기하객체의 세 꼭지점 사이의 각도 중 작은 각도를 반환한다.

Kind: instance method of [PolyLine](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

`polyLine.isEquals(_geometry) ⇒ Boolean`

주어진 Geometry 객체와 같은지 비교한다.

Kind: instance method of [PolyLine](#)

Returns: Boolean - true:같음, false:다름

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |
|-----------|--------------------------|-------------|

`polyLine.isContains(_geometry) ⇒ Boolean`

주어진 공간기하객체를 포함하는지 비교한다.

Kind: instance method of [PolyLine](#)

Returns: Boolean - 포함하면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`polyLine.isWithin(_geometry) ⇒ Boolean`

주어진 공간기하객체에 포함되는지 비교한다.

Kind: instance method of [PolyLine](#)

Returns: Boolean - 포함되면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`polyLine.getBoundary() ⇒ Envelope`

공간기하객체를 포함하는 사각형의 Boundary 영역을 반환한다.

Kind: instance method of [PolyLine](#)

Returns: [Envelope](#) - Envelope 영역

`polyLine.getCentroid() ⇒ Coordinate`

공간기하객체의 중심좌표를 반환한다.

Kind: instance method of [PolyLine](#)

Returns: [Coordinate](#) - 중심좌표

`polyLine.getVertices()` ⇒ [Array.<Coordinate>](#)

공간기하객체의 모든 꼭지점을 반환한다.

Kind: instance method of [PolyLine](#)

Overrides: [getVertices](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array

`polyLine.minDistance(_coordinate)` ⇒ Number

주어진 좌표와의 최단거리를 반환한다.

Kind: instance method of [PolyLine](#)

Returns: Number - 최단거리

| Param | Type | Description |
|--------------------------|----------------------------|-------------|
| <code>_coordinate</code> | Coordinate | 좌표 |

`polyLine.distance(_geometry)` ⇒ Number

주어진 공간기하객체와의 중심점 간의 거리를 반환한다.

Kind: instance method of [PolyLine](#)

Returns: Number - 거리

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | 공간 기하 객체 |

`polyLine.getLength()` ⇒ Number

공간기하객체의 길이를 반환한다.

Kind: instance method of [PolyLine](#)

Returns: Number - 길이

`polyLine.move(offsetX, offsetY)` ⇒ [Geometry](#)

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance method of [PolyLine](#)

Overrides: [move](#)

Returns: [Geometry](#) - 이동된 공간 기하 객체

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

`polyLine.moveCentroid(중심)`

주어진 중심좌표로 공간기하객체를 이동한다.

Kind: instance method of [PolyLine](#)

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

중심 [Coordinate](#) 좌표

`polyLine.resize(upper, lower, left, right)` ⇒ [Geometry](#)

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance method of [PolyLine](#)

Overrides: [resize](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|--------|----------------------------|
| upper | Number | 상단 라인 이동 Offset(위 방향으로 +) |
| lower | Number | 하단 라인 이동 Offset(아래 방향으로 +) |
| left | Number | 좌측 라인 이동 Offset(좌측 방향으로 +) |
| right | Number | 우측 라인 이동 Offset(우측 방향으로 +) |

`polyLine.resizeBox(width, height)` ⇒ [Geometry](#)

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [PolyLine](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|--------|--------|-------------|
| width | Number | 너비 |
| height | Number | 높이 |

`polyLine.rotate(angle, origin)` ⇒ [Geometry](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [PolyLine](#)

Overrides: [rotate](#)

Returns: [Geometry](#) - 회전된 공간 기하 객체

| Param | Type | Description |
|--------|----------------------------|----------------------|
| angle | Number | 회전 각도 |
| origin | Coordinate | 기준 좌표(default: 중심좌표) |

`polyLine.fitToBoundary(envelope) ⇒ Geometry`

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

Kind: instance method of [PolyLine](#)

Returns: [Geometry](#) - 적용된 공간 기하 객체

| Param | Type | Description |
|----------|--------------------------|-------------|
| envelope | Envelope | Envelope 영역 |

`polyLine.convertCoordinate(coordinate) ⇒ Coordinate`

파라미터가 [x, y] 형식의 좌표 Array 이면 OG.geometry.Coordinate 인스턴스를 new 하여 반환한다.

Kind: instance method of [PolyLine](#)

| Param | Type | Description |
|------------|---|---|
| coordinate | Coordinate Array. <Number> | [x, y] 형식의 좌표 Array 또는 OG.geometry.Coordinate 인스턴스 |

`polyLine.distanceToLine(p, line) ⇒ Number`

포인트 P 로부터 라인 AB의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [PolyLine](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|---------------------|
| p | Coordinate | Array.<Number> 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

polyLine.distanceLineToLine(line1, line2) ⇒ Number

라인1 로부터 라인2 의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [PolyLine](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |

polyLine.intersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표들을 반환한다.

Kind: instance method of [PolyLine](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

polyLine.shortestIntersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표중 시작좌표에 가장 가까운 좌표를 반환한다.

Kind: instance method of [PolyLine](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

polyLine.intersectLineToLine(line1, line2, extension) ⇒ [Coordinate](#)

라인1 과 라인2 의 교차점을 계산한다.

Kind: instance method of [PolyLine](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-----------|--|-----------------------|
| line1 | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |
| extension | boolean | 라인을 연장하여 교차점을 계산하는 여부 |

polyLine.intersectCircleToLine(center, radius, from, to) ⇒ [Array.<Coordinate>](#)

주어진 원과 주어진 라인의 교차점을 계산한다.

Kind: instance method of [PolyLine](#)

Returns: [Array.<Coordinate>](#) - 교차점

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

center [Coordinate](#) 중심점
radius Number 반경
from [Coordinate](#) line 라인 시작좌표
to [Coordinate](#) line 라인 끝좌표

polyLine.intersectPointToLine(p, line) ⇒ [Coordinate](#)

포인트 P 로부터 라인 AB 까지 수직인 가상선을 생각할때, 그 교차점을 계산한다.

Note: NON-ROBUST!

Kind: instance method of [PolyLine](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

polyLine.getPercentageDistanceFromPoint(_coordinate) ⇒ Object

주어진 좌표에 대해 공간기하객체 바운더리 대비 가로,세로 위치 퍼센테이지 비율을 구한다.

Kind: instance method of [PolyLine](#)

Returns: Object - {px , py}

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

polyLine.isContainsPoint(_coordinate) ⇒ boolean

공간기하객체가 주어진 좌표를 포함하는지를 반환한다.

Kind: instance method of [PolyLine](#)

Returns: boolean - true,false

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

polyLine.getPointFromPercentageDistance(pXpY) ⇒ [Coordinate](#)

공간기하객체에 대한 퍼센테이지 좌표의 실제 좌표를 구한다.

Kind: instance method of [PolyLine](#)

Returns: [Coordinate](#) - 실 좌표

| Param | Type | Description |
|-------|-------|-------------|
| pXpY | Array | 퍼센테이지 좌표 |

polyLine.getParallelLine(from, to, distance) ⇒ [Array.<Coordinate>](#)

주어진 선분과 일정 거리에 있는 평행한 선분을 반환한다.

Kind: instance method of [PolyLine](#)

Returns: [Array.<Coordinate>](#) - 평행선 시작좌표, 끝좌표 Array

| Param | Type | Description |
|----------|----------------------------|-------------|
| from | Coordinate | 라인 시작좌표 |
| to | Coordinate | 라인 끝좌표 |
| distance | | |

polyLine.getParallelPath(line, distance)

주어진 라인과 일정 거리에 있는 평행한 라인을 반환한다.

Kind: instance method of [PolyLine](#)

| Param | Type | Description |
|----------|--|-------------|
| line | Array.<Coordinate> | 라인 좌표 Array |
| distance | | |

`polyLine.reset()`

저장된 boundary 를 클리어하여 새로 계산하도록 한다.

Kind: instance method of [PolyLine](#)

`geometry.Polygon` \Leftarrow [PolyLine](#)

Kind: static class of [geometry](#)

Extends: [PolyLine](#)

Requires: module:OG.geometry.Coordinate, module:OG.geometry.Envelope,
module:OG.geometry.Geometry

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Polygon](#) \Leftarrow [PolyLine](#)
 - [new OG.geometry.Polygon\(vertices\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean

- [.style](#) : [Style](#)
- [.boundary](#) : [Envelope](#)
- [.getVertices\(\)](#) ⇒ [Array.<Coordinate>](#)
- [.move\(offsetX, offsetY\)](#) ⇒ [Geometry](#)
- [.resize\(upper, lower, left, right\)](#) ⇒ [Geometry](#)
- [.rotate\(angle, origin\)](#) ⇒ [Geometry](#)
- [.toString\(\)](#) ⇒ String
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(_geometry\)](#) ⇒ Boolean
- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)

- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ [Object](#)
- [.isContainsPoint\(_coordinate\)](#) ⇒ [boolean](#)
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)
- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

```
new OG.geometry.Polygon(vertices)
```

Polygon 공간 기하 객체(Spatial Geometry Object)

| Param | Type | Description |
|----------|--|----------------------|
| vertices | Array.<Coordinate> | Line Vertex 좌표 Array |

Example

```
var geom = new OG.geometry.Polygon([[20, 5], [30, 15], [40, 25], [50, 15], [60, 5], [20, 5]]);
```

```
polygon.vertices : Array.<Coordinate>
```

Line Vertex 좌표 Array

Kind: instance property of [Polygon](#)

`polygon.TYPE` : `Number`

공간 기하 객체 타입

Kind: instance property of [Polygon](#)

Overrides: [TYPE](#)

`polygon.IS_CLOSED` : `Boolean`

닫힌 기하 객체 인지 여부

Kind: instance property of [Polygon](#)

Overrides: [IS_CLOSED](#)

`polygon.style` : [Style](#)

스타일 속성

Kind: instance property of [Polygon](#)

Overrides: [style](#)

`polygon.boundary` : [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역

Kind: instance property of [Polygon](#)

`polygon.getVertices()` ⇒ [Array.<Coordinate>](#)

공간기하객체의 모든 꼭지점을 반환한다.

Kind: instance method of [Polygon](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array

`polygon.move(offsetX, offsetY)` ⇒ [Geometry](#)

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance method of [Polygon](#)

Returns: [Geometry](#) - 이동된 공간 기하 객체

| Param | Type | Description |
|----------------------|--------|-------------|
| <code>offsetX</code> | Number | 가로 Offset |
| <code>offsetY</code> | Number | 세로 Offset |

`polygon.resize(upper, lower, left, right)` ⇒ [Geometry](#)

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance method of [Polygon](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

upper Number 상단 라인 이동 Offset(위 방향으로 +)
lower Number 하단 라인 이동 Offset(아래 방향으로 +)
left Number 좌측 라인 이동 Offset(좌측 방향으로 +)
right Number 우측 라인 이동 Offset(우측 방향으로 +)

polygon.rotate(angle, origin) ⇒ [Geometry](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [Polygon](#)

Returns: [Geometry](#) - 회전된 공간 기하 객체

| Param | Type | Description |
|--------|----------------------------|----------------------|
| angle | Number | 회전 각도 |
| origin | Coordinate | 기준 좌표(default: 중심좌표) |

polygon.toString() ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [Polygon](#)

Returns: String - 프라퍼티 정보

polygon.angleBetweenPoints(prev, next) ⇒ Number

공간기하객체의 두 꼭지점 사이에 가상의 선을 그렸을때, 그 기울기를 구한다.

Kind: instance method of [Polygon](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

`polygon.isRightAngleBetweenPoints(prev, next) ⇒ Object`

공간기하객체의 두 꼭지점 사이의 기울기가 수평또는 수직인지 판별한다.

Kind: instance method of [Polygon](#)

Returns: Object - {flag : true or false, type: horizontal or vertical or none}

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

`polygon.angleBetweenThreePoints(prev, next) ⇒ Number`

공간기하객체의 세 꼭지점 사이의 각도 중 작은 각도를 반환한다.

Kind: instance method of [Polygon](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

`polygon.isEquals(_geometry) ⇒ Boolean`

주어진 Geometry 객체와 같은지 비교한다.

Kind: instance method of [Polygon](#)

Returns: Boolean - true:같음, false:다름

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |

`polygon.isContains(_geometry) ⇒ Boolean`

주어진 공간기하객체를 포함하는지 비교한다.

Kind: instance method of [Polygon](#)

Returns: Boolean - 포함하면 true

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |

`polygon.isWithin(_geometry) ⇒ Boolean`

주어진 공간기하객체에 포함되는지 비교한다.

Kind: instance method of [Polygon](#)

Returns: Boolean - 포함되면 true

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |

`polygon.getBoundary() ⇒ Envelope`

공간기하객체를 포함하는 사각형의 Boundary 영역을 반환한다.

Kind: instance method of [Polygon](#)

Returns: [Envelope](#) - Envelope 영역

`polygon.getCentroid()` ⇒ [Coordinate](#)

공간기하객체의 중심좌표를 반환한다.

Kind: instance method of [Polygon](#)

Returns: [Coordinate](#) - 중심좌표

`polygon.minDistance(_coordinate)` ⇒ Number

주어진 좌표와의 최단거리를 반환한다.

Kind: instance method of [Polygon](#)

Returns: Number - 최단거리

| Param | Type | Description |
|--------------------------|----------------------------|-------------|
| <code>_coordinate</code> | Coordinate | 좌표 |

`polygon.distance(_geometry)` ⇒ Number

주어진 공간기하객체와의 중심점 간의 거리를 반환한다.

Kind: instance method of [Polygon](#)

Returns: Number - 거리

| Param | Type | Description |
|-----------|--------------------------|-------------|
| _geometry | Geometry | 공간 기하 객체 |

`polygon.getLength()` ⇒ [Number](#)

공간기하객체의 길이를 반환한다.

Kind: instance method of [Polygon](#)

Returns: [Number](#) - 길이

`polygon.moveCentroid(중심)`

주어진 중심좌표로 공간기하객체를 이동한다.

Kind: instance method of [Polygon](#)

| Param | Type | Description |
|-------|----------------------------|-------------|
| 중심 | Coordinate | 좌표 |

`polygon.resizeBox(width, height)` ⇒ [Geometry](#)

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [Polygon](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|--------|------------------------|-------------|
| width | Number | 너비 |
| height | Number | 높이 |

`polygon.fitToBoundary(envelope)` ⇒ [Geometry](#)

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

Kind: instance method of [Polygon](#)

Returns: [Geometry](#) - 적용된 공간 기하 객체

| Param | Type | Description |
|----------|--------------------------|-------------|
| envelope | Envelope | Envelope 영역 |

`polygon.convertCoordinate(coordinate)` ⇒ [Coordinate](#)

파라미터가 [x, y] 형식의 좌표 Array 이면 OG.geometry.Coordinate 인스턴스를 new 하여 반환한다.

Kind: instance method of [Polygon](#)

| Param | Type | Description |
|------------|---|---|
| coordinate | Coordinate Array. <Number> | [x, y] 형식의 좌표 Array 또는 OG.geometry.Coordinate 인스턴스 |

`polygon.distanceToLine(p, line)` ⇒ Number

포인트 P 로부터 라인 AB의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Polygon](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

p [Coordinate](#) | Array.<Number> 기준좌표
line [Array.<Coordinate>](#) 라인 시작좌표, 끝좌표 Array

polygon.distanceLineToLine(line1, line2) ⇒ Number

라인1로부터 라인2의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Polygon](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |

polygon.intersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표들을 반환한다.

Kind: instance method of [Polygon](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

polygon.shortestIntersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표중 시작좌표에 가장 가까운 좌표를 반환한다.

Kind: instance method of [Polygon](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

polygon.intersectLineToLine(line1, line2, extension) ⇒ [Coordinate](#)

라인1 과 라인2 의 교차점을 계산한다.

Kind: instance method of [Polygon](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-----------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |
| extension | boolean | 라인을 연장하여 교차점을 계산하는 여부 |

polygon.intersectCircleToLine(center, radius, from, to) ⇒ [Array.<Coordinate>](#)

주어진 원과 주어진 라인의 교차점을 계산한다.

Kind: instance method of [Polygon](#)

Returns: [Array.<Coordinate>](#) - 교차점

| Param | Type | Description |
|--------|----------------------------|--------------|
| center | Coordinate | 중심점 |
| radius | Number | 반경 |
| from | Coordinate | line 라인 시작좌표 |
| to | Coordinate | line 라인 끝좌표 |

polygon.intersectPointToLine(p, line) ⇒ [Coordinate](#)

포인트 P로부터 라인 AB까지 수직인 가상선을 생각할때, 그 교차점을 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Polygon](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-------|--|---------------------|
| p | Coordinate | Array.<Number> 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

`polygon.getPercentageDistanceFromPoint(_coordinate) ⇒ Object`

주어진 좌표에 대해 공간기하객체 바운더리 대비 가로,세로 위치 퍼센테이지 비율을 구한다.

Kind: instance method of [Polygon](#)

Returns: Object - {px , py}

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

`polygon.isContainsPoint(_coordinate) ⇒ boolean`

공간기하객체가 주어진 좌표를 포함하는지를 반환한다.

Kind: instance method of [Polygon](#)

Returns: boolean - true,false

| Param | Type | Description |
|-------------|----------------------------|-------------|
| _coordinate | Coordinate | 좌표 |

`polygon.getPointFromPercentageDistance(pXpY)` ⇒ [Coordinate](#)

공간기하객체에 대한 퍼센테이지 좌표의 실제 좌표를 구한다.

Kind: instance method of [Polygon](#)

Returns: [Coordinate](#) - 실 좌표

Param Type Description

pXpY Array 퍼센테이지 좌표

`polygon.getParallelLine(from, to, distance)` ⇒ [Array.<Coordinate>](#)

주어진 선분과 일정 거리에 있는 평행한 선분을 반환한다.

Kind: instance method of [Polygon](#)

Returns: [Array.<Coordinate>](#) - 평행선 시작좌표, 끝좌표 Array

| Param | Type | Description |
|----------|----------------------------|-------------|
| from | Coordinate | 라인 시작좌표 |
| to | Coordinate | 라인 끝좌표 |
| distance | | |

`polygon.getParallelPath(line, distance)`

주어진 라인과 일정 거리에 있는 평행한 라인을 반환한다.

Kind: instance method of [Polygon](#)

| Param | Type | Description |
|----------|--|-------------|
| line | Array.<Coordinate> | 라인 좌표 Array |
| distance | | |

`polygon.reset()`

저장된 boundary 를 클리어하여 새로 계산하도록 한다.

Kind: instance method of [Polygon](#)

`geometry.Rectangle` \Leftarrow [Polygon](#)

Kind: static class of [geometry](#)

Extends: [Polygon](#)

Requires: module:OG.geometry.Coordinate, module:OG.geometry.Envelope,
module:OG.geometry.Geometry

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Rectangle](#) \Leftarrow [Polygon](#)
 - [new OG.geometry.Rectangle\(upperLeft, width, height\)](#)
 - [.vertices](#) : [Array.<Coordinate>](#)
 - [.TYPE](#) : Number
 - [.IS_CLOSED](#) : Boolean
 - [.style](#) : [Style](#)
 - [.boundary](#) : [Envelope](#)
 - [.getVertices\(\)](#) \Rightarrow [Array.<Coordinate>](#)
 - [.move\(offsetX, offsetY\)](#) \Rightarrow [Geometry](#)
 - [.resize\(upper, lower, left, right\)](#) \Rightarrow [Geometry](#)
 - [.rotate\(angle, origin\)](#) \Rightarrow [Geometry](#)

- [.toString\(\)](#) ⇒ String
- [.angleBetweenPoints\(prev, next\)](#) ⇒ Number
- [.isRightAngleBetweenPoints\(prev, next\)](#) ⇒ Object
- [.angleBetweenThreePoints\(prev, next\)](#) ⇒ Number
- [.isEqual\(_geometry\)](#) ⇒ Boolean
- [.isContains\(_geometry\)](#) ⇒ Boolean
- [.isWithin\(_geometry\)](#) ⇒ Boolean
- [.getBoundary\(\)](#) ⇒ [Envelope](#)
- [.getCentroid\(\)](#) ⇒ [Coordinate](#)
- [.minDistance\(_coordinate\)](#) ⇒ Number
- [.distance\(_geometry\)](#) ⇒ Number
- [.getLength\(\)](#) ⇒ Number
- [.moveCentroid\(중심\)](#)
- [.resizeBox\(width, height\)](#) ⇒ [Geometry](#)
- [.fitToBoundary\(envelope\)](#) ⇒ [Geometry](#)
- [.convertCoordinate\(coordinate\)](#) ⇒ [Coordinate](#)
- [.distanceToLine\(p, line\)](#) ⇒ Number
- [.distanceLineToLine\(line1, line2\)](#) ⇒ Number
- [.intersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.shortestIntersectToLine\(line\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectLineToLine\(line1, line2, extension\)](#) ⇒ [Coordinate](#)
- [.intersectCircleToLine\(center, radius, from, to\)](#) ⇒ [Array.<Coordinate>](#)
- [.intersectPointToLine\(p, line\)](#) ⇒ [Coordinate](#)
- [.getPercentageDistanceFromPoint\(_coordinate\)](#) ⇒ Object
- [.isContainsPoint\(_coordinate\)](#) ⇒ boolean
- [.getPointFromPercentageDistance\(pXpY\)](#) ⇒ [Coordinate](#)

- [.getParallelLine\(from, to, distance\)](#) ⇒ [Array.<Coordinate>](#)
- [.getParallelPath\(line, distance\)](#)
- [.reset\(\)](#)

```
new OG.geometry.Rectangle(upperLeft, width, height)
```

Rectangle 공간 기하 객체(Spatial Geometry Object)

| Param | Type | Description |
|-----------|----------------------------|-------------|
| upperLeft | Coordinate | 좌상단좌표 |
| width | Number | 너비 |
| height | Number | 높이 |

Example

```
var geom = new OG.geometry.Rectangle([20, 5], 10, 10);
```

```
rectangle.vertices : Array.<Coordinate>
```

Line Vertex 좌표 Array

Kind: instance property of [Rectangle](#)

```
rectangle.TYPE : Number
```

공간 기하 객체 타입

Kind: instance property of [Rectangle](#)

Overrides: [TYPE](#)

rectangle.IS_CLOSED : Boolean

닫힌 기하 객체 인지 여부

Kind: instance property of [Rectangle](#)

rectangle.style : [Style](#)

스타일 속성

Kind: instance property of [Rectangle](#)

Overrides: [style](#)

rectangle.boundary : [Envelope](#)

공간기하객체를 포함하는 사각형의 Boundary 영역

Kind: instance property of [Rectangle](#)

rectangle.getVertices() ⇒ [Array.<Coordinate>](#)

공간기하객체의 모든 꼭지점을 반환한다.

Kind: instance method of [Rectangle](#)

Returns: [Array.<Coordinate>](#) - 꼭지점 좌표 Array

`rectangle.move(offsetX, offsetY)` ⇒ [Geometry](#)

가로, 세로 Offset 만큼 좌표를 이동한다.

Kind: instance method of [Rectangle](#)

Returns: [Geometry](#) - 이동된 공간 기하 객체

| Param | Type | Description |
|---------|--------|-------------|
| offsetX | Number | 가로 Offset |
| offsetY | Number | 세로 Offset |

`rectangle.resize(upper, lower, left, right)` ⇒ [Geometry](#)

상, 하, 좌, 우 외곽선을 이동하여 Envelope 을 리사이즈 한다.

Kind: instance method of [Rectangle](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|-------|--------|----------------------------|
| upper | Number | 상단 라인 이동 Offset(위 방향으로 +) |
| lower | Number | 하단 라인 이동 Offset(아래 방향으로 +) |
| left | Number | 좌측 라인 이동 Offset(좌측 방향으로 +) |
| right | Number | 우측 라인 이동 Offset(우측 방향으로 +) |

`rectangle.rotate(angle, origin)` ⇒ [Geometry](#)

기준 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [Rectangle](#)

Returns: [Geometry](#) - 회전된 공간 기하 객체

| Param | Type | Description |
|--------|----------------------------|----------------------|
| angle | Number | 회전 각도 |
| origin | Coordinate | 기준 좌표(default: 중심좌표) |

rectangle.toString() ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [Rectangle](#)

Overrides: [toString](#)

Returns: String - 프라퍼티 정보

rectangle.angleBetweenPoints(prev, next) ⇒ Number

공간기하객체의 두 꼭지점 사이에 가상의 선을 그렸을때, 그 기울기를 구한다.

Kind: instance method of [Rectangle](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|----------------------------|-------------|
| prev | Coordinate | 꼭지점 1 |
| next | Coordinate | 꼭지점 2 |

rectangle.isRightAngleBetweenPoints(prev, next) ⇒ Object

공간기하객체의 두 꼭지점 사이의 기울기가 수평또는 수직인지 판별한다.

Kind: instance method of [Rectangle](#)

Returns: Object - {flag : true or false, type: horizontal or vertical or none}

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

rectangle.angleBetweenThreePoints(prev, next) ⇒ Number

공간기하객체의 세 꼭지점 사이의 각도 중 작은 각도를 반환한다.

Kind: instance method of [Rectangle](#)

Returns: Number - 기울기

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|----------------------------|-------|
| prev | Coordinate | 꼭지점 1 |
|------|----------------------------|-------|

| | | |
|------|----------------------------|-------|
| next | Coordinate | 꼭지점 2 |
|------|----------------------------|-------|

rectangle.isEquals(_geometry) ⇒ Boolean

주어진 Geometry 객체와 같은지 비교한다.

Kind: instance method of [Rectangle](#)

Returns: Boolean - true:같음, false:다름

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|-----------|--------------------------|-------------|
| _geometry | Geometry | Geometry 객체 |
|-----------|--------------------------|-------------|

`rectangle.isContains(_geometry) ⇒ Boolean`

주어진 공간기하객체를 포함하는지 비교한다.

Kind: instance method of [Rectangle](#)

Returns: Boolean - 포함하면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`rectangle.isWithin(_geometry) ⇒ Boolean`

주어진 공간기하객체에 포함되는지 비교한다.

Kind: instance method of [Rectangle](#)

Returns: Boolean - 포함되면 true

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | Geometry 객체 |

`rectangle.getBoundary() ⇒ Envelope`

공간기하객체를 포함하는 사각형의 Boundary 영역을 반환한다.

Kind: instance method of [Rectangle](#)

Returns: [Envelope](#) - Envelope 영역

`rectangle.getCentroid() ⇒ Coordinate`

공간기하객체의 중심좌표를 반환한다.

Kind: instance method of [Rectangle](#)

Returns: [Coordinate](#) - 중심좌표

`rectangle.minDistance(_coordinate) ⇒ Number`

주어진 좌표와의 최단거리를 반환한다.

Kind: instance method of [Rectangle](#)

Returns: Number - 최단거리

| Param | Type | Description |
|--------------------------|----------------------------|-------------|
| <code>_coordinate</code> | Coordinate | 좌표 |

`rectangle.distance(_geometry) ⇒ Number`

주어진 공간기하객체와의 중심점 간의 거리를 반환한다.

Kind: instance method of [Rectangle](#)

Returns: Number - 거리

| Param | Type | Description |
|------------------------|--------------------------|-------------|
| <code>_geometry</code> | Geometry | 공간 기하 객체 |

`rectangle.getLength() ⇒ Number`

공간기하객체의 길이를 반환한다.

Kind: instance method of [Rectangle](#)

Returns: Number - 길이

`rectangle.moveCentroid(중심)`

주어진 중심좌표로 공간기하객체를 이동한다.

Kind: instance method of [Rectangle](#)

| Param | Type | Description |
|-------|----------------------------|-------------|
| 중심 | Coordinate | 좌표 |

`rectangle.resizeBox(width, height) ⇒ Geometry`

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [Rectangle](#)

Returns: [Geometry](#) - 리사이즈된 공간 기하 객체

| Param | Type | Description |
|--------|--------|-------------|
| width | Number | 너비 |
| height | Number | 높이 |

`rectangle.fitToBoundary(envelope) ⇒ Geometry`

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

Kind: instance method of [Rectangle](#)

Returns: [Geometry](#) - 적용된 공간 기하 객체

| Param | Type | Description |
|----------|--------------------------|-------------|
| envelope | Envelope | Envelope 영역 |

rectangle.convertCoordinate(coordinate) ⇒ [Coordinate](#)

파라미터가 [x, y] 형식의 좌표 Array 이면 OG.geometry.Coordinate 인스턴스를 new 하여 반환한다.

Kind: instance method of [Rectangle](#)

| Param | Type | Description |
|------------|---|---|
| coordinate | Coordinate Array. <Number> | [x, y] 형식의 좌표 Array 또는 OG.geometry.Coordinate 인스턴스 |

rectangle.distanceToLine(p, line) ⇒ Number

포인트 P 로부터 라인 AB의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Rectangle](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|---|--------------------|
| p | Coordinate Array.<Number> | 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

rectangle.distanceLineToLine(line1, line2) ⇒ Number

라인1 로부터 라인2 의 거리를 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Rectangle](#)

Returns: Number - 거리

| Param | Type | Description |
|-------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |

rectangle.intersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표들을 반환한다.

Kind: instance method of [Rectangle](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

rectangle.shortestIntersectToLine(line) ⇒ [Array.<Coordinate>](#)

기하도형이 주어진 라인과 교차하는 좌표중 시작좌표에 가장 가까운 좌표를 반환한다.

Kind: instance method of [Rectangle](#)

| Param | Type | Description |
|-------|--|--------------------|
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

rectangle.intersectLineToLine(line1, line2, extension) ⇒ [Coordinate](#)

라인1 과 라인2 의 교차점을 계산한다.

Kind: instance method of [Rectangle](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-----------|--|--------------------------|
| line1 | Array.<Coordinate> | line1 라인 시작좌표, 끝좌표 Array |
| line2 | Array.<Coordinate> | line2 라인 시작좌표, 끝좌표 Array |
| extension | boolean | 라인을 연장하여 교차점을 계산하는 여부 |

rectangle.intersectCircleToLine(center, radius, from, to) ⇒ [Array.<Coordinate>](#)

주어진 원과 주어진 라인의 교차점을 계산한다.

Kind: instance method of [Rectangle](#)

Returns: [Array.<Coordinate>](#) - 교차점

| Param | Type | Description |
|--------|----------------------------|--------------|
| center | Coordinate | 중심점 |
| radius | Number | 반경 |
| from | Coordinate | line 라인 시작좌표 |
| to | Coordinate | line 라인 끝좌표 |

rectangle.intersectPointToLine(p, line) ⇒ [Coordinate](#)

포인트 P로부터 라인 AB까지 수직인 가상선을 생각할때, 그 교차점을 계산한다.

Note: NON-ROBUST!

Kind: instance method of [Rectangle](#)

Returns: [Coordinate](#) - 교차점

| Param | Type | Description |
|-------|--|---------------------|
| p | Coordinate | Array.<Number> 기준좌표 |
| line | Array.<Coordinate> | 라인 시작좌표, 끝좌표 Array |

`rectangle.getPercentageDistanceFromPoint(_coordinate) ⇒ Object`

주어진 좌표에 대해 공간기하객체 바운더리 대비 가로,세로 위치 퍼센테이지 비율을 구한다.

Kind: instance method of [Rectangle](#)

Returns: Object - {px , py}

| Param | Type | Description |
|--------------------------|----------------------------|-------------|
| <code>_coordinate</code> | Coordinate | 좌표 |

`rectangle.isContainsPoint(_coordinate) ⇒ boolean`

공간기하객체가 주어진 좌표를 포함하는지를 반환한다.

Kind: instance method of [Rectangle](#)

Returns: boolean - true,false

| Param | Type | Description |
|--------------------------|----------------------------|-------------|
| <code>_coordinate</code> | Coordinate | 좌표 |

`rectangle.getPointFromPercentageDistance(pXpY) ⇒ Coordinate`

공간기하객체에 대한 퍼센테이지 좌표의 실제 좌표를 구한다.

Kind: instance method of [Rectangle](#)

Returns: [Coordinate](#) - 실 좌표

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

pXpY Array 퍼센테이지 좌표

rectangle.getParallelLine(from, to, distance) ⇒ [Array.<Coordinate>](#)

주어진 선분과 일정 거리에 있는 평행한 선분을 반환한다.

Kind: instance method of [Rectangle](#)

Returns: [Array.<Coordinate>](#) - 평행선 시작좌표, 끝좌표 Array

| Param | Type | Description |
|----------|----------------------------|-------------|
| from | Coordinate | 라인 시작좌표 |
| to | Coordinate | 라인 끝좌표 |
| distance | | |

rectangle.getParallelPath(line, distance)

주어진 라인과 일정 거리에 있는 평행한 라인을 반환한다.

Kind: instance method of [Rectangle](#)

| Param | Type | Description |
|----------|--|-------------|
| line | Array.<Coordinate> | 라인 좌표 Array |
| distance | | |

rectangle.reset()

저장된 boundary 를 클리어하여 새로 계산하도록 한다.

Kind: instance method of [Rectangle](#)

geometry.Style \Leftarrow [HashMap](#)

Kind: static class of [geometry](#)

Extends: [HashMap](#)

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Style](#) \Leftarrow [HashMap](#)
 - [new OG.geometry.Style\(style\)](#)
 - [.map](#) : Object
 - [.put\(key, value\)](#)
 - [.get\(key\)](#) \Rightarrow Object
 - [.containsKey\(key\)](#) \Rightarrow Boolean
 - [.containsValue\(value\)](#) \Rightarrow Boolean
 - [.isEmpty\(\)](#) \Rightarrow Boolean
 - [.clear\(\)](#)
 - [.remove\(key\)](#)
 - [.keys\(\)](#) \Rightarrow Array.<String>
 - [.values\(\)](#) \Rightarrow Array.<Object>
 - [.size\(\)](#) \Rightarrow Number
 - [.toString\(\)](#) \Rightarrow String

new OG.geometry.Style(style)

스타일(StyleSheet) Property 정보 클래스

| Param | Type | Description |
|-------------|---------------------|-------------|
| styleObject | 키:값 매핑된 스타일 프라퍼티 정보 | |

Example

```
var style = new OG.geometry.Style({
  'cursor': 'default',
  'stroke': 'black'
});
```

style.map : Object

key:value 매핑 JSON 오브젝트

Kind: instance property of [Style](#)

style.put(key, value)

key : value 를 매핑한다.

Kind: instance method of [Style](#)

| Param | Type | Description |
|-------|--------|-------------|
| key | String | 키 |
| value | Object | 값 |

style.get(key) ⇒ Object

key 에 대한 value 를 반환한다.

Kind: instance method of [Style](#)

Returns: Object - 값

Param Type Description

key String키

`style.containsKey(key) ⇒ Boolean`

주어진 key 를 포함하는지 여부를 반환한다.

Kind: instance method of [Style](#)

Param Type Description

key String키

`style.containsValue(value) ⇒ Boolean`

주어진 value 를 포함하는지 여부를 반환한다.

Kind: instance method of [Style](#)

Param Type Description

value Object값

`style.isEmpty() ⇒ Boolean`

Empty 여부를 반환한다.

Kind: instance method of [Style](#)

`style.clear()`

매핑정보를 클리어한다.

Kind: instance method of [Style](#)

`style.remove(key)`

주어진 `key` 의 매핑정보를 삭제한다.

Kind: instance method of [Style](#)

| Param | Type | Description |
|-------|--------|-------------|
| key | String | 키 |

`style.keys()` ⇒ `Array.<String>`

key 목록을 반환한다.

Kind: instance method of [Style](#)

Returns: `Array.<String>` - 키목록

`style.values()` ⇒ `Array.<Object>`

value 목록을 반환한다.

Kind: instance method of [Style](#)

Returns: Array.<Object> - 값목록

style.size() ⇒ Number

매핑된 key:value 갯수를 반환한다.

Kind: instance method of [Style](#)

style.toString() ⇒ String

객체 프라퍼티 정보를 JSON 스트링으로 반환한다.

Kind: instance method of [Style](#)

Returns: String - 프라퍼티 정보

OG.graph : object

Kind: static namespace of [OG](#)

- [.graph](#) : object
 - [.Canvas](#)
 - [new OG.graph.Canvas\(container, containerSize, backgroundColor, backgroundImage\)](#)
 - [.initConfig\(config\)](#)
 - [.getRenderer\(\)](#) ⇒ OG.RaphaelRenderer

- [.getContainer\(\)](#) ⇒ HTMLElement
- [.getEventHandler\(\)](#) ⇒ OG.EventHandler
- [.addSlider\(\)](#)
- [.removeSlider\(\)](#)
- [.drawShape\(position, shape, size, style, id, parentId, preventEvent\)](#) ⇒ Element
- [.drawTransformer\(position, label, inputs, outputs, id\)](#) ⇒ Element
- [.setShapeStyle\(shapeElement, style\)](#)
- [.setTextListInController\(shapeElement, textList\)](#)
- [.getTextListInController\(shapeElement\)](#)
- [.drawLabel\(shapeElement, text, style\)](#) ⇒ Element
- [.redrawConnectedEdge\(element\)](#)
- [.reconnect\(edge\)](#) ⇒ Element
- [.connect\(fromElement, toElement, style, label, fromP, toP, preventTrigger, id, edgeShape\)](#) ⇒ * | Element
- [.connectWithTerminalId\(fromTerminal, toTerminal, style, label\)](#) ⇒ String | String | [geometry](#)
- [.disconnect\(element\)](#)
- [.group\(elements\)](#) ⇒ Element
- [.ungroup\(groupElements\)](#) ⇒ Array.<Element>
- [.addToGroup\(groupElement, elements\)](#)
- [.collapse\(element\)](#)
- [.expand\(element\)](#)
- [.clear\(\)](#)
- [.removeShape\(element\)](#)
- [.removeChild\(element\)](#)
- [.removeGuide\(element\)](#)

- [.removeAllGuide\(\)](#)
- [.getRootElement\(\)](#) ⇒ Element
- [.getRootGroup\(\)](#) ⇒ Element
- [.getElementByPoint\(position\)](#) ⇒ Element
- [.getElementsByBBox\(envelope\)](#) ⇒ Array.<Element>
- [.setAttr\(element, attribute\)](#)
- [.getAttr\(element, attrName\)](#) ⇒ Object
- [.toFront\(element\)](#)
- [.toBack\(element\)](#)
- [.bringForward\(element\)](#)
- [.sendBackward\(element\)](#)
- [.getCanvasSize\(\)](#) ⇒ Array.<Number>
- [.setCanvasSize\(size\)](#)
- [.fitCanvasSize\(minSize, fitScale\)](#)
- [.setViewBox\(position, size, isFit\)](#)
- [.getScale\(\)](#) ⇒ Number
- [.setScale\(scale\)](#)
- [.show\(element\)](#)
- [.hide\(element\)](#)
- [.appendChild\(srcElement, targetElement\)](#) ⇒ Element
- [.insertAfter\(srcElement, targetElement\)](#) ⇒ Element
- [.insertBefore\(srcElement, targetElement\)](#) ⇒ Element
- [.move\(element, offset\)](#) ⇒ Element
- [.moveCentroid\(element, position\)](#) ⇒ Element
- [.rotate\(element, angle\)](#) ⇒ Element
- [.resize\(element, offset\)](#) ⇒ Element

- [.resizeBox\(element, size\)](#) ⇒ Element
- [.clone\(element\)](#) ⇒ Element
- [.getBoundary\(element\)](#) ⇒ [Envelope](#)
- [.getElementById\(id\)](#) ⇒ Element
- [.getElementsByType\(shapeType, excludeType\)](#) ⇒ Array.<Element>
- [.getElementsByShapeId\(shapeId\)](#) ⇒ Array.<Element>
- [.getRelatedElementsFromEdge\(edgeElement\)](#) ⇒ Object
- [.getParent\(Element\)](#) ⇒ Element
- [.getChilids\(element\)](#) ⇒ Array
- [.getAllShapes\(\)](#) ⇒ Array
- [.getAllEdges\(\)](#) ⇒ Array
- [.getBBox\(element\)](#) ⇒ Object
- [.getRootBBox\(\)](#) ⇒ Object
- [.getRealRootBBox\(\)](#) ⇒ Object
- [.isSVG\(\)](#) ⇒ Boolean
- [.isVML\(\)](#) ⇒ Boolean
- [.setCustomData\(shapeElement, data\)](#)
- [.getCustomData\(shapeElement\)](#) ⇒ Object
- [.setExtCustomData\(shapeElement, data\)](#)
- [.getExtCustomData\(shapeElement\)](#) ⇒ Object
- [.toXML\(\)](#) ⇒ String
- [.toJSON\(\)](#) ⇒ Object
- [.loadXML\(xml\)](#) ⇒ Object
- [.loadJSON\(json\)](#) ⇒ Object
- [.undo\(\)](#)

- [.redo\(\)](#)
- [.getPrevEdges\(element\)](#) ⇒ Array.<Element>
- [.getNextEdges\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapes\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapeIds\(element\)](#) ⇒ Array.<String>
- [.getNextShapes\(element\)](#) ⇒ Array.<Element>
- [.getNextShapeIds\(element\)](#) ⇒ Array.<String>
- [.onDrawShape\(callbackFunc\)](#)
- [.onUndo\(callbackFunc\)](#)
- [.onRedo\(callbackFunc\)](#)
- [.onDivideLane\(callbackFunc\)](#)
- [.onDrawLabel\(callbackFunc\)](#)
- [.onLabelChanged\(callbackFunc\)](#)
- [.onBeforeLabelChange\(callbackFunc\)](#)
- [.onRedrawShape\(callbackFunc\)](#)
- [.onRemoveShape\(callbackFunc\)](#)
- [.onRotateShape\(callbackFunc\)](#)
- [.onMoveShape\(callbackFunc\)](#)
- [.onResizeShape\(callbackFunc\)](#)
- [.onBeforeConnectShape\(callbackFunc\)](#)
- [.onBeforeRemoveShape\(callbackFunc\)](#)
- [.onConnectShape\(callbackFunc\)](#)
- [.onDisconnectShape\(callbackFunc\)](#)
- [.onGroup\(callbackFunc\)](#)
- [.onUnGroup\(callbackFunc\)](#)
- [.onCollapsed\(callbackFunc\)](#)

- [.onExpanded\(callbackFunc\)](#)

graph.Canvas

Kind: static class of [graph](#)

Requires: module:OG.common.*, module:OG.geometry.*, module:OG.shape.*, module:OG.renderer.*, module:OG.handler.*, module:OG.layout.*, module:raphael-2.1.0

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.Canvas](#)
 - [new OG.graph.Canvas\(container, containerSize, backgroundColor, backgroundImage\)](#)
 - [.initConfig\(config\)](#)
 - [.getRenderer\(\)](#) ⇒ OG.RaphaelRenderer
 - [.getContainer\(\)](#) ⇒ HTMLElement
 - [.getEventHandler\(\)](#) ⇒ OG.EventHandler
 - [.addSlider\(\)](#)
 - [.removeSlider\(\)](#)
 - [.drawShape\(position, shape, size, style, id, parentId, preventEvent\)](#) ⇒ Element
 - [.drawTransformer\(position, label, inputs, outputs, id\)](#) ⇒ Element
 - [.setShapeStyle\(shapeElement, style\)](#)
 - [.setTextListInController\(shapeElement, textList\)](#)
 - [.getTextListInController\(shapeElement\)](#)

- [.drawLabel\(shapeElement, text, style\)](#) ⇒ Element
- [.redrawConnectedEdge\(element\)](#)
- [.reconnect\(edge\)](#) ⇒ Element
- [.connect\(fromElement, toElement, style, label, fromP, toP, preventTrigger, id, edgeShape\)](#) ⇒ * | Element
- [.connectWithTerminalId\(fromTerminal, toTerminal, style, label\)](#) ⇒ String | String | [geometry](#)
- [.disconnect\(element\)](#)
- [.group\(elements\)](#) ⇒ Element
- [.ungroup\(groupElements\)](#) ⇒ Array.<Element>
- [.addToGroup\(groupElement, elements\)](#)
- [.collapse\(element\)](#)
- [.expand\(element\)](#)
- [.clear\(\)](#)
- [.removeShape\(element\)](#)
- [.removeChild\(element\)](#)
- [.removeGuide\(element\)](#)
- [.removeAllGuide\(\)](#)
- [.getRootElement\(\)](#) ⇒ Element
- [.getRootGroup\(\)](#) ⇒ Element
- [.getElementByPoint\(position\)](#) ⇒ Element
- [.getElementsByBBox\(envelope\)](#) ⇒ Array.<Element>
- [.setAttr\(element, attribute\)](#)
- [.getAttr\(element, attrName\)](#) ⇒ Object
- [.toFront\(element\)](#)
- [.toBack\(element\)](#)
- [.bringForward\(element\)](#)

- [.sendBackward\(element\)](#)
- [.getCanvasSize\(\)](#) ⇒ Array.<Number>
- [.setCanvasSize\(size\)](#)
- [.fitCanvasSize\(minSize, fitScale\)](#)
- [.setViewBox\(position, size, isFit\)](#)
- [.getScale\(\)](#) ⇒ Number
- [.setScale\(scale\)](#)
- [.show\(element\)](#)
- [.hide\(element\)](#)
- [.appendChild\(srcElement, targetElement\)](#) ⇒ Element
- [.insertAfter\(srcElement, targetElement\)](#) ⇒ Element
- [.insertBefore\(srcElement, targetElement\)](#) ⇒ Element
- [.move\(element, offset\)](#) ⇒ Element
- [.moveCentroid\(element, position\)](#) ⇒ Element
- [.rotate\(element, angle\)](#) ⇒ Element
- [.resize\(element, offset\)](#) ⇒ Element
- [.resizeBox\(element, size\)](#) ⇒ Element
- [.clone\(element\)](#) ⇒ Element
- [.getBoundary\(element\)](#) ⇒ [Envelope](#)
- [.getElementById\(id\)](#) ⇒ Element
- [.getElementsByType\(shapeType, excludeType\)](#) ⇒ Array.<Element>
- [.getElementsByShapeId\(shapeId\)](#) ⇒ Array.<Element>
- [.getRelatedElementsFromEdge\(edgeElement\)](#) ⇒ Object
- [.getParent\(Element\)](#) ⇒ Element
- [.getChilts\(element\)](#) ⇒ Array
- [.getAllShapes\(\)](#) ⇒ Array

- [.getAllEdges\(\)](#) ⇒ Array
- [.getBBox\(element\)](#) ⇒ Object
- [.getRootBBox\(\)](#) ⇒ Object
- [.getRealRootBBox\(\)](#) ⇒ Object
- [.isSVG\(\)](#) ⇒ Boolean
- [.isVML\(\)](#) ⇒ Boolean
- [.setCustomData\(shapeElement, data\)](#)
- [.getCustomData\(shapeElement\)](#) ⇒ Object
- [.setExtCustomData\(shapeElement, data\)](#)
- [.getExtCustomData\(shapeElement\)](#) ⇒ Object
- [.toXML\(\)](#) ⇒ String
- [.toJSON\(\)](#) ⇒ Object
- [.loadXML\(xml\)](#) ⇒ Object
- [.loadJSON\(json\)](#) ⇒ Object
- [.undo\(\)](#)
- [.redo\(\)](#)
- [.getPrevEdges\(element\)](#) ⇒ Array.<Element>
- [.getNextEdges\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapes\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapeIds\(element\)](#) ⇒ Array.<String>
- [.getNextShapes\(element\)](#) ⇒ Array.<Element>
- [.getNextShapeIds\(element\)](#) ⇒ Array.<String>
- [.onDrawShape\(callbackFunc\)](#)
- [.onUndo\(callbackFunc\)](#)
- [.onRedo\(callbackFunc\)](#)
- [.onDivideLane\(callbackFunc\)](#)

- [.onDrawLabel\(callbackFunc\)](#)
- [.onLabelChanged\(callbackFunc\)](#)
- [.onBeforeLabelChange\(callbackFunc\)](#)
- [.onRedrawShape\(callbackFunc\)](#)
- [.onRemoveShape\(callbackFunc\)](#)
- [.onRotateShape\(callbackFunc\)](#)
- [.onMoveShape\(callbackFunc\)](#)
- [.onResizeShape\(callbackFunc\)](#)
- [.onBeforeConnectShape\(callbackFunc\)](#)
- [.onBeforeRemoveShape\(callbackFunc\)](#)
- [.onConnectShape\(callbackFunc\)](#)
- [.onDisconnectShape\(callbackFunc\)](#)
- [.onGroup\(callbackFunc\)](#)
- [.onUnGroup\(callbackFunc\)](#)
- [.onCollapsed\(callbackFunc\)](#)
- [.onExpanded\(callbackFunc\)](#)

new OG.graph.Canvas(container, containerSize, backgroundColor, backgroundImage)

OpenGraph 캔버스 클래스

| Param | Type | Description |
|-----------------|----------------------|------------------------|
| container | HTMLElement String | 컨테이너 DOM element or ID |
| containerSize | Array.<Number> | 컨테이너 Width, Height |
| backgroundColor | String | 캔버스 배경색 |
| backgroundImage | String | 캔버스 배경이미지 |

Example

```
var canvas = new OG.Canvas('canvas', [1000, 800], 'white',  
'url(./images/grid.gif)');  
  
var circleShape = canvas.drawShape([100, 100], new OG.CircleShape(), [100,  
100]);  
var ellipseShape = canvas.drawShape([300, 200], new OG.EllipseShape('label'),  
[100, 50]);  
  
var edge = canvas.connect(circleShape, ellipseShape);
```

canvas.initConfig(config)

Canvas 의 설정값을 초기화한다.

- selectable : 클릭선택 가능여부(디폴트 true)
- dragSelectable : 마우스드래그선택 가능여부(디폴트 true)
- movable : 이동 가능여부(디폴트 true)
- resizable : 리사이즈 가능여부(디폴트 true)
- connectable : 연결 가능여부(디폴트 true)
- selfConnectable : Self 연결 가능여부(디폴트 true)
- connectCloneable : 드래그하여 연결시 대상 없을 경우 자동으로 Shape 복사하여 연결 처리 여부(디폴트 true)
- connectRequired : 드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부(디폴트 true)
- labelEditable : 라벨 수정여부(디폴트 true)
- groupDropable : 그룹핑 가능여부(디폴트 true)
- enableHotKey : 핫키 가능여부(디폴트 true)
- enableContextMenu : 마우스 우클릭 메뉴 가능여부(디폴트 true)
- autoExtensional : 캔버스 자동 확장 기능(디폴트 true)
- useSlider : 확대축소 슬라이더 사용 여부
- stickGuide : 스틱 가이드 표시 여부
- checkBridgeEdge : 연결된 두 오브젝트의 소속에 따른 연결선 스타일 변화 여부

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|------|-------------------|
| configObject | JSON | 포맷의 configuration |

`canvas.getRenderer()` ⇒ `OG.RaphaelRenderer`

랜더러를 반환한다.

Kind: instance method of [Canvas](#)

`canvas.getContainer()` ⇒ `HTMLElement`

컨테이너 DOM element 를 반환한다.

Kind: instance method of [Canvas](#)

`canvas.getEventHandler()` ⇒ `OG.EventHandler`

이벤트 핸들러를 반환한다.

Kind: instance method of [Canvas](#)

`canvas.addSlider()`

확대 축소 슬라이더를 설치한다.

Kind: instance method of [Canvas](#)

`canvas.removeSlider()`

확대 축소 슬라이더를 삭제한다.

Kind: instance method of [Canvas](#)

`canvas.drawShape(position, shape, size, style, id, parentId, preventEvent) ⇒ Element`

Shape 을 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [Canvas](#)

Returns: Element - Group DOM Element with geometry

| Param | Type | Description |
|--------------|--------------------------------|-----------------------------|
| position | Array.<Number> | 드로잉할 위치 좌표(중앙 기준) |
| shape | IShape | Shape |
| size | Array.<Number> | Shape Width, Height |
| style | Style Object | 스타일 (Optional) |
| id | String | Element ID 지정 (Optional) |
| parentId | String | 부모 Element ID 지정 (Optional) |
| preventEvent | Boolean | 이벤트 생성 방지 |

Example

```
canvas.drawShape([100, 100], new OG.CircleShape(), [50, 50], {stroke:'red'});
```

`canvas.drawTransformer(position, label, inputs, outputs, id) ⇒ Element`

Transformer Shape 을 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [Canvas](#)

Returns: Element - Group DOM Element with geometry

| Param | Type | Description |
|----------|----------------|--------------------------|
| position | Array.<Number> | 드로잉할 위치 좌표(중앙 기준) |
| label | String | Label |
| inputs | Array.<String> | 인풋에 위치할 리스트 |
| outputs | Array.<String> | 아웃풋에 위치할 리스트 |
| id | String | Element ID 지정 (Optional) |

Example

```
canvas.drawTransformer([100, 100], 'label' ['str1','str2'],['out']);
```

```
canvas.setShapeStyle(shapeElement, style)
```

Shape 의 스타일을 변경한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|---------|-------------------|
| shapeElement | Element | Shape DOM element |
| style | Object | 스타일 |

```
canvas.setTextListInController(shapeElement, textList)
```

Shape 의 선 연결 커스텀 컨트롤러를 설정한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|---------|-------------------|
| shapeElement | Element | Shape DOM element |
| textList | Array | 텍스트 리스트 |

```
canvas.getTextListInController(shapeElement)
```

Shape 의 선 연결 커스텀 컨트롤러를 가져온다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|---------|-------------------|
| shapeElement | Element | Shape DOM element |

```
canvas.drawLabel(shapeElement, text, style) ⇒ Element
```

Shape 의 Label 을 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [Canvas](#)

Returns: Element - DOM Element

| Param | Type | Description |
|--------------|--------------------------------|-------------------------|
| shapeElement | Element String | Shape DOM element or ID |
| text | String | 텍스트 |
| style | Style Object | 스타일 |

```
canvas.redrawConnectedEdge(element)
```

Shape 의 연결된 Edge 를 redraw 한다.(이동 또는 리사이즈시)

Kind: instance method of [Canvas](#)

| Param | Type |
|---------|---------|
| element | Element |

```
canvas.reconnect(edge) ⇒ Element
```

연결된 터미널의 vertices 를 초기화한다.

Kind: instance method of [Canvas](#)

Returns: Element - 연결된 Edge 엘리먼트

| Param | Type | Description |
|-------|---------|-------------|
| edge | Element | Edge Shape |

```
canvas.connect(fromElement, toElement, style, label, fromP, toP, preventTrigger, id, edgeShape) ⇒ *  
| Element
```

두개의 Shape 을 Edge 로 연결한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|----------------|--------------------------------|---|
| fromElement | Element | from Shape Element |
| toElement | Element | to Shape Element |
| style | Style Object | 스타일 |
| label | String | Label |
| fromP | | fromElement 와 연결될 터미널 좌표 x,y (optional) |
| toP | | toElement 와 연결될 터미널 좌표 x,y (optional) |
| preventTrigger | | 참 일 경우 이벤트 발생을 방지 |
| id | | 연결선의 아이디 |
| edgeShape | Element | 이 값이 없으면 신규 OG.EdgeShape 를 생성 |

```
canvas.connectWithTerminalId(fromTerminal, toTerminal, style, label) ⇒ String | String | geometry
```

두개의 터미널 아이디로 부터 얻어진 Shape를 Edge 로 연결한다.

Kind: instance method of [Canvas](#)

Returns: String - id 부여 할 아이디String - shapeId shapeId[geometry](#) - geom Edge

geometry

| Param | Type | Description |
|--------------|--------------------------------|------------------|
| fromTerminal | String | from Terminal Id |
| toTerminal | String | to Terminal Id |
| style | Style Object | 스타일 |
| label | String | Label |

`canvas.disconnect(element)`

연결속성정보를 삭제한다. Edge 인 경우는 라인만 삭제하고, 일반 Shape 인 경우는 연결된 모든 Edge 를 삭제한다.

Kind: instance method of [Canvas](#)

| Param | Type |
|---------|---------|
| element | Element |

`canvas.group(elements) ⇒ Element`

주어진 Shape 들을 그룹핑한다.

Kind: instance method of [Canvas](#)

Returns: Element - Group Shape Element

| Param | Type |
|----------|-----------------|
| elements | Array.<Element> |

`canvas.ungroup(groupElements) ⇒ Array.<Element>`

주어진 그룹들을 그룹해제한다.

Kind: instance method of [Canvas](#)

Returns: Array.<Element> - ungrouped Elements

| Param | Type |
|---------------|-----------------|
| groupElements | Array.<Element> |

```
canvas.addToGroup(groupElement, elements)
```

주어진 Shape 들을 그룹에 추가한다.

Kind: instance method of [Canvas](#)

| Param | Type |
|--------------|-----------------|
| groupElement | Element |
| elements | Array.<Element> |

```
canvas.collapse(element)
```

주어진 Shape 이 그룹인 경우 collapse 한다.

Kind: instance method of [Canvas](#)

| Param | Type |
|---------|---------|
| element | Element |

```
canvas.expand(element)
```

주어진 Shape 이 그룹인 경우 expand 한다.

Kind: instance method of [Canvas](#)

| Param | Type |
|---------|---------|
| element | Element |

`canvas.clear()`

드로잉된 모든 오브젝트를 클리어한다.

Kind: instance method of [Canvas](#)

`canvas.removeShape(element)`

Shape 을 캔버스에서 관련된 모두를 삭제한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`canvas.removeChild(element)`

하위 엘리먼트만 제거한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`canvas.removeGuide(element)`

ID에 해당하는 Element 의 Move & Resize 용 가이드를 제거한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`canvas.removeAllGuide()`

모든 Move & Resize 용 가이드를 제거한다.

Kind: instance method of [Canvas](#)

`canvas.getRootElement()` ⇒ Element

랜더러 캔버스 Root Element 를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Element - Element

`canvas.getRootGroup()` ⇒ Element

랜더러 캔버스 Root Group Element 를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Element - Element

`canvas.getElementByPoint(position) ⇒ Element`

주어진 지점을 포함하는 Top Element 를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Element - Element

| Param | Type | Description |
|----------|----------------|-------------|
| position | Array.<Number> | 위치 좌표 |

`canvas.getElementsByBBox(envelope) ⇒ Array.<Element>`

주어진 Boundary Box 영역에 포함되는 Shape(GEOM, TEXT, IMAGE) Element 를 반환한다.

모든 vertices를 포함한 엘리먼트를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Array.<Element> - Element

| Param | Type | Description |
|----------|--------------------------|-----------------|
| envelope | Envelope | Boundary Box 영역 |

`canvas.setAttr(element, attribute)`

엘리먼트에 속성값을 설정한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

attributeObject 속성값

canvas.getAttr(element, attrName) ⇒ Object

엘리먼트 속성값을 반환한다.

Kind: instance method of [Canvas](#)

Returns: Object - attribute 속성값

| Param | Type | Description |
|----------|------------------|---------------|
| element | Element String | Element 또는 ID |
| attrName | String | 속성이름 |

canvas.toFront(element)

ID에 해당하는 Element 를 최상단 레이어로 이동한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

canvas.toBack(element)

ID에 해당하는 Element 를 최하단 레이어로 이동한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`canvas.bringForward(element)`

ID에 해당하는 Element 를 앞으로 한단계 이동한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`canvas.sendBackward(element)`

ID에 해당하는 Element 를 뒤로 한단계 이동한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`canvas.getCanvasSize()` ⇒ Array.<Number>

랜더러 캔버스의 사이즈(Width, Height)를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Array.<Number> - Canvas Width, Height

`canvas.setCanvasSize(size)`

랜더러 캔버스의 사이즈(Width, Height)를 변경한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|-------|----------------|----------------------|
| size | Array.<Number> | Canvas Width, Height |

```
canvas.fitCanvasSize(minSize, fitScale)
```

랜더러 캔버스의 사이즈(Width, Height)를 실제 존재하는 Shape 의 영역에 맞게 변경한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|----------|----------------|--|
| minSize | Array.<Number> | Canvas 최소 Width, Height |
| fitScale | Boolean | 주어진 minSize 에 맞게 fit 여부(Default:false) |

```
canvas.setViewBox(position, size, isFit)
```

새로운 View Box 영역을 설정한다. (ZoomIn & ZoomOut 가능)

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|----------|----------------|----------------------|
| position | Array.<Number> | 위치 좌표(좌상단 기준) |
| size | Array.<Number> | Canvas Width, Height |
| isFit | Boolean | Fit 여부 |

```
canvas.getScale() ⇒ Number
```

Scale 을 반환한다. (리얼 사이즈 : Scale = 1)

Kind: instance method of [Canvas](#)

Returns: Number - 스케일값

```
canvas.setScale(scale)
```

Scale 을 설정한다. (리얼 사이즈 : Scale = 1)

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|-------|--------|-------------|
| scale | Number | 스케일값 |

```
canvas.show(element)
```

ID에 해당하는 Element 를 캔버스에서 show 한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

```
canvas.hide(element)
```

ID에 해당하는 Element 를 캔버스에서 hide 한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`canvas.appendChild(srcElement, targetElement) ⇒ Element`

Source Element 를 Target Element 아래에 append 한다.

Kind: instance method of [Canvas](#)

Returns: Element - Source Element

| Param | Type | Description |
|----------------------------|-------------------------|-------------|
| <code>srcElement</code> | Element StringElement | 또는 ID |
| <code>targetElement</code> | Element StringElement | 또는 ID |

`canvas.insertAfter(srcElement, targetElement) ⇒ Element`

Source Element 를 Target Element 이후에 insert 한다.

Kind: instance method of [Canvas](#)

Returns: Element - Source Element

| Param | Type | Description |
|----------------------------|-------------------------|-------------|
| <code>srcElement</code> | Element StringElement | 또는 ID |
| <code>targetElement</code> | Element StringElement | 또는 ID |

`canvas.insertBefore(srcElement, targetElement) ⇒ Element`

Source Element 를 Target Element 이전에 insert 한다.

Kind: instance method of [Canvas](#)

Returns: Element - Source Element

| Param | Type | Description |
|---------------|-------------------------|-------------|
| srcElement | Element StringElement | 또는 ID |
| targetElement | Element StringElement | 또는 ID |

canvas.move(element, offset) ⇒ Element

해당 Element 를 가로, 세로 Offset 만큼 이동한다.

Kind: instance method of [Canvas](#)

Returns: Element - Element

| Param | Type | Description |
|---------|-------------------------|-------------|
| element | Element StringElement | 또는 ID |
| offset | Array.<Number> | [가로, 세로] |

canvas.moveCentroid(element, position) ⇒ Element

주어진 중심좌표로 해당 Element 를 이동한다.

Kind: instance method of [Canvas](#)

Returns: Element - Element

| Param | Type | Description |
|----------|-------------------------|-------------|
| element | Element StringElement | 또는 ID |
| position | Array.<Number> | [x, y] |

canvas.rotate(element, angle) ⇒ Element

중심 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [Canvas](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |
| angle | Number | 각도 |

canvas.resize(element, offset) ⇒ Element

상, 하, 좌, 우 외곽선을 이동한 만큼 리사이즈 한다.

Kind: instance method of [Canvas](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|-------------------------|
| element | Element String | Element 또는 ID |
| offset | Array.<Number> | [상, 하, 좌, 우] 각 방향으로 + 값 |

canvas.resizeBox(element, size) ⇒ Element

중심좌표는 고정한 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [Canvas](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|-----------------|
| element | Element String | Element 또는 ID |
| size | Array.<Number> | [Width, Height] |

canvas.clone(element) ⇒ Element

노드 Element 를 복사한다.

Kind: instance method of [Canvas](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

canvas.getBoundary(element) ⇒ [Envelope](#)

ID에 해당하는 Element 의 바운더리 영역을 리턴한다.

Kind: instance method of [Canvas](#)

Returns: [Envelope](#) - Envelope 영역

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

canvas.getElementById(id) ⇒ Element

ID로 Node Element 를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Element - Element

| Param | Type |
|-------|--------|
| id | String |

canvas.getElementsByType(shapeType, excludeType) ⇒ Array.<Element>

Shape 타입에 해당하는 Node Element 들을 반환한다.

Kind: instance method of [Canvas](#)

Returns: Array.<Element> - Element's Array

| Param | Type | Description |
|-------------|--------|---|
| shapeType | String | Shape 타입(GEOM, HTML, IMAGE, EDGE, GROUP), Null 이면 모든 타입 |
| excludeType | String | 제외 할 타입 |

`canvas.getElementsByShapeId(shapeId) ⇒ Array.<Element>`

Shape ID에 해당하는 Node Element 들을 반환한다.

Kind: instance method of [Canvas](#)

Returns: Array.<Element> - Element's Array

| Param | Type | Description |
|---------|--------|-------------|
| shapeId | String | Shape ID |

`canvas.getRelatedElementsFromEdge(edgeElement) ⇒ Object`

Edge 엘리먼트와 연결된 fromShape, toShape 엘리먼트를 반환한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|-------------|------------------|---------------|
| edgeElement | Element String | Element 또는 ID |

`canvas.getParent(Element) ⇒ Element`

부모 엘리먼트를 반환한다. 부모가 루트일때는 반환하지 않는다.

Kind: instance method of [Canvas](#)

Returns: Element - Element 엘리먼트

| Param | Type | Description |
|---------|---------|-------------|
| Element | Element | 엘리먼트 |

`canvas.getChilds(element) ⇒ Array`

그룹의 하위 엘리먼트를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Array - Elements

| Param | Type | Description |
|---------|---------|-------------|
| element | Element | 엘리먼트 |

`canvas.getAllShapes() ⇒ Array`

캔버스의 모든 Shape 들을 리턴

Kind: instance method of [Canvas](#)

Returns: Array - Elements

`canvas.getAllEdges() ⇒ Array`

캔버스의 모든 Edge를 리턴

Kind: instance method of [Canvas](#)

Returns: Array - Edge Elements

`canvas.getBBox(element) ⇒ Object`

해당 엘리먼트의 BoundingBox 영역 정보를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Object - {width, height, x, y, x2, y2}

| Param | Type |
|---------|------------------|
| element | Element String |

`canvas.getRootBBox() ⇒ Object`

부모노드기준으로 캔버스 루트 엘리먼트의 BoundingBox 영역 정보를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Object - {width, height, x, y, x2, y2}

`canvas.getRealRootBBox() ⇒ Object`

부모노드기준으로 캔버스 루트 엘리먼트의 실제 Shape 이 차지하는 BoundingBox 영역 정보를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Object - {width, height, x, y, x2, y2}

`canvas.isSVG()` ⇒ Boolean

SVG 인지 여부를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Boolean - svg 여부

`canvas.isVML()` ⇒ Boolean

VML 인지 여부를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Boolean - vml 여부

`canvas.setCustomData(shapeElement, data)`

주어진 Shape 엘리먼트에 커스텀 데이터를 저장한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|------------------|-------------------------|
| shapeElement | Element String | Shape DOM Element or ID |
| data | Object | JSON 포맷의 Object |

`canvas.getCustomData(shapeElement)` ⇒ Object

주어진 Shape 엘리먼트에 저장된 커스텀 데이터를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Object - JSON 포맷의 Object

| Param | Type | Description |
|--------------|------------------|-------------------------|
| shapeElement | Element String | Shape DOM Element or ID |

```
canvas.setExtCustomData(shapeElement, data)
```

주어진 Shape 엘리먼트에 확장 커스텀 데이터를 저장한다.

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|------------------|-------------------------|
| shapeElement | Element String | Shape DOM Element or ID |
| data | Object | JSON 포맷의 Object |

```
canvas.getExtCustomData(shapeElement) ⇒ Object
```

주어진 Shape 엘리먼트에 저장된 확장 커스텀 데이터를 반환한다.

Kind: instance method of [Canvas](#)

Returns: Object - JSON 포맷의 Object

| Param | Type | Description |
|--------------|------------------|-------------------------|
| shapeElement | Element String | Shape DOM Element or ID |

```
canvas.toXML() ⇒ String
```

Canvas 에 그려진 Shape 들을 OpenGraph XML 문자열로 export 한다.

Kind: instance method of [Canvas](#)

Returns: String - XML 문자열

`canvas.toJSON()` ⇒ Object

Canvas 에 그려진 Shape 들을 OpenGraph JSON 객체로 export 한다.

Kind: instance method of [Canvas](#)

Returns: Object - JSON 포맷의 Object

`canvas.loadXML(xml)` ⇒ Object

OpenGraph XML 문자열로 부터 Shape 을 드로잉한다.

Kind: instance method of [Canvas](#)

Returns: Object - {width, height, x, y, x2, y2}

| Param | Type | Description |
|-------|------------------|------------------------|
| xml | String Element | XML 문자열 또는 DOM Element |

`canvas.loadJSON(json)` ⇒ Object

JSON 객체로 부터 Shape 을 드로잉한다.

Kind: instance method of [Canvas](#)

Returns: Object - {width, height, x, y, x2, y2}

| Param | Type | Description |
|-------|--------|-----------------|
| json | Object | JSON 포맷의 Object |

canvas.undo()

캔버스 undo.

Kind: instance method of [Canvas](#)

canvas.redo()

캔버스 redo.

Kind: instance method of [Canvas](#)

canvas.getPrevEdges(element) ⇒ Array.<Element>

연결된 이전 Edge Element 들을 반환한다.

Kind: instance method of [Canvas](#)

Returns: Array.<Element> - Previous Element's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

canvas.getNextEdges(element) ⇒ Array.<Element>

연결된 이후 Edge Element 들을 반환한다.

Kind: instance method of [Canvas](#)

Returns: Array.<Element> - Previous Element's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

canvas.getPrevShapes(element) ⇒ Array.<Element>

연결된 이전 노드 Element 들을 반환한다.

Kind: instance method of [Canvas](#)

Returns: Array.<Element> - Previous Element's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

canvas.getPrevShapeIds(element) ⇒ Array.<String>

연결된 이전 노드 Element ID들을 반환한다.

Kind: instance method of [Canvas](#)

Returns: Array.<String> - Previous Element Id's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

canvas.getNextShapes(element) ⇒ Array.<Element>

연결된 이후 노드 Element 들을 반환한다.

Kind: instance method of [Canvas](#)

Returns: Array.<Element> - Previous Element's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

canvas.getNextShapeIds(element) ⇒ Array.<String>

연결된 이후 노드 Element ID들을 반환한다.

Kind: instance method of [Canvas](#)

Returns: Array.<String> - Previous Element Id's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

canvas.onDrawShape(callbackFunc)

Shape 이 처음 Draw 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|---------------------------|
| callbackFunc | function | 콜백함수(event, shapeElement) |

canvas.onUndo(callbackFunc)

Undo 되었을때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|-------------|
| callbackFunc | function | 콜백함수(event) |

canvas.onRedo(callbackFunc)

Redo 되었을때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|-------------|
| callbackFunc | function | 콜백함수(event) |

canvas.onDivideLane(callbackFunc)

Lane 이 divide 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|--------------------------|
| callbackFunc | function | 콜백함수(event, dividedLane) |

canvas.onDrawLabel(callbackFunc)

라벨이 Draw 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|--------------------------------------|
| callbackFunc | function | 콜백함수(event, shapeElement, labelText) |

canvas.onLabelChanged(callbackFunc)

라벨이 Change 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|--|
| callbackFunc | function | 콜백함수(event, shapeElement, afterText, beforeText) |

canvas.onBeforeLabelChange(callbackFunc)

라벨이 Change 되기전 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|--|
| callbackFunc | function | 콜백함수(event, shapeElement, afterText, beforeText) |

canvas.onRedrawShape(callbackFunc)

Shape 이 Redraw 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|---------------------------|
| callbackFunc | function | 콜백함수(event, shapeElement) |

`canvas.onRemoveShape(callbackFunc)`

Shape 이 Remove 될 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|---------------------------|
| callbackFunc | function | 콜백함수(event, shapeElement) |

`canvas.onRotateShape(callbackFunc)`

Shape 이 Rotate 될 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|-----------------------------|
| callbackFunc | function | 콜백함수(event, element, angle) |

`canvas.onMoveShape(callbackFunc)`

Shape 이 Move 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|-----------------------------------|
| callbackFunc | function | 콜백함수(event, shapeElement, offset) |

`canvas.onResizeShape(callbackFunc)`

Shape 이 Resize 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|-----------------------------------|
| callbackFunc | function | 콜백함수(event, shapeElement, offset) |

canvas.onBeforeConnectShape(callbackFunc)

Shape 이 Connect 되기전 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|--|
| callbackFunc | function | 콜백함수(event, edgeElement, fromElement, toElement) |

canvas.onBeforeRemoveShape(callbackFunc)

Shape 이 Remove 되기전 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|----------------------|
| callbackFunc | function | 콜백함수(event, element) |

canvas.onConnectShape(callbackFunc)

Shape 이 Connect 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|--|
| callbackFunc | function | 콜백함수(event, edgeElement, fromElement, toElement) |

canvas.onDisconnectShape(callbackFunc)

Shape 이 Disconnect 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|--|
| callbackFunc | function | 콜백함수(event, edgeElement, fromElement, toElement) |

canvas.onGroup(callbackFunc)

Shape 이 Grouping 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|---------------------------|
| callbackFunc | function | 콜백함수(event, groupElement) |

canvas.onUnGroup(callbackFunc)

Shape 이 UnGrouping 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|--------------------------------|
| callbackFunc | function | 콜백함수(event, ungroupedElements) |

`canvas.onCollapsed(callbackFunc)`

Group 이 Collapse 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|----------------------|
| callbackFunc | function | 콜백함수(event, element) |

`canvas.onExpanded(callbackFunc)`

Group 이 Expand 되었을 때의 이벤트 리스너

Kind: instance method of [Canvas](#)

| Param | Type | Description |
|--------------|----------|----------------------|
| callbackFunc | function | 콜백함수(event, element) |

`OG.handler` : object

Kind: static namespace of [OG](#)

- [.handler](#) : object
 - [.EventHandler](#)
 - [new OG.handler.EventHandler\(renderer, config\)](#)
 - [.enableEditLabel\(element\)](#)
 - [.setMovable\(element, isMovable\)](#)
 - [.setConnectable\(element, guide, isConnectable\)](#)

- [.setResizable\(element, guide, isResizable\)](#)
- [.setClickSelectable\(element, isSelectable\)](#)
- [.setGroupDropable\(element\)](#)
- [.setDragSelectable\(isSelectable\)](#)
- [.setEnabledHotKey\(isEnableHotKey\)](#)
- [.enableRootContextMenu\(\)](#)
- [.enableShapeContextMenu\(\)](#)
- [.selectShape\(element\)](#)
- [.selectShapes\(element\)](#)
- [.bringToFront\(\)](#)
- [.sendToBack\(\)](#)
- [.bringForward\(\)](#)
- [.sendBackward\(\)](#)
- [.deleteSelectedShape\(\)](#)
- [.changeShape\(\)](#)
- [.showProperty\(\)](#)
- [.selectAll\(\)](#)
- [.copySelectedShape\(\)](#)
- [.cutSelectedShape\(\)](#)
- [.pasteSelectedShape\(\)](#)
- [.duplicateSelectedShape\(\)](#)
- [.groupSelectedShape\(\)](#)
- [.ungroupSelectedShape\(\)](#)
- [.rotateSelectedShape\(angle\)](#)
- [.setLineWidthSelectedShape\(lineWidth\)](#)
- [.setLineColorSelectedShape\(lineColor\)](#)

- [.setLoopTypeSelectedShape\(lineType\)](#)
- [.setLineStyleSelectedShape\(lineStyle\)](#)
- [.setArrowStartSelectedShape\(arrowType\)](#)
- [.setArrowEndSelectedShape\(arrowType\)](#)
- [.setFillColorSelectedShape\(fillColor\)](#)
- [.setFontFamilySelectedShape\(fontFamily\)](#)
- [.setFontSizeSelectedShape\(fontSize\)](#)
- [.setFontColorSelectedShape\(fontColor\)](#)
- [.setFontWeightSelectedShape\(fontWeight\)](#)
- [.setFontStyleSelectedShape\(fontStyle\)](#)
- [.setTextDecorationSelectedShape\(textDecoration\)](#)
- [.setLabelDirectionSelectedShape\(labelDirection\)](#)
- [.setLabelAngleSelectedShape\(labelAngle\)](#)
- [.setLabelPositionSelectedShape\(labelPosition\)](#)
- [.setLabelVerticalSelectedShape\(verticalAlign\)](#)
- [.setLabelHorizontalSelectedShape\(horizontalAlign\)](#)
- [.setLabelSelectedShape\(label\)](#)
- [.setEdgeFromLabelSelectedShape\(label\)](#)
- [.setEdgeToLabelSelectedShape\(label\)](#)
- [.zoomIn\(\)](#)
- [.zoomOut\(\)](#)
- [.fitWindow\(\)](#)
- [.setConnectGuide\(element, isConnectable\)](#)

handler.EventHandler

Kind: static class of [handler](#)

Requires: module:OG.renderer.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.EventHandler](#)
 - [new OG.handler.EventHandler\(renderer, config\)](#)
 - [.enableEditLabel\(element\)](#)
 - [.setMovable\(element, isMovable\)](#)
 - [.setConnectable\(element, guide, isConnectable\)](#)
 - [.setResizable\(element, guide, isResizable\)](#)
 - [.setClickSelectable\(element, isSelectable\)](#)
 - [.setGroupDropable\(element\)](#)
 - [.setDragSelectable\(isSelectable\)](#)
 - [.setEnabledHotKey\(isEnableHotKey\)](#)
 - [.enableRootContextMenu\(\)](#)
 - [.enableShapeContextMenu\(\)](#)
 - [.selectShape\(element\)](#)
 - [.selectShapes\(element\)](#)
 - [.bringToFront\(\)](#)
 - [.sendToBack\(\)](#)
 - [.bringForward\(\)](#)
 - [.sendBackward\(\)](#)

- [.deleteSelectedShape\(\)](#)
- [.changeShape\(\)](#)
- [.showProperty\(\)](#)
- [.selectAll\(\)](#)
- [.copySelectedShape\(\)](#)
- [.cutSelectedShape\(\)](#)
- [.pasteSelectedShape\(\)](#)
- [.duplicateSelectedShape\(\)](#)
- [.groupSelectedShape\(\)](#)
- [.ungroupSelectedShape\(\)](#)
- [.rotateSelectedShape\(angle\)](#)
- [.setLineWidthSelectedShape\(lineWidth\)](#)
- [.setLineColorSelectedShape\(lineColor\)](#)
- [.setLoopTypeSelectedShape\(lineType\)](#)
- [.setLineStyleSelectedShape\(lineStyle\)](#)
- [.setArrowStartSelectedShape\(arrowType\)](#)
- [.setArrowEndSelectedShape\(arrowType\)](#)
- [.setFillColorSelectedShape\(fillColor\)](#)
- [.setFontFamilySelectedShape\(fontFamily\)](#)
- [.setFontSizeSelectedShape\(fontSize\)](#)
- [.setFontColorSelectedShape\(fontColor\)](#)
- [.setFontWeightSelectedShape\(fontWeight\)](#)
- [.setFontStyleSelectedShape\(fontStyle\)](#)
- [.setTextDecorationSelectedShape\(textDecoration\)](#)
- [.setLabelDirectionSelectedShape\(labelDirection\)](#)
- [.setLabelAngleSelectedShape\(labelAngle\)](#)

- [.setLabelPositionSelectedShape\(labelPosition\)](#)
- [.setLabelVerticalSelectedShape\(verticalAlign\)](#)
- [.setLabelHorizontalSelectedShape\(horizontalAlign\)](#)
- [.setLabelSelectedShape\(label\)](#)
- [.setEdgeFromLabelSelectedShape\(label\)](#)
- [.setEdgeToLabelSelectedShape\(label\)](#)
- [.zoomIn\(\)](#)
- [.zoomOut\(\)](#)
- [.fitWindow\(\)](#)
- [.setConnectGuide\(element, isConnectable\)](#)

```
new OG.handler.EventHandler(renderer, config)
```

Event Handler

| Param | Type | Description |
|----------|----------------------------------|---------------|
| renderer | <u>IRenderer</u> | 렌더러 |
| config | Object | Configuration |

```
eventHandler.enableEditLabel(element)
```

주어진 Shape Element 의 라벨을 수정 가능하도록 한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

element Element Shape Element

eventHandler.setMovable(element, isMovable)

Shape 엘리먼트의 이동 가능여부를 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|-----------|---------------|-------------|
| element | Element Shape | 엘리먼트 |
| isMovable | Boolean | 가능여부 |

eventHandler.setConnectable(element, guide, isConnectable)

Shape 엘리먼트의 라인모양을 클릭하여 Shape 끼리 커넥트가 가능하게 한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|---------------|---------------|----------------|
| element | Element Shape | 엘리먼트 |
| guide | Object | JSON 포맷 가이드 정보 |
| isConnectable | Boolean | 가능여부 |

eventHandler.setResizable(element, guide, isResizable)

Shape 엘리먼트의 리사이즈 가능여부를 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|---------|---------------|-------------|
| element | Element Shape | 엘리먼트 |

guide Object JSON 포맷 가이드 정보
isResizable Boolean 가능여부

```
eventHandler.setClickSelectable(element, isSelectable)
```

주어진 Shape Element 를 마우스 클릭하여 선택가능하도록 한다.

선택가능해야 리사이즈가 가능하다.

선택시 커넥트 모드일 경우 connect 가능하게 한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|--------------|---------------|-------------|
| element | Element Shape | Element |
| isSelectable | Boolean | 선택가능여부 |

```
eventHandler.setGroupDropable(element)
```

Lane, Pool 엘리먼트가 새로 생성될 시 그룹을 맺도록 한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|---------|---------------|-------------|
| element | Element Shape | 엘리먼트 |

```
eventHandler.setDragSelectable(isSelectable)
```

마우스 드래그 영역지정 선택가능여부를 설정한다.

선택가능해야 리사이즈가 가능하다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|--------------|---------|-------------|
| isSelectable | Boolean | 선택가능여부 |

`eventHandler.setEnableHotKey(isEnableHotKey)`

HotKey 사용 가능여부를 설정한다. (Delete, Ctrl+A, Ctrl+C, Ctrl+V, Ctrl+G, Ctrl+U)

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|----------------|---------|-------------|
| isEnableHotKey | Boolean | 핫키가능여부 |

`eventHandler.enableRootContextMenu()`

캔버스에 마우스 우클릭 메뉴를 가능하게 한다.

Kind: instance method of [EventHandler](#)

`eventHandler.enableShapeContextMenu()`

Shape 에 마우스 우클릭 메뉴를 가능하게 한다.

Kind: instance method of [EventHandler](#)

`eventHandler.selectShape(element)`

주어진 Shape Element 를 선택된 상태로 되게 한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|---------|---------------|-------------|
| element | Element Shape | 엘리먼트 |

```
eventHandler.selectShapes(element)
```

주어진 다수의 Shape Element 를 선택된 상태로 되게 한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|---------|---------------|-------------|
| element | Element Shape | 엘리먼트 |

```
eventHandler.bringToFront()
```

메뉴 : 맨 앞으로 가져오기

Kind: instance method of [EventHandler](#)

```
eventHandler.sendToBack()
```

메뉴 : 맨 뒤로 보내기

Kind: instance method of [EventHandler](#)

```
eventHandler.bringForward()
```


메뉴 : 앞으로 가져오기

Kind: instance method of [EventHandler](#)

`eventHandler.sendBackward()`

메뉴 : 뒤로 보내기

Kind: instance method of [EventHandler](#)

`eventHandler.deleteSelectedShape()`

메뉴 : 선택된 Shape 들을 삭제한다.

Kind: instance method of [EventHandler](#)

`eventHandler.changeShape()`

메뉴 : Shape를 선택한 모양으로 변경한다.

Kind: instance method of [EventHandler](#)

`eventHandler.showProperty()`

메뉴 : 속성 창 이벤트

Kind: instance method of [EventHandler](#)

`eventHandler.selectAll()`

메뉴 : 모든 Shape 들을 선택한다.

Kind: instance method of [EventHandler](#)

`eventHandler.copySelectedShape()`

메뉴 : 선택된 Shape 들을 복사한다.

Kind: instance method of [EventHandler](#)

`eventHandler.cutSelectedShape()`

메뉴 : 선택된 Shape 들을 잘라내기한다.

Kind: instance method of [EventHandler](#)

`eventHandler.pasteSelectedShape()`

메뉴 : 선택된 Shape 들을 붙여넣기 한다.

Kind: instance method of [EventHandler](#)

`eventHandler.duplicateSelectedShape()`

메뉴 : 선택된 Shape 들을 복제한다.

Kind: instance method of [EventHandler](#)

```
eventHandler.groupSelectedShape()
```

메뉴 : 선택된 Shape 들을 그룹핑한다.

Kind: instance method of [EventHandler](#)

```
eventHandler.ungroupSelectedShape()
```

메뉴 : 선택된 Shape 들을 그룹해제한다.

Kind: instance method of [EventHandler](#)

```
eventHandler.rotateSelectedShape(angle)
```

메뉴 : 선택된 Shape 들을 회전한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|-------|--------|-------------|
| angle | Number | 회전각도 |

```
eventHandler.setLineWidthSelectedShape(lineWidth)
```

메뉴 : 선택된 Shape 들의 Line Width 를 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type |
|-----------|--------|
| lineWidth | Number |

```
eventHandler.setLineColorSelectedShape(lineColor)
```

메뉴 : 선택된 Shape 들의 Line Color 를 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type |
|-----------|--------|
| lineColor | String |

```
eventHandler.setLoopTypeSelectedShape(lineType)
```

메뉴 : 선택된 Shape 들의 Line Type 을 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|----------|--------|-------------------------------|
| lineType | String | ['straight' 'plain' 'bezier'] |

```
eventHandler.setLineStyleSelectedShape(lineStyle)
```

메뉴 : 선택된 Shape 들의 Line Style 을 설정한다.

Kind: instance method of [EventHandler](#)


```
eventHandler.setFontFamilySelectedShape(fontFamily)
```

메뉴 : 선택된 Shape 들의 Font Family 를 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type |
|------------|--------|
| fontFamily | String |

```
eventHandler.setFontSizeSelectedShape(fontSize)
```

메뉴 : 선택된 Shape 들의 Font Size 를 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type |
|----------|--------|
| fontSize | Number |

```
eventHandler.setFontColorSelectedShape(fontColor)
```

메뉴 : 선택된 Shape 들의 Font Color 를 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type |
|-----------|--------|
| fontColor | String |

```
eventHandler.setFontWeightSelectedShape(fontWeight)
```

메뉴 : 선택된 Shape 들의 Font Weight 를 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|------------|--------|-------------------|
| fontWeight | String | ['bold' 'normal'] |

```
eventHandler.setFontStyleSelectedShape(fontStyle)
```

메뉴 : 선택된 Shape 들의 Font Style 을 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|-----------|--------|---------------------|
| fontStyle | String | ['italic' 'normal'] |

```
eventHandler.setTextDecorationSelectedShape(textDecoration)
```

메뉴 : 선택된 Shape 들의 Text Decoration 을 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|----------------|--------|----------------------|
| textDecoration | String | ['underline' 'none'] |

```
eventHandler.setLabelDirectionSelectedShape(labelDirection)
```

메뉴 : 선택된 Shape 들의 Label Direction 을 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|----------------------|---------------------------|-------------|
| labelDirectionString | ['vertical' 'horizontal'] | |

eventHandler.setLabelAngleSelectedShape(labelAngle)

메뉴 : 선택된 Shape 들의 Label Angle 을 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type |
|------------|--------|
| labelAngle | Number |

eventHandler.setLabelPositionSelectedShape(labelPosition)

메뉴 : 선택된 Shape 들의 Label Position 을 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|---------------------|--|-------------|
| labelPositionString | ['top' 'bottom' 'left' 'right' 'center'] | |

eventHandler.setLabelVerticalSelectedShape(verticalAlign)

메뉴 : 선택된 Shape 들의 라벨 Vertical Align 를 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|---------------------|---------------------------|-------------|
| verticalAlignString | ['top' 'middle' 'bottom'] | |


```
eventHandler.setLabelHorizontalSelectedShape(horizontalAlign)
```

메뉴 : 선택된 Shape 들의 라벨 Horizontal Align 를 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|-----------------|--------|--------------------------|
| horizontalAlign | String | ['start' 'middle' 'end'] |

```
eventHandler.setLabelSelectedShape(label)
```

메뉴 : 선택된 Shape 의 라벨을 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type |
|-------|--------|
| label | String |

```
eventHandler.setEdgeFromLabelSelectedShape(label)
```

메뉴 : 선택된 Edge Shape 의 시작점 라벨을 설정한다.

Kind: instance method of [EventHandler](#)

| Param | Type |
|-------|--------|
| label | String |

```
eventHandler.setEdgeToLabelSelectedShape(label)
```

메뉴 : 선택된 Edge Shape 의 끝점 라벨을 설정한다.

Kind: instance method of [EventHandler](#)

Param Type
label String

eventHandler.zoomIn()

메뉴 : Zoom In

Kind: instance method of [EventHandler](#)

eventHandler.zoomOut()

메뉴 : Zoom Out

Kind: instance method of [EventHandler](#)

eventHandler.fitWindow()

메뉴 : 그려진 Shape 들을 캔버스 사이즈에 맞게 조절한다.

Kind: instance method of [EventHandler](#)

eventHandler.setConnectGuide(element, isConnectable)

Shape 엘리먼트의 setConnectGuide 에 관련된 이벤트

Kind: instance method of [EventHandler](#)

| Param | Type | Description |
|---------------|---------|-------------|
| element | Element | Shape 엘리먼트 |
| isConnectable | Boolean | 가능여부 |

OG.layout : object

Kind: static namespace of [OG](#)

OG.renderer : object

Kind: static namespace of [OG](#)

- [.renderer](#) : object
 - [.IRenderer](#)
 - [new OG.renderer.IRenderer\(container, containerSize, backgroundColor, backgroundImage, config\)](#)
 - [.drawShape\(position, shape, size, style, id\)](#) ⇒ Element
 - [.drawGeom\(geometry, style\)](#) ⇒ Element
 - [.drawText\(position, text, size, style, id\)](#) ⇒ Element
 - [.drawImage\(position, imgSrc, size, style, id\)](#) ⇒ Element
 - [.drawEdge\(line, style, id, isSelf\)](#) ⇒ Element
 - [.drawLabel\(shapeElement, text, style\)](#) ⇒ Element
 - [.drawEdgeLabel\(shapeElement, text, type\)](#) ⇒ Element
 - [.redrawShape\(element, excludeEdgeId\)](#)

- [.redrawConnectedEdge\(element\)](#)
- [.connect\(fromTerminal, toTerminal, edge, style, label, preventTrigger\)](#) ⇒ Element
- [.disconnect\(element\)](#)
- [.drawDropOverGuide\(element\)](#)
- [.drawGuide\(element\)](#) ⇒ Object
- [.removeGuide\(element\)](#)
- [.removeAllGuide\(\)](#)
- [.removeConnectGuide\(element\)](#)
- [.removeAllConnectGuide\(\)](#)
- [.removeOtherConnectGuide\(element\)](#)
- [.removeAllVirtualEdge\(\)](#)
- [.drawEdgeGuide\(element\)](#) ⇒ Object
- [.drawRubberBand\(position, size, style\)](#) ⇒ Element
- [.removeRubberBand\(root\)](#)
- [.drawDraggableGuide\(element\)](#) ⇒ Element
- [.drawCollapseGuide\(element\)](#) ⇒ Element
- [.removeCollapseGuide\(element\)](#)
- [.group\(elements\)](#) ⇒ Element
- [.ungroup\(groupElements\)](#) ⇒ Array.<Element>
- [.addToGroup\(groupElement, elements\)](#)
- [.collapse\(element\)](#)
- [.expand\(element\)](#)
- [.clear\(\)](#)
- [.removeShape\(element\)](#)
- [.remove\(element\)](#)

- [.removeChild\(element\)](#)
- [.getRootElement\(\)](#) ⇒ Element
- [.getRootGroup\(\)](#) ⇒ Element
- [.getElementByPoint\(position\)](#) ⇒ Element
- [.getElementsByBBox\(envelope\)](#) ⇒ Array.<Element>
- [.setAttr\(element, attribute\)](#)
- [.getAttr\(element, attrName\)](#) ⇒ Object
- [.setShapeStyle\(element, style\)](#)
- [.toFront\(element\)](#)
- [.toBack\(element\)](#)
- [.bringForward\(element\)](#)
- [.sendBackward\(element\)](#)
- [.getCanvasSize\(\)](#) ⇒ Array.<Number>
- [.setCanvasSize\(size\)](#)
- [.fitCanvasSize\(minSize, fitScale\)](#)
- [.setViewBox\(position, size, isFit\)](#)
- [.getScale\(\)](#) ⇒ Number
- [.setScale\(scale\)](#)
- [.show\(element\)](#)
- [.hide\(element\)](#)
- [.appendChild\(srcElement, targetElement\)](#) ⇒ Element
- [.insertAfter\(srcElement, targetElement\)](#) ⇒ Element
- [.insertBefore\(srcElement, targetElement\)](#) ⇒ Element
- [.move\(element, offset\)](#) ⇒ Element
- [.moveCentroid\(element, position\)](#) ⇒ Element
- [.rotate\(element, angle\)](#) ⇒ Element

- [.resize\(element, offset\)](#) ⇒ Element
- [.resizeBox\(element, size\)](#) ⇒ Element
- [.clone\(element\)](#) ⇒ Element
- [.getElementById\(id\)](#) ⇒ Element
- [.getElementsByType\(shapeType, excludeType\)](#) ⇒ Array.<Element>
- [.getBBox\(element\)](#) ⇒ Object
- [.getRootBBox\(\)](#) ⇒ Object
- [.getRealRootBBox\(\)](#) ⇒ Object
- [.getContainer\(\)](#) ⇒ Element
- [.isSVG\(\)](#) ⇒ Boolean
- [.isVML\(\)](#) ⇒ Boolean
- [.getPrevEdges\(element\)](#) ⇒ Array.<Element>
- [.getNextEdges\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapes\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapeIds\(element\)](#) ⇒ Array.<String>
- [.getNextShapes\(element\)](#) ⇒ Array.<Element>
- [.getNextShapeIds\(element\)](#) ⇒ Array.<String>
- [.getConnectGuideElements\(Element\)](#) ⇒ Array
- [.isTopGroup\(Element\)](#) ⇒ boolean
- [.getParent\(element\)](#) ⇒ Element
- [.getChilds\(element\)](#) ⇒ Array
- [.isGroup\(element\)](#) ⇒ boolean
- [.getAllShapes\(\)](#) ⇒ Array
- [.getAllEdges\(\)](#) ⇒ Array
- [.getAllNotEdges\(\)](#) ⇒ Array
- [.isEdge\(\)](#) ⇒ boolean

- [.isShape\(\)](#) ⇒ boolean
- [.initHistory\(\)](#)
- [.addHistory\(\)](#)
- [.undo\(\)](#)
- [.redo\(\)](#)
- [.RaphaelRenderer](#) ⇐ [IRenderer](#)
 - [new OG.renderer.RaphaelRenderer\(container, containerSize, backgroundColor, backgroundImage, config\)](#)
 - [.drawHtml\(position, html, size, style, id\)](#) ⇒ Element
 - [._getPointOfInflectionFromEdge\(\)](#)
 - [.reconnect\(edge\)](#) ⇒ Element
 - [.disconnectOneWay\(element, connectDirection\)](#)
 - [.drawStickGuide\(element, position\)](#)
 - [.setTextListInController\(element, textList\)](#)
 - [.getTextListInController\(element\)](#)
 - [.getConnectGuideElements\(Element\)](#) ⇒ Array
 - [.getNotConnectGuideElements\(Element\)](#) ⇒ Array
 - [.removeConnectGuide\(element\)](#)
 - [.removeAllConnectGuide\(\)](#)
 - [.removeOtherConnectGuide\(element\)](#)
 - [.getSpots\(element\)](#) ⇒ Array
 - [.getCircleSpots\(element\)](#) ⇒ Array
 - [.createVirtualSpot\(x, x, element\)](#) ⇒ Element
 - [.getVirtualSpot\(element\)](#) ⇒ Element
 - [.removeVirtualSpot\(element\)](#) ⇒ Element

- [.selectSpot\(선택한\)](#)
- [.getChildNodes\(element\)](#) ⇒ Array
- [.trimEdge\(element\)](#)
- [.trimConnectInnerVertice\(element\)](#) ⇒ Element
- [.trimConnectIntersection\(element\)](#) ⇒ Element
- [.getBoundary\(element\)](#) ⇒ [Envelope](#)
- [.setHighlight\(element, highlight\)](#)
- [.removeHighlight\(element, highlight\)](#)
- [.createTerminalString\(Element, point\)](#) ⇒ String
- [.createDefaultTerminalString\(Element\)](#) ⇒ String
- [.toFrontEdges\(\)](#)
- [.removeAllEdgeGuide\(\)](#)
- [.createVirtualEdge\(x, x, targetEle\)](#) ⇒ Element
- [.updateVirtualEdge\(x, x\)](#)
- [.getTargetfromVirtualEdge\(x, x\)](#)
- [.removeAllVirtualEdge\(\)](#)
- [.isLane\(Element\)](#) ⇒ boolean
- [.isPool\(Element\)](#) ⇒ boolean
- [.isScopeActivity\(Element\)](#) ⇒ boolean
- [.isHorizontalLane\(Element\)](#) ⇒ boolean
- [.isVerticalLane\(Element\)](#) ⇒ boolean
- [.isHorizontalPool\(Element\)](#) ⇒ boolean
- [.isVerticalPool\(Element\)](#) ⇒ boolean
- [.getChildLane\(Element\)](#) ⇒ Array
- [.enableDivideCount\(Element\)](#) ⇒ Number
- [.getExceptTitleLaneArea\(Element, boundary\)](#)

- [.divideLane\(Element, quarterOrder\)](#)
- [.getBaseLanes\(Element\)](#) ⇒ Array
- [.getRootLane\(Element\)](#) ⇒ Element
- [.getIndexOfLane\(Element\)](#) ⇒ Number
- [.getDepthOfLane\(Element\)](#) ⇒ Number
- [.reEstablishLane\(Element\)](#)
- [.getBoundaryOfElements\(elements\)](#) ⇒ [Envelope](#)
- [.getNearestBaseLaneIndexAsDirection\(Element, direction\)](#) ⇒ Number
- [.getBoundaryOfInnerShapesGroup\(Element\)](#) ⇒ [Envelope](#)
- [.getSmallestBaseLane\(Element, baseLane\)](#)
- [.resizeLane\(Element, offset\)](#)
- [.removeLaneShape\(Element\)](#)
- [.getInnerShapesOfLane\(Element\)](#)
- [.fitLaneOrder\(Element\)](#)
- [.getRootGroupOfShape\(Element\)](#) ⇒ Element
- [.checkBridgeEdge\(Element\)](#)
- [.checkAllBridgeEdge\(\)](#)
- [.getInnerShapesOfGroup\(Element\)](#)
- [.getFrontForCoordinate\(point\)](#) ⇒ Element
- [.getFrontForBoundary\(boundary\)](#) ⇒ Element
- [.trimEdgeDirection\(Edge, FromShape, ToShape\)](#) ⇒ Element
- [.putInnerShapeToPool\(Element\)](#) ⇒ Element
- [.setDropablePool\(Element\)](#) ⇒ Element
- [.offDropablePool\(\)](#)
- [.drawShape\(position, shape, size, style, id\)](#) ⇒ Element

- [.drawGeom\(geometry, style\)](#) ⇒ Element
- [.drawText\(position, text, size, style, id\)](#) ⇒ Element
- [.drawImage\(position, imgSrc, size, style, id\)](#) ⇒ Element
- [.drawEdge\(line, style, id, isSelf\)](#) ⇒ Element
- [.drawLabel\(shapeElement, text, style\)](#) ⇒ Element
 - [~getCenterOfEdge\(element\)](#) ⇒ OG.Coordinate
- [.drawEdgeLabel\(shapeElement, text, type\)](#) ⇒ Element
- [.redrawShape\(element, excludeEdgeId\)](#)
- [.redrawConnectedEdge\(element\)](#)
- [.connect\(fromTerminal, toTerminal, edge, style, label, preventTrigger\)](#) ⇒ Element
- [.disconnect\(element\)](#)
- [.drawDropOverGuide\(element\)](#)
- [.drawGuide\(element\)](#) ⇒ Object
- [.removeGuide\(element\)](#)
- [.removeAllGuide\(\)](#)
- [.drawEdgeGuide\(element\)](#) ⇒ Object
- [.drawRubberBand\(position, size, style\)](#) ⇒ Element
- [.removeRubberBand\(root\)](#)
- [.drawDraggableGuide\(element\)](#) ⇒ Element
- [.drawCollapseGuide\(element\)](#) ⇒ Element
- [.removeCollapseGuide\(element\)](#)
- [.group\(elements\)](#) ⇒ Element
- [.ungroup\(groupElements\)](#) ⇒ Array.<Element>
- [.addToGroup\(groupElement, elements\)](#)

- [.collapse\(element\)](#)
- [.expand\(element\)](#)
- [.clear\(\)](#)
- [.removeShape\(element\)](#)
- [.remove\(element\)](#)
- [.removeChild\(element\)](#)
- [.getRootElement\(\)](#) ⇒ Element
- [.getRootGroup\(\)](#) ⇒ Element
- [.getElementByPoint\(position\)](#) ⇒ Element
- [.getElementsByBBox\(envelope\)](#) ⇒ Array.<Element>
- [.setAttr\(element, attribute\)](#)
- [.getAttr\(element, attrName\)](#) ⇒ Object
- [.setShapeStyle\(element, style\)](#)
- [.toFront\(element\)](#)
- [.toBack\(element\)](#)
- [.bringForward\(element\)](#)
- [.sendBackward\(element\)](#)
- [.getCanvasSize\(\)](#) ⇒ Array.<Number>
- [.setCanvasSize\(size\)](#)
- [.fitCanvasSize\(minSize, fitScale\)](#)
- [.setViewBox\(position, size, isFit\)](#)
- [.getScale\(\)](#) ⇒ Number
- [.setScale\(scale\)](#)
- [.show\(element\)](#)
- [.hide\(element\)](#)
- [.appendChild\(srcElement, targetElement\)](#) ⇒ Element

- [.insertAfter\(srcElement, targetElement\)](#) ⇒ Element
- [.insertBefore\(srcElement, targetElement\)](#) ⇒ Element
- [.move\(element, offset\)](#) ⇒ Element
- [.moveCentroid\(element, position\)](#) ⇒ Element
- [.rotate\(element, angle\)](#) ⇒ Element
- [.resize\(element, offset\)](#) ⇒ Element
- [.resizeBox\(element, size\)](#) ⇒ Element
- [.clone\(element\)](#) ⇒ Element
- [.getElementById\(id\)](#) ⇒ Element
- [.getElementsByType\(shapeType, excludeType\)](#) ⇒ Array.
<Element>
- [.getBBox\(element\)](#) ⇒ Object
- [.getRootBBox\(\)](#) ⇒ Object
- [.getRealRootBBox\(\)](#) ⇒ Object
- [.getContainer\(\)](#) ⇒ Element
- [.isSVG\(\)](#) ⇒ Boolean
- [.isVML\(\)](#) ⇒ Boolean
- [.getPrevEdges\(element\)](#) ⇒ Array.<Element>
- [.getNextEdges\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapes\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapeIds\(element\)](#) ⇒ Array.<String>
- [.getNextShapes\(element\)](#) ⇒ Array.<Element>
- [.getNextShapeIds\(element\)](#) ⇒ Array.<String>
- [.isTopGroup\(Element\)](#) ⇒ boolean
- [.getParent\(element\)](#) ⇒ Element
- [.getChilDs\(element\)](#) ⇒ Array
- [.isGroup\(element\)](#) ⇒ boolean

- [.getAllShapes\(\)](#) ⇒ Array
- [.getAllEdges\(\)](#) ⇒ Array
- [.getAllNotEdges\(\)](#) ⇒ Array
- [.isEdge\(\)](#) ⇒ boolean
- [.isShape\(\)](#) ⇒ boolean
- [.initHistory\(\)](#)
- [.addHistory\(\)](#)
- [.undo\(\)](#)
- [.redo\(\)](#)

renderer.IRenderer

Kind: static class of [renderer](#)

Requires: module:OG.common.*, module:OG.geometry.*, module:OG.shape.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.IRenderer](#)
 - [new OG.renderer.IRenderer\(container, containerSize, backgroundColor, backgroundImage, config\)](#)
 - [.drawShape\(position, shape, size, style, id\)](#) ⇒ Element
 - [.drawGeom\(geometry, style\)](#) ⇒ Element
 - [.drawText\(position, text, size, style, id\)](#) ⇒ Element
 - [.drawImage\(position, imgSrc, size, style, id\)](#) ⇒ Element

- [.drawEdge\(line, style, id, isSelf\)](#) ⇒ Element
- [.drawLabel\(shapeElement, text, style\)](#) ⇒ Element
- [.drawEdgeLabel\(shapeElement, text, type\)](#) ⇒ Element
- [.redrawShape\(element, excludeEdgeId\)](#)
- [.redrawConnectedEdge\(element\)](#)
- [.connect\(fromTerminal, toTerminal, edge, style, label, preventTrigger\)](#) ⇒ Element
- [.disconnect\(element\)](#)
- [.drawDropOverGuide\(element\)](#)
- [.drawGuide\(element\)](#) ⇒ Object
- [.removeGuide\(element\)](#)
- [.removeAllGuide\(\)](#)
- [.removeConnectGuide\(element\)](#)
- [.removeAllConnectGuide\(\)](#)
- [.removeOtherConnectGuide\(element\)](#)
- [.removeAllVirtualEdge\(\)](#)
- [.drawEdgeGuide\(element\)](#) ⇒ Object
- [.drawRubberBand\(position, size, style\)](#) ⇒ Element
- [.removeRubberBand\(root\)](#)
- [.drawDraggableGuide\(element\)](#) ⇒ Element
- [.drawCollapseGuide\(element\)](#) ⇒ Element
- [.removeCollapseGuide\(element\)](#)
- [.group\(elements\)](#) ⇒ Element
- [.ungroup\(groupElements\)](#) ⇒ Array.<Element>
- [.addToGroup\(groupElement, elements\)](#)
- [.collapse\(element\)](#)

- [.expand\(element\)](#)
- [.clear\(\)](#)
- [.removeShape\(element\)](#)
- [.remove\(element\)](#)
- [.removeChild\(element\)](#)
- [.getRootElement\(\)](#) ⇒ Element
- [.getRootGroup\(\)](#) ⇒ Element
- [.getElementByPoint\(position\)](#) ⇒ Element
- [.getElementsByBBox\(envelope\)](#) ⇒ Array.<Element>
- [.setAttr\(element, attribute\)](#)
- [.getAttr\(element, attrName\)](#) ⇒ Object
- [.setShapeStyle\(element, style\)](#)
- [.toFront\(element\)](#)
- [.toBack\(element\)](#)
- [.bringForward\(element\)](#)
- [.sendBackward\(element\)](#)
- [.getCanvasSize\(\)](#) ⇒ Array.<Number>
- [.setCanvasSize\(size\)](#)
- [.fitCanvasSize\(minSize, fitScale\)](#)
- [.setViewBox\(position, size, isFit\)](#)
- [.getScale\(\)](#) ⇒ Number
- [.setScale\(scale\)](#)
- [.show\(element\)](#)
- [.hide\(element\)](#)
- [.appendChild\(srcElement, targetElement\)](#) ⇒ Element
- [.insertAfter\(srcElement, targetElement\)](#) ⇒ Element

- [.insertBefore\(srcElement, targetElement\)](#) ⇒ Element
- [.move\(element, offset\)](#) ⇒ Element
- [.moveCentroid\(element, position\)](#) ⇒ Element
- [.rotate\(element, angle\)](#) ⇒ Element
- [.resize\(element, offset\)](#) ⇒ Element
- [.resizeBox\(element, size\)](#) ⇒ Element
- [.clone\(element\)](#) ⇒ Element
- [.getElementById\(id\)](#) ⇒ Element
- [.getElementsByType\(shapeType, excludeType\)](#) ⇒ Array.<Element>
- [.getBBox\(element\)](#) ⇒ Object
- [.getRootBBox\(\)](#) ⇒ Object
- [.getRealRootBBox\(\)](#) ⇒ Object
- [.getContainer\(\)](#) ⇒ Element
- [.isSVG\(\)](#) ⇒ Boolean
- [.isVML\(\)](#) ⇒ Boolean
- [.getPrevEdges\(element\)](#) ⇒ Array.<Element>
- [.getNextEdges\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapes\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapeIds\(element\)](#) ⇒ Array.<String>
- [.getNextShapes\(element\)](#) ⇒ Array.<Element>
- [.getNextShapeIds\(element\)](#) ⇒ Array.<String>
- [.getConnectGuideElements\(Element\)](#) ⇒ Array
- [.isTopGroup\(Element\)](#) ⇒ boolean
- [.getParent\(element\)](#) ⇒ Element
- [.getChilds\(element\)](#) ⇒ Array
- [.isGroup\(element\)](#) ⇒ boolean

- [.getAllShapes\(\)](#) ⇒ Array
- [.getAllEdges\(\)](#) ⇒ Array
- [.getAllNotEdges\(\)](#) ⇒ Array
- [.isEdge\(\)](#) ⇒ boolean
- [.isShape\(\)](#) ⇒ boolean
- [.initHistory\(\)](#)
- [.addHistory\(\)](#)
- [.undo\(\)](#)
- [.redo\(\)](#)

new OG.renderer.IRenderer(container, containerSize, backgroundColor, backgroundImage, config)

도형의 Style 과 Shape 정보를 통해 캔버스에 렌더링 기능을 정의한 인터페이스

| Param | Type | Description |
|-----------------|----------------------|------------------------|
| container | HTMLElement String | 컨테이너 DOM element or ID |
| containerSize | Array.<Number> | 컨테이너 Width, Height |
| backgroundColor | String | 캔버스 배경색 |
| backgroundImage | String | 캔버스 배경이미지 |
| config | Object | Configuration |

iRenderer.drawShape(position, shape, size, style, id) ⇒ Element

Shape 을 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Group DOM Element with geometry

| Param | Type | Description |
|----------|--------------------------------|---------------------|
| position | Array.<Number> | 드로잉할 위치 좌표(중앙 기준) |
| shape | IShape | Shape |
| size | Array.<Number> | Shape Width, Height |
| style | Style Object | 스타일 |
| id | String | Element ID 지정 |

Example

```
renderer.drawShape([100, 100], new OG.CircleShape(), [50, 50], {stroke:'red'});
```

iRenderer.drawGeom(geometry, style) ⇒ Element

Geometry 를 캔버스에 드로잉한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Group DOM Element with geometry

| Param | Type | Description |
|----------|--------------------------------|-------------|
| geometry | Geometry | 기하 객체 |
| style | Style Object | 스타일 |

iRenderer.drawText(position, text, size, style, id) ⇒ Element

Text 를 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

(스타일 'text-anchor': 'start' or 'middle' or 'end' 에 따라 위치 기준이 다름)

Kind: instance method of [IRenderer](#)

Returns: Element - DOM Element

| Param | Type | Description |
|----------|-----------------------------------|---|
| position | Array. <Number> | 드로잉할 위치 좌표(스타일 'text-anchor': 'start' or 'middle' or 'end' 에 따라 기준이 다름) |
| text | String | 텍스트 |
| size | Array. <Number> | Text Width, Height, Angle |
| style | Style Object | 스타일 |
| id | String | Element ID 지정 |

Example

```
renderer.drawText([100, 100], 'Hello', null, {'text-anchor': 'start'});
```

iRenderer.drawImage(position, imgSrc, size, style, id) ⇒ Element

Image 를 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [IRenderer](#)

Returns: Element - DOM Element

| Param | Type | Description |
|----------|--------------------------------|----------------------------|
| position | Array.<Number> | 드로잉할 위치 좌표(좌상단 기준) |
| imgSrc | String | 이미지경로 |
| size | Array.<Number> | Image Width, Height, Angle |
| style | Style Object | 스타일 |
| id | String | Element ID 지정 |

Example

```
renderer.drawImage([100, 100], 'img.jpg', [50, 50]);
```

iRenderer.drawEdge(line, style, id, isSelf) ⇒ Element

라인을 캔버스에 드로잉한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Group DOM Element with geometry

| Param | Type | Description |
|--------|--------------------------------|---------------|
| line | Line | 라인 |
| style | Style Object | 스타일 |
| id | String | Element ID 지정 |
| isSelf | Boolean | 셀프 연결 여부 |

`iRenderer.drawLabel(shapeElement, text, style) ⇒ Element`

Shape 의 Label 을 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [IRenderer](#)

Returns: Element - DOM Element

| Param | Type | Description |
|--------------|------------------|-------------------------|
| shapeElement | Element String | Shape DOM element or ID |
| text | String | 텍스트 |
| style | Object | 스타일 |

`iRenderer.drawEdgeLabel(shapeElement, text, type) ⇒ Element`

Edge 의 from, to Label 을 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [IRenderer](#)

Returns: Element - DOM Element

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|--------------|------------------|-------------------------|
| shapeElement | Element String | Shape DOM element or ID |
| text | String | 텍스트 |
| type | String | 유형(FROM or TO) |

```
iRenderer.redrawShape(element, excludeEdgeId)
```

Element 에 저장된 geom, angle, image, text 정보로 shape 을 redraw 한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|-----------------------------|--------------------|-------------|
| element | Element | Shape 엘리먼트 |
| excludeEdgeIdArray.<String> | redraw 제외할 Edge ID | |

```
iRenderer.redrawConnectedEdge(element)
```

Shape 의 연결된 Edge 를 redraw 한다.(이동 또는 리사이즈시)

Kind: instance method of [IRenderer](#)

| Param | Type |
|---------|---------|
| element | Element |

```
iRenderer.connect(fromTerminal, toTerminal, edge, style, label, preventTrigger) ⇒ Element
```

두개의 터미널을 연결하고, 속성정보에 추가한다.

Kind: instance method of [IRenderer](#)

Returns: Element - 연결된 Edge 엘리먼트

| Param | Type | Description |
|----------------|--------------------------------|--------------------|
| fromTerminal | Element Array.<Number> | 시작점 (fromTerminal) |
| toTerminal | Element Array.<Number> | 끝점 (toTerminal) |
| edge | Element | Edge Shape |
| style | Style Object | 스타일 |
| label | String | Label |
| preventTrigger | Boolean | 이벤트 트리거 발생 막기 |

iRenderer.disconnect(element)

연속속성정보를 삭제한다. Edge 인 경우는 라인만 삭제하고, 일반 Shape 인 경우는 연결된 모든 Edge 를 삭제한다.

Kind: instance method of [IRenderer](#)

| Param | Type |
|---------|---------|
| element | Element |

iRenderer.drawDropOverGuide(element)

ID에 해당하는 Element 의 Edge 연결시 Drop Over 가이드를 드로잉한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

iRenderer.drawGuide(element) ⇒ Object

ID에 해당하는 Element 의 Move & Resize 용 가이드를 드로잉한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`iRenderer.removeGuide(element)`

ID에 해당하는 Element 의 Move & Resize 용 가이드를 제거한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`iRenderer.removeAllGuide()`

모든 Move & Resize 용 가이드를 제거한다.

Kind: instance method of [IRenderer](#)

`iRenderer.removeConnectGuide(element)`

ID에 해당하는 Element 의 Connect Guide 를 제거한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`iRenderer.removeAllConnectGuide()`

캔버스의 모든 Connect Guide 를 제거한다.

Kind: instance method of [IRenderer](#)

```
iRenderer.removeOtherConnectGuide(element)
```

ID에 해당하는 Element 이외의 모든 Connect Guide 를 제거한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

```
iRenderer.removeAllVirtualEdge()
```

캔버스의 가상선을 삭제한다.

Kind: instance method of [IRenderer](#)

```
iRenderer.drawEdgeGuide(element) ⇒ Object
```

ID에 해당하는 Edge Element 의 Move & Resize 용 가이드를 드로잉한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

iRenderer.drawRubberBand(position, size, style) ⇒ Element

Rectangle 모양의 마우스 드래그 선택 박스 영역을 드로잉한다.

Kind: instance method of [IRenderer](#)

Returns: Element - DOM Element

| Param | Type | Description |
|----------|--------------------------------|---------------------------|
| position | Array.<Number> | 드로잉할 위치 좌표(좌상단) |
| size | Array.<Number> | Text Width, Height, Angle |
| style | Style Object | 스타일 |

iRenderer.removeRubberBand(root)

Rectangle 모양의 마우스 드래그 선택 박스 영역을 제거한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|-------|---------|--------------------------------|
| root | Element | first, rubberBand 정보를 저장한 엘리먼트 |

iRenderer.drawDraggableGuide(element) ⇒ Element

ID에 해당하는 Element 의 Draggable 가이드를 드로잉한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

iRenderer.drawCollapseGuide(element) ⇒ Element

ID에 해당하는 Element 의 Collapse 가이드를 드로잉한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`iRenderer.removeCollapseGuide(element)`

ID에 해당하는 Element 의 Collapse 가이드를 제거한다.

Kind: instance method of [IRenderer](#)

| Param | Type |
|---------|---------|
| element | Element |

`iRenderer.group(elements) ⇒ Element`

주어진 Shape 들을 그룹핑한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Group Shape Element

| Param | Type |
|----------|-----------------|
| elements | Array.<Element> |

`iRenderer.ungroup(groupElements) ⇒ Array.<Element>`

주어진 그룹들을 그룹해제한다.

Kind: instance method of [IRenderer](#)

Returns: Array.<Element> - ungrouped Elements

| Param | Type |
|---------------|-----------------|
| groupElements | Array.<Element> |

```
iRenderer.addToGroup(groupElement, elements)
```

주어진 Shape 들을 그룹에 추가한다.

Kind: instance method of [IRenderer](#)

| Param | Type |
|--------------|-----------------|
| groupElement | Element |
| elements | Array.<Element> |

```
iRenderer.collapse(element)
```

주어진 Shape 이 그룹인 경우 collapse 한다.

Kind: instance method of [IRenderer](#)

| Param | Type |
|---------|---------|
| element | Element |

```
iRenderer.expand(element)
```

주어진 Shape 이 그룹인 경우 expand 한다.

Kind: instance method of [IRenderer](#)

| Param | Type |
|---------|---------|
| element | Element |

`iRenderer.clear()`

드로잉된 모든 오브젝트를 클리어한다.

Kind: instance method of [IRenderer](#)

`iRenderer.removeShape(element)`

Shape 을 캔버스에서 관련된 모두를 삭제한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`iRenderer.remove(element)`

ID에 해당하는 Element 를 캔버스에서 제거한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`iRenderer.removeChild(element)`

하위 엘리먼트만 제거한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`iRenderer.getRootElement()` ⇒ Element

랜더러 캔버스 Root Element 를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Element

`iRenderer.getRootGroup()` ⇒ Element

랜더러 캔버스 Root Group Element 를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Element

`iRenderer.getElementByPoint(position)` ⇒ Element

주어진 지점을 포함하는 Top Element 를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Element

| Param | Type | Description |
|---------------|----------|-------------|
| positionArray | <Number> | 위치 좌표 |

iRenderer.getElementsByBBox(envelope) ⇒ Array.<Element>

주어진 Boundary Box 영역에 포함되는 Shape(GEOM, TEXT, IMAGE) Element 를 반환한다.

모든 vertices를 포함한 엘리먼트를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Array.<Element> - Element

| Param | Type | Description |
|----------|--------------------------|-----------------|
| envelope | Envelope | Boundary Box 영역 |

iRenderer.setAttr(element, attribute)

엘리먼트에 속성값을 설정한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|-----------------|------------------|---------------|
| element | Element String | Element 또는 ID |
| attributeObject | | 속성값 |

iRenderer.getAttr(element, attrName) ⇒ Object

엘리먼트 속성값을 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Object - attribute 속성값

| Param | Type | Description |
|----------|------------------|---------------|
| element | Element String | Element 또는 ID |
| attrName | String | 속성이름 |

```
iRenderer.setShapeStyle(element, style)
```

Shape 의 스타일을 변경한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |
| style | Object | 스타일 |

```
iRenderer.toFront(element)
```

ID에 해당하는 Element 를 최상단 레이어로 이동한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

```
iRenderer.toBack(element)
```

ID에 해당하는 Element 를 최하단 레이어로 이동한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`iRenderer.bringForward(element)`

ID에 해당하는 Element 를 앞으로 한단계 이동한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`iRenderer.sendBackward(element)`

ID에 해당하는 Element 를 뒤로 한단계 이동한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`iRenderer.getCanvasSize()` ⇒ Array.<Number>

랜더러 캔버스의 사이즈(Width, Height)를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Array.<Number> - Canvas Width, Height

iRenderer.setCanvasSize(size)

랜더러 캔버스의 사이즈(Width, Height)를 변경한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|-------|----------------|----------------------|
| size | Array.<Number> | Canvas Width, Height |

iRenderer.fitCanvasSize(minSize, fitScale)

랜더러 캔버스의 사이즈(Width, Height)를 실제 존재하는 Shape 의 영역에 맞게 변경한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|----------|----------------|--|
| minSize | Array.<Number> | Canvas 최소 Width, Height |
| fitScale | Boolean | 주어진 minSize 에 맞게 fit 여부(Default:false) |

iRenderer.setViewBox(position, size, isFit)

새로운 View Box 영역을 설정한다. (ZoomIn & ZoomOut 가능)

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|----------|----------------|----------------------|
| position | Array.<Number> | 위치 좌표(좌상단 기준) |
| size | Array.<Number> | Canvas Width, Height |
| isFit | Boolean | Fit 여부 |

iRenderer.getScale() ⇒ Number

Scale 을 반환한다. (리얼 사이즈 : Scale = 1)

Kind: instance method of [IRenderer](#)

Returns: Number - 스케일값

iRenderer.setScale(scale)

Scale 을 설정한다. (리얼 사이즈 : Scale = 1)

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|-------|--------|-------------|
| scale | Number | 스케일값 |

iRenderer.show(element)

ID에 해당하는 Element 를 캔버스에서 show 한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

iRenderer.hide(element)

ID에 해당하는 Element 를 캔버스에서 hide 한다.

Kind: instance method of [IRenderer](#)

| Param | Type | Description |
|---------|-------------------------|-------------|
| element | Element StringElement | 또는 ID |

`iRenderer.appendChild(srcElement, targetElement) ⇒ Element`

Source Element 를 Target Element 아래에 append 한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Source Element

| Param | Type | Description |
|---------------|-------------------------|-------------|
| srcElement | Element StringElement | 또는 ID |
| targetElement | Element StringElement | 또는 ID |

`iRenderer.insertAfter(srcElement, targetElement) ⇒ Element`

Source Element 를 Target Element 이후에 insert 한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Source Element

| Param | Type | Description |
|---------------|-------------------------|-------------|
| srcElement | Element StringElement | 또는 ID |
| targetElement | Element StringElement | 또는 ID |

`iRenderer.insertBefore(srcElement, targetElement) ⇒ Element`

Source Element 를 Target Element 이전에 insert 한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Source Element

| Param | Type | Description |
|---------------|-------------------------------|-------------|
| srcElement | Element StringElement 또는 ID | |
| targetElement | Element StringElement 또는 ID | |

iRenderer.move(element, offset) ⇒ Element

해당 Element 를 가로, 세로 Offset 만큼 이동한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Element

| Param | Type | Description |
|---------|-------------------------------|-------------|
| element | Element StringElement 또는 ID | |
| offset | Array.<Number> | [가로, 세로] |

iRenderer.moveCentroid(element, position) ⇒ Element

주어진 중심좌표로 해당 Element 를 이동한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Element

| Param | Type | Description |
|----------|-------------------------------|-------------|
| element | Element StringElement 또는 ID | |
| position | Array.<Number> | [x, y] |

`iRenderer.rotate(element, angle) ⇒ Element`

중심 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |
| angle | Number | 각도 |

`iRenderer.resize(element, offset) ⇒ Element`

상, 하, 좌, 우 외곽선을 이동한 만큼 리사이즈 한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|-------------------------|
| element | Element String | Element 또는 ID |
| offset | Array.<Number> | [상, 하, 좌, 우] 각 방향으로 + 값 |

`iRenderer.resizeBox(element, size) ⇒ Element`

중심좌표는 고정한 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

size Array.<Number> [Width, Height]

iRenderer.clone(element) ⇒ Element

노드 Element 를 복사한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

iRenderer.getElementById(id) ⇒ Element

ID로 Node Element 를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Element - Element

| Param | Type |
|-------|--------|
| id | String |

iRenderer.getElementsByType(shapeType, excludeType) ⇒ Array.<Element>

Shape 타입에 해당하는 Node Element 들을 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Array.<Element> - Element's Array

| Param | Type | Description |
|-------------|--------|---|
| shapeType | String | Shape 타입(GEOM, HTML, IMAGE, EDGE, GROUP), Null 이면 모든 타입 |
| excludeType | String | 제외 할 타입 |

iRenderer.getBBox(element) ⇒ Object

해당 엘리먼트의 BoundingBox 영역 정보를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Object - {width, height, x, y, x2, y2}

| Param | Type |
|---------|------------------|
| element | Element String |

iRenderer.getRootBBox() ⇒ Object

부모노드기준으로 캔버스 루트 엘리먼트의 BoundingBox 영역 정보를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Object - {width, height, x, y, x2, y2}

iRenderer.getRealRootBBox() ⇒ Object

부모노드기준으로 캔버스 루트 엘리먼트의 실제 Shape 이 차지하는 BoundingBox 영역 정보를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Object - {width, height, x, y, x2, y2}

iRenderer.getContainer() ⇒ Element

캔버스의 컨테이너 DOM element 를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Element - 컨테이너

iRenderer.isSVG() ⇒ Boolean

SVG 인지 여부를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Boolean - svg 여부

iRenderer.isVML() ⇒ Boolean

VML 인지 여부를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Boolean - vml 여부

iRenderer.getPrevEdges(element) ⇒ Array.<Element>

연결된 이전 Edge Element 들을 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Array.<Element> - Previous Element's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

iRenderer.getNextEdges(element) ⇒ Array.<Element>

연결된 이후 Edge Element 들을 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Array.<Element> - Previous Element's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

iRenderer.getPrevShapes(element) ⇒ Array.<Element>

연결된 이전 노드 Element 들을 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Array.<Element> - Previous Element's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

iRenderer.getPrevShapeIds(element) ⇒ Array.<String>

연결된 이전 노드 Element ID들을 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Array.<String> - Previous Element Id's Array

| Param | Type | Description |
|---------|-------------------------|-------------|
| element | Element StringElement | 또는 ID |

iRenderer.getNextShapes(element) ⇒ Array.<Element>

연결된 이후 노드 Element 들을 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Array.<Element> - Previous Element's Array

| Param | Type | Description |
|---------|-------------------------|-------------|
| element | Element StringElement | 또는 ID |

iRenderer.getNextShapeIds(element) ⇒ Array.<String>

연결된 이후 노드 Element ID들을 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Array.<String> - Previous Element Id's Array

| Param | Type | Description |
|---------|-------------------------|-------------|
| element | Element StringElement | 또는 ID |

iRenderer.getConnectGuideElements(Element) ⇒ Array

Node 엘리먼트의 커넥트 가이드 엘리먼트를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Array - Array Element

| Param | Type | Description |
|---------|---------|-------------|
| Element | Element | 엘리먼트 |

`iRenderer.isTopGroup(Element) ⇒ boolean`

최상위 그룹 엘리먼트인지 반환한다.

Kind: instance method of [IRenderer](#)

Returns: boolean - true false

| Param | Type | Description |
|---------|---------|-------------|
| Element | Element | 엘리먼트 |

`iRenderer.getParent(element) ⇒ Element`

부모 엘리먼트를 반환한다. 부모가 루트일때는 반환하지 않는다.

Kind: instance method of [IRenderer](#)

Returns: Element - element 엘리먼트

| Param | Type | Description |
|---------|---------|-------------|
| element | Element | 엘리먼트 |

`iRenderer.getChilds(element) ⇒ Array`

그룹의 하위 엘리먼트를 반환한다.

Kind: instance method of [IRenderer](#)

Returns: Array - Elements

| Param | Type | Description |
|---------|---------|-------------|
| element | Element | 엘리먼트 |

`iRenderer.isGroup(element) ⇒ boolean`

그룹의 Shape 인지 반환한다. RootGroup 일 경우는 제외.

Kind: instance method of [IRenderer](#)

Returns: boolean - true false

| Param | Type | Description |
|---------|---------|-------------|
| element | Element | 엘리먼트 |

`iRenderer.getAllShapes() ⇒ Array`

캔버스의 모든 Shape 들을 리턴

Kind: instance method of [IRenderer](#)

Returns: Array - Elements

`iRenderer.getAllEdges() ⇒ Array`

캔버스의 모든 Edge를 리턴

Kind: instance method of [IRenderer](#)

Returns: Array - Edge Elements

`iRenderer.getAllNotEdges() ⇒ Array`

캔버스의 모든 Edge 가 아닌 shpaes 를 리턴

Kind: instance method of [IRenderer](#)

Returns: Array - Edge Elements

`iRenderer.isEdge() ⇒ boolean`

Edge 여부를 판단.

Kind: instance method of [IRenderer](#)

Returns: boolean - true false

`iRenderer.isShape() ⇒ boolean`

Shape 여부를 판단.

Kind: instance method of [IRenderer](#)

Returns: boolean - true false

`iRenderer.initHistory()`

캔버스의 히스토리를 초기화한다.

Kind: instance method of [IRenderer](#)

`iRenderer.addHistory()`

캔버스에 히스토리를 추가한다.

Kind: instance method of [IRenderer](#)

`iRenderer.undo()`

캔버스의 Undo

Kind: instance method of [IRenderer](#)

`iRenderer.redo()`

캔버스의 Redo

Kind: instance method of [IRenderer](#)

`renderer.RaphaelRenderer` \Leftarrow [IRenderer](#)

Kind: static class of [renderer](#)

Extends: [IRenderer](#)

Requires: module:OG.common.*, module:OG.geometry.*, module:OG.shape.*,

module:raphael-2.1.0

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.RaphaelRenderer](#) \Leftarrow [IRenderer](#)
 - [new OG.renderer.RaphaelRenderer\(container, containerSize, backgroundColor, backgroundImage, config\)](#)
 - [.drawHtml\(position, html, size, style, id\)](#) \Rightarrow Element
 - [._getPointOfInflectionFromEdge\(\)](#)
 - [.reconnect\(edge\)](#) \Rightarrow Element
 - [.disconnectOneWay\(element, connectDirection\)](#)
 - [.drawStickGuide\(element, position\)](#)
 - [.setTextListInController\(element, textList\)](#)
 - [.getTextListInController\(element\)](#)
 - [.getConnectGuideElements\(Element\)](#) \Rightarrow Array
 - [.getNotConnectGuideElements\(Element\)](#) \Rightarrow Array
 - [.removeConnectGuide\(element\)](#)
 - [.removeAllConnectGuide\(\)](#)
 - [.removeOtherConnectGuide\(element\)](#)
 - [.getSpots\(element\)](#) \Rightarrow Array
 - [.getCircleSpots\(element\)](#) \Rightarrow Array
 - [.createVirtualSpot\(x, x, element\)](#) \Rightarrow Element
 - [.getVirtualSpot\(element\)](#) \Rightarrow Element
 - [.removeVirtualSpot\(element\)](#) \Rightarrow Element
 - [.selectSpot\(선택한\)](#)
 - [.getChildNodes\(element\)](#) \Rightarrow Array
 - [.trimEdge\(element\)](#)

- [.trimConnectInnerVertice\(element\)](#) ⇒ Element
- [.trimConnectIntersection\(element\)](#) ⇒ Element
- [.getBoundary\(element\)](#) ⇒ [Envelope](#)
- [.setHighlight\(element, highlight\)](#)
- [.removeHighlight\(element, highlight\)](#)
- [.createTerminalString\(Element, point\)](#) ⇒ String
- [.createDefaultTerminalString\(Element\)](#) ⇒ String
- [.toFrontEdges\(\)](#)
- [.removeAllEdgeGuide\(\)](#)
- [.createVirtualEdge\(x, x, targetEle\)](#) ⇒ Element
- [.updateVirtualEdge\(x, x\)](#)
- [.getTargetfromVirtualEdge\(x, x\)](#)
- [.removeAllVirtualEdge\(\)](#)
- [.isLane\(Element\)](#) ⇒ boolean
- [.isPool\(Element\)](#) ⇒ boolean
- [.isScopeActivity\(Element\)](#) ⇒ boolean
- [.isHorizontalLane\(Element\)](#) ⇒ boolean
- [.isVerticalLane\(Element\)](#) ⇒ boolean
- [.isHorizontalPool\(Element\)](#) ⇒ boolean
- [.isVerticalPool\(Element\)](#) ⇒ boolean
- [.getChildLane\(Element\)](#) ⇒ Array
- [.enableDivideCount\(Element\)](#) ⇒ Number
- [.getExceptTitleLaneArea\(Element, boundary\)](#)
- [.divideLane\(Element, quarterOrder\)](#)
- [.getBaseLanes\(Element\)](#) ⇒ Array
- [.getRootLane\(Element\)](#) ⇒ Element

- [.getIndexOfLane\(Element\)](#) ⇒ Number
- [.getDepthOfLane\(Element\)](#) ⇒ Number
- [.reEstablishLane\(Element\)](#)
- [.getBoundaryOfElements\(elements\)](#) ⇒ [Envelope](#)
- [.getNearestBaseLaneIndexAsDirection\(Element, direction\)](#) ⇒ Number
- [.getBoundaryOfInnerShapesGroup\(Element\)](#) ⇒ [Envelope](#)
- [.getSmallestBaseLane\(Element, baseLane\)](#)
- [.resizeLane\(Element, offset\)](#)
- [.removeLaneShape\(Element\)](#)
- [.getInnerShapesOfLane\(Element\)](#)
- [.fitLaneOrder\(Element\)](#)
- [.getRootGroupOfShape\(Element\)](#) ⇒ Element
- [.checkBridgeEdge\(Element\)](#)
- [.checkAllBridgeEdge\(\)](#)
- [.getInnerShapesOfGroup\(Element\)](#)
- [.getFrontForCoordinate\(point\)](#) ⇒ Element
- [.getFrontForBoundary\(boundary\)](#) ⇒ Element
- [.trimEdgeDirection\(Edge, FromShape, ToShape\)](#) ⇒ Element
- [.putInnerShapeToPool\(Element\)](#) ⇒ Element
- [.setDropablePool\(Element\)](#) ⇒ Element
- [.offDropablePool\(\)](#)
- [.drawShape\(position, shape, size, style, id\)](#) ⇒ Element
- [.drawGeom\(geometry, style\)](#) ⇒ Element
- [.drawText\(position, text, size, style, id\)](#) ⇒ Element
- [.drawImage\(position, imgSrc, size, style, id\)](#) ⇒ Element
- [.drawEdge\(line, style, id, isSelf\)](#) ⇒ Element

- [.drawLabel\(shapeElement, text, style\)](#) ⇒ Element
 - [~getCenterOfEdge\(element\)](#) ⇒ OG.Coordinate
- [.drawEdgeLabel\(shapeElement, text, type\)](#) ⇒ Element
- [.redrawShape\(element, excludeEdgeId\)](#)
- [.redrawConnectedEdge\(element\)](#)
- [.connect\(fromTerminal, toTerminal, edge, style, label, preventTrigger\)](#) ⇒ Element
- [.disconnect\(element\)](#)
- [.drawDropOverGuide\(element\)](#)
- [.drawGuide\(element\)](#) ⇒ Object
- [.removeGuide\(element\)](#)
- [.removeAllGuide\(\)](#)
- [.drawEdgeGuide\(element\)](#) ⇒ Object
- [.drawRubberBand\(position, size, style\)](#) ⇒ Element
- [.removeRubberBand\(root\)](#)
- [.drawDraggableGuide\(element\)](#) ⇒ Element
- [.drawCollapseGuide\(element\)](#) ⇒ Element
- [.removeCollapseGuide\(element\)](#)
- [.group\(elements\)](#) ⇒ Element
- [.ungroup\(groupElements\)](#) ⇒ Array.<Element>
- [.addToGroup\(groupElement, elements\)](#)
- [.collapse\(element\)](#)
- [.expand\(element\)](#)
- [.clear\(\)](#)
- [.removeShape\(element\)](#)

- [.remove\(element\)](#)
- [.removeChild\(element\)](#)
- [.getRootElement\(\)](#) ⇒ Element
- [.getRootGroup\(\)](#) ⇒ Element
- [.getElementByPoint\(position\)](#) ⇒ Element
- [.getElementsByBBox\(envelope\)](#) ⇒ Array.<Element>
- [.setAttr\(element, attribute\)](#)
- [.getAttr\(element, attrName\)](#) ⇒ Object
- [.setShapeStyle\(element, style\)](#)
- [.toFront\(element\)](#)
- [.toBack\(element\)](#)
- [.bringForward\(element\)](#)
- [.sendBackward\(element\)](#)
- [.getCanvasSize\(\)](#) ⇒ Array.<Number>
- [.setCanvasSize\(size\)](#)
- [.fitCanvasSize\(minSize, fitScale\)](#)
- [.setViewBox\(position, size, isFit\)](#)
- [.getScale\(\)](#) ⇒ Number
- [.setScale\(scale\)](#)
- [.show\(element\)](#)
- [.hide\(element\)](#)
- [.appendChild\(srcElement, targetElement\)](#) ⇒ Element
- [.insertAfter\(srcElement, targetElement\)](#) ⇒ Element
- [.insertBefore\(srcElement, targetElement\)](#) ⇒ Element
- [.move\(element, offset\)](#) ⇒ Element
- [.moveCentroid\(element, position\)](#) ⇒ Element

- [.rotate\(element, angle\)](#) ⇒ Element
- [.resize\(element, offset\)](#) ⇒ Element
- [.resizeBox\(element, size\)](#) ⇒ Element
- [.clone\(element\)](#) ⇒ Element
- [.getElementById\(id\)](#) ⇒ Element
- [.getElementsByType\(shapeType, excludeType\)](#) ⇒ Array.<Element>
- [.getBBox\(element\)](#) ⇒ Object
- [.getRootBBox\(\)](#) ⇒ Object
- [.getRealRootBBox\(\)](#) ⇒ Object
- [.getContainer\(\)](#) ⇒ Element
- [.isSVG\(\)](#) ⇒ Boolean
- [.isVML\(\)](#) ⇒ Boolean
- [.getPrevEdges\(element\)](#) ⇒ Array.<Element>
- [.getNextEdges\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapes\(element\)](#) ⇒ Array.<Element>
- [.getPrevShapeIds\(element\)](#) ⇒ Array.<String>
- [.getNextShapes\(element\)](#) ⇒ Array.<Element>
- [.getNextShapeIds\(element\)](#) ⇒ Array.<String>
- [.isTopGroup\(Element\)](#) ⇒ boolean
- [.getParent\(element\)](#) ⇒ Element
- [.getChilds\(element\)](#) ⇒ Array
- [.isGroup\(element\)](#) ⇒ boolean
- [.getAllShapes\(\)](#) ⇒ Array
- [.getAllEdges\(\)](#) ⇒ Array
- [.getAllNotEdges\(\)](#) ⇒ Array
- [.isEdge\(\)](#) ⇒ boolean

- [.isShape\(\)](#) ⇒ boolean
- [.initHistory\(\)](#)
- [.addHistory\(\)](#)
- [.undo\(\)](#)
- [.redo\(\)](#)

```
new OG.renderer.RaphaelRenderer(container, containerSize, backgroundColor, backgroundImage, config)
```

Raphael 라이브러리를 이용하여 구현한 랜더러 캔버스 클래스

- 노드에 추가되는 속성 : `_type`, `_shape`, `_selected`, `_from`, `_to`, `_fromedge`, `_toedge`
- 노드에 저장되는 값 : `shape` : { `geom`, `angle`, `image`, `text` }, `data` : 커스텀 Object

| Param | Type | Description |
|-----------------|----------------------|------------------------|
| container | HTMLElement String | 컨테이너 DOM element or ID |
| containerSize | Array.<Number> | 컨테이너 Width, Height |
| backgroundColor | String | 캔버스 배경색 |
| backgroundImage | String | 캔버스 배경이미지 |
| config | Object | Configuration |

```
raphaelRenderer.drawHtml(position, html, size, style, id) ⇒ Element
```

임베드 HTML String 을 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - DOM Element

| Param | Type | Description |
|---------------|--------------------------------|----------------------------|
| positionArray | <Number> | 드로잉할 위치 좌표(중앙 기준) |
| html | String | 임베드 HTML String |
| size | Array.<Number> | Image Width, Height, Angle |
| style | Style Object | 스타일 |
| id | String | Element ID 지정 |

`raphaelRenderer._getPointOfInflectionFromEdge()`

사용자가 지정한 변곡점을 반환

- 조건:
 - 오직 4개의 점을 가지고있는 다각선에서
 - 변곡점이 자동으로 그려진 점의 위치가 아닐 때
- issue: 사용자가 지정한 변곡점(?)은 유지되어야 함

Kind: instance method of [RaphaelRenderer](#)

`raphaelRenderer.reconnect(edge) ⇒ Element`

연결된 터미널의 vertices 를 초기화하여 재연결한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - 연결된 Edge 엘리먼트

| Param | Type | Description |
|-------|---------|-------------|
| edge | Element | Edge Shape |

```
raphaelRenderer.disconnectOneWay(element, connectDirection)
```

단방향 연결속성정보를 삭제한다. Edge 인 경우에만 해당한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|------------------|---------|---------------------|
| element | Element | |
| connectDirection | String | 연결방향 'from' or 'to' |

```
raphaelRenderer.drawStickGuide(element, position)
```

ID에 해당하는 Element 의 Stick 용 가이드를 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|----------|------------------|---------------|
| element | Element String | Element 또는 ID |
| position | Object | |

```
raphaelRenderer.setTextListInController(element, textList)
```

Shape 의 선 연결 커스텀 컨트롤러를 설정한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

element Element | StringElement 또는 ID
textListArray 텍스트 리스트

raphaelRenderer.getTextListInController(element)

Shape 의 선 연결 커스텀 컨트롤러를 가져온다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|---------|-------------------------------|-------------|
| element | Element StringElement 또는 ID | |

raphaelRenderer.getConnectGuideElements(Element) ⇒ Array

Node 엘리먼트의 커넥트 가이드 영역 엘리먼트를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [getConnectGuideElements](#)

Returns: Array - Array Element

| Param | Type | Description |
|---------|--------------|-------------|
| Element | Element 엘리먼트 | |

raphaelRenderer.getNotConnectGuideElements(Element) ⇒ Array

Node 엘리먼트의 커넥트 가이드 영역을 제외한 엘리먼트를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array - Array Element

| Param | Type | Description |
|---------|---------|-------------|
| Element | Element | 엘리먼트 |

```
raphaelRenderer.removeConnectGuide(element)
```

ID에 해당하는 Element 의 Connect Guide 를 제거한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [removeConnectGuide](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

```
raphaelRenderer.removeAllConnectGuide()
```

캔버스의 모든 Connect Guide 를 제거한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [removeAllConnectGuide](#)

```
raphaelRenderer.removeOtherConnectGuide(element)
```

ID에 해당하는 Element 이외의 모든 Connect Guide 를 제거한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [removeOtherConnectGuide](#)

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

elementElement | StringElement 또는 ID

raphaelRenderer.getSpots(element) ⇒ Array

Element 내부의 Spot 들을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array - Spot Element Array

| Param | Type | Description |
|---------|-------------------------------|-------------|
| element | Element StringElement 또는 ID | |

raphaelRenderer.getCircleSpots(element) ⇒ Array

Element 내부의 변곡점 Spot 들만 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array - Spot Element Array

| Param | Type | Description |
|---------|-------------------------------|-------------|
| element | Element StringElement 또는 ID | |

raphaelRenderer.createVirtualSpot(x, x, element) ⇒ Element

주어진 좌표와 가장 Edge Element의 가장 가까운 거리에 가상 변곡점 스폿을 생성한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Spot Element

| Param | Type | Description |
|---------|------------------|------------------|
| x | Number | 이벤트의 캔버스 기준 x 좌표 |
| y | Number | 이벤트의 캔버스 기준 y 좌표 |
| element | Element String | Element 또는 ID |

`raphaelRenderer.getVirtualSpot(element)` ⇒ Element

Element 내부의 가상 변곡점 스팟을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Spot Element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.removeVirtualSpot(element)` ⇒ Element

Element 내부의 가상 변곡점 스팟을 삭제한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Spot Element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.selectSpot(선택한)`

Element 내부의 Spot 중 선택한 스팟을 제외하고 모두 삭제하고, 가이드라인도 삭제한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|-------|---------|--------------------------|
| 선택한 | Element | Stringspot Element 또는 ID |

`raphaelRenderer.getChildNodes(element)` ⇒ Array

하위 엘리먼트들을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array - Array Element

| Param | Type | Description |
|---------|---------|---------------------|
| element | Element | StringElement 또는 ID |

`raphaelRenderer.trimEdge(element)`

Edge Element 내부의 패스중 나열된 두 꼭지점이 매우 짧은 선일 경우 하나의 꼭지점으로 정리한다.

Edge Element 내부의 패스중 나열된 세 꼭지점이 평행에 가까울 경우 하나의 선분으로 정리한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|---------|---------|---------------------|
| element | Element | StringElement 또는 ID |

`raphaelRenderer.trimConnectInnerVertice(element)` ⇒ Element

Edge Element의 연결 정보가 있을 경우 연결대상과 꼭지점의 다중 중복을 정리한다.

다중 중복 정리 후 Edge 의 모양이 직선인 경우 새로운 plain 을 제작한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.trimConnectIntersection(element) ⇒ Element`

Edge Element의 연결 정보가 있을 경우 선분과 연결대상의 연결점을 자연스럽게 한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.getBoundary(element) ⇒ Envelope`

ID에 해당하는 Element 의 바운더리 영역을 리턴한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: [Envelope](#) - Envelope 영역

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.setHighlight(element, highlight)`

Element 에 하이라이트 속성을 부여한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|-----------------|-------------------------|------------------|
| element | Element StringElement | 또는 ID |
| highlightObject | | HIGHLIGHT 속성 집합. |

`raphaelRenderer.removeHighlight(element, highlight)`

Element 에 하이라이트 속성을 제거한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|-----------------|-------------------------|------------------|
| element | Element StringElement | 또는 ID |
| highlightObject | | HIGHLIGHT 속성 집합. |

`raphaelRenderer.createTerminalString(Element, point) ⇒ String`

터미널 문자열을 생성한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: String - terminal 터미널 문자열

| Param | Type | Description |
|---------|-------------------------|---------------|
| Element | Element StringElement | 또는 ID |
| point | Array | 연결 좌표정보 [x,y] |

`raphaelRenderer.createDefaultTerminalString(Element) ⇒ String`

디폴트 터미널 문자열을 생성한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: String - terminal 터미널 문자열

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.toFrontEdges()`

캔버스의 Edge 들을 항상 최상단으로 이동시킨다.

Kind: instance method of [RaphaelRenderer](#)

`raphaelRenderer.removeAllEdgeGuide()`

캔버스의 Edge 들의 가이드를 제거한다.

Kind: instance method of [RaphaelRenderer](#)

`raphaelRenderer.createVirtualEdge(x, x, targetEle) ⇒ Element`

주어진 좌표와 선택된 Element 사이에 가상 연결선을 생성한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Edge Element

| Param | Type | Description |
|-----------|------------------|------------------|
| x | Number | 이벤트의 캔버스 기준 x 좌표 |
| y | Number | 이벤트의 캔버스 기준 y 좌표 |
| targetEle | Element String | Element 또는 ID |

`raphaelRenderer.updateVirtualEdge(x, x)`

캔버스의 가상 연결선을 업데이트한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|-------|--------|------------------|
| x | Number | 이벤트의 캔버스 기준 x 좌표 |
| x | Number | 이벤트의 캔버스 기준 y 좌표 |

`raphaelRenderer.getTargetfromVirtualEdge(x, x)`

캔버스의 가상선의 타겟 엘리먼트를 구한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|-------|--------|------------------|
| x | Number | 이벤트의 캔버스 기준 x 좌표 |
| x | Number | 이벤트의 캔버스 기준 y 좌표 |

`raphaelRenderer.removeAllVirtualEdge()`

캔버스의 가상선을 삭제한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [removeAllVirtualEdge](#)

`raphaelRenderer.isLane(Element) ⇒ boolean`

도형의 Lane 타입 여부를 판별한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: boolean - true false

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.isPool(Element) ⇒ boolean`

도형의 Pool 타입 여부를 판별한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: boolean - true false

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.isScopeActivity(Element) ⇒ boolean`

도형의 ScopeActivity 타입 여부를 판별한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: boolean - true false

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.isHorizontalLane(Element) ⇒ boolean`

도형의 HorizontalLaneShape 타입 여부를 판별한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: boolean - true false

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

raphaelRenderer.isVerticalLane(Element) ⇒ boolean

도형의 VerticalLaneShape 타입 여부를 판별한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: boolean - true false

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

raphaelRenderer.isHorizontalPool(Element) ⇒ boolean

도형의 HorizontalPoolShape 타입 여부를 판별한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: boolean - true false

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

raphaelRenderer.isVerticalPool(Element) ⇒ boolean

도형의 VerticalPoolShape 타입 여부를 판별한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: boolean - true false

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.getChildLane(Element) ⇒ Array`

Lane 타입 도형 하위의 Lane 타입들을 리턴한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array - childsLanes

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.enableDivideCount(Element) ⇒ Number`

Lane 타입이 내부적으로 분기가 가능한 수를 리턴한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Number - 0,1,2

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.getExceptTitleLaneArea(Element, boundary)`

Lane, Pool 의 타이틀 영역을 제외한 boundary 를 리턴한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|---------|------------------|---|
| Element | Element String | Element 또는 ID boundary Envelope |

`raphaelRenderer.divideLane(Element, quarterOrder)`

Lane 을 분기한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|--------------|------------------|---|
| Element | Element String | Element 또는 ID |
| quarterOrder | String | 분기 명령 QUARTER_UPPER QUARTER_LOW QUARTER_BISECTOR QUARTER_THIRDS |

`raphaelRenderer.getBaseLanes(Element) ⇒ Array`

Lane 의 최상의 Lane 으로부터 모든 Base Lane 들을 반환한다.

Base Lane 은 자식 Lane 을 가지지 않는 Lane 을 뜻함.

반환하는 Array 는 좌표상의 값을 기준으로 정렬되어 있는 상태이다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array - childBaseLanes

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.getRootLane(Element) ⇒ Element`

Lane 의 최상위 Lane 을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Lane Element

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.getIndexOfLane(Element) ⇒ Number`

Lane 의 BaseLane 으로부터 자신의 순서를 구한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Number - index

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.getDepthOfLane(Element) ⇒ Number`

Lane 의 최상위 Lane 으로부터 Depth를 구한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Number - depth

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

ElementElement | StringElement 또는 ID

raphaelRenderer.reEstablishLane(Element)

Lane 의 BaseLane 영역을 기준으로 전체 Lane 의 구조를 재정립한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|-------|--------------------------------------|-------------|
| | ElementElement StringElement 또는 ID | |

raphaelRenderer.getBoundaryOfElements(elements) ⇒ [Envelope](#)

주어진 Shape 들의 바운더리 영역을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: [Envelope](#) - Envelope 영역

| Param | Type |
|----------|-----------------|
| elements | Array.<Element> |

raphaelRenderer.getNearestBaseLaneIndexAsDirection(Element, direction) ⇒ Number

Lane 의 baseLane 들 중

Lane의 주어진 direction 과 BaseLane 의 주어진 direction 이 가장 가까운 BaseLane 의 인덱스를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Number - index

| Param | Type | Description |
|-----------|------------------|---------------|
| Element | Element String | Element 또는 ID |
| direction | String | |

`raphaelRenderer.getBoundaryOfInnerShapesGroup(Element)` ⇒ [Envelope](#)

Group 의 내부 도형들의 Boundary를 반환한다.

Lane 이면 최상위 Lane의 내부 도형들의 Boundary를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: [Envelope](#) - Envelope 영역

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.getSmallestBaseLane(Element, baseLane)`

Lane 의 BaseLane 중 길이가 가장 작은 Lane 을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|----------|------------------|---------------|
| Element | Element String | Element 또는 ID |
| baseLane | Element | |

`raphaelRenderer.resizeLane(Element, offset)`

Lane 을 리사이즈한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|---------|------------------|-------------------------|
| Element | Element String | Element 또는 ID |
| offset | Array.<Number> | [상, 하, 좌, 우] 각 방향으로 + 값 |

`raphaelRenderer.removeLaneShape(Element)`

Lane 을 삭제한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.getInnerShapesOfLane(Element)`

Lane 내부 도형들을 구한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.fitLaneOrder(Element)`

Lane 의 내부 도형들을 앞으로 이동시킨다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

ElementElement | StringElement 또는 ID

raphaelRenderer.getRootGroupOfShape(Element) ⇒ Element

Shape 가 소속된 의 최상위 그룹 엘리먼트를 반환한다.

그룹이 소속이 아닌 엘리먼트는 자신을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|-------------|
| Element | Element String | 또는 ID |

raphaelRenderer.checkBridgeEdge(Element)

Edge 가 Gourp 사이를 넘어가는 경우 스타일에 변화를 준다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|---------|------------------|-------------|
| Element | Element String | 또는 ID |

raphaelRenderer.checkAllBridgeEdge()

모든 Edge 를 checkBridgeEdge

Kind: instance method of [RaphaelRenderer](#)

`raphaelRenderer.getInnerShapesOfGroup(Element)`

Group 내부의 모든 shape 을 리턴한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| Element | Element String | Element 또는 ID |

`raphaelRenderer.getFrontForCoordinate(point) ⇒ Element`

주어진 좌표를 포함하는 Element 중 가장 Front 에 위치한 Element 를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Element

| Param | Type | Description |
|-------|----------------|-------------|
| point | Array.<Number> | 좌표값 |

`raphaelRenderer.getFrontForBoundary(boundary) ⇒ Element`

Boundary 를 포함하는 가장 Front 에 위치한 Group Element 를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Element

| Param | Type | Description |
|----------|--------------------------|-------------|
| boundary | Envelope | 영역 |

`raphaelRenderer.trimEdgeDirection(Edge, FromShape, ToShape) ⇒ Element`

신규 Edge 의 vertices 를 연결대상 도형에 따라 설정한다

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Edge Element

| Param | Type | Description |
|-----------|------------------|---------------|
| Edge | Element String | Element 또는 ID |
| FromShape | Element String | Element 또는 ID |
| ToShape | Element String | Element 또는 ID |

`raphaelRenderer.putInnerShapeToPool(Element) ⇒ Element`

Lane 또는 Pool 내부 도형들을 그룹에 포함시킨다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Element

| Param | Type |
|---------|------------------|
| Element | Element String |

`raphaelRenderer.setDropablePool(Element) ⇒ Element`

신규 Lane 또는 Pool 이 캔버스상에서 드래그하여 그려지도록 사전작업을 수행한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Element

| Param | Type |
|---------|------------------|
| Element | Element String |

`raphaelRenderer.offDropablePool()`

신규 Lane 또는 Pool 드랍모드를 해제한다.

Kind: instance method of [RaphaelRenderer](#)

`raphaelRenderer.drawShape(position, shape, size, style, id) ⇒ Element`

Shape 을 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [drawShape](#)

Returns: Element - Group DOM Element with geometry

| Param | Type | Description |
|----------|--------------------------------|---------------------|
| position | Array.<Number> | 드로잉할 위치 좌표(중앙 기준) |
| shape | IShape | Shape |
| size | Array.<Number> | Shape Width, Height |
| style | Style Object | 스타일 |
| id | String | Element ID 지정 |

Example

```
renderer.drawShape([100, 100], new OG.CircleShape(), [50, 50], {stroke:'red'});
```

`raphaelRenderer.drawGeom(geometry, style) ⇒ Element`

Geometry 를 캔버스에 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [drawGeom](#)

Returns: Element - Group DOM Element with geometry

| Param | Type | Description |
|----------|--------------------------------|-------------|
| geometry | Geometry | 기하 객체 |
| style | Style Object | 스타일 |

raphaelRenderer.drawText(position, text, size, style, id) ⇒ Element

Text 를 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

(스타일 'text-anchor': 'start' or 'middle' or 'end' 에 따라 위치 기준이 다름)

Kind: instance method of [RaphaelRenderer](#)

Overrides: [drawText](#)

Returns: Element - DOM Element

| Param | Type | Description |
|----------|-----------------------------------|---|
| position | Array. <Number> | 드로잉할 위치 좌표(스타일 'text-anchor': 'start' or 'middle' or 'end' 에 따라 기준이 다름) |
| text | String | 텍스트 |
| size | Array. <Number> | Text Width, Height, Angle |
| style | Style Object | 스타일 |
| id | String | Element ID 지정 |

Example

```
renderer.drawText([100, 100], 'Hello', null, {'text-anchor': 'start'});
```

raphaelRenderer.drawImage(position, imgSrc, size, style, id) ⇒ Element

Image 를 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [drawImage](#)

Returns: Element - DOM Element

| Param | Type | Description |
|----------|--------------------------------|----------------------------|
| position | Array.<Number> | 드로잉할 위치 좌표(좌상단 기준) |
| imgSrc | String | 이미지경로 |
| size | Array.<Number> | Image Width, Height, Angle |
| style | Style Object | 스타일 |
| id | String | Element ID 지정 |

Example

```
renderer.drawImage([100, 100], 'img.jpg', [50, 50]);
```

raphaelRenderer.drawEdge(line, style, id, isSelf) ⇒ Element

라인을 캔버스에 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [drawEdge](#)

Returns: Element - Group DOM Element with geometry

| Param | Type | Description |
|--------|--------------------------------|---------------|
| line | Line | 라인 |
| style | Style Object | 스타일 |
| id | String | Element ID 지정 |
| isSelf | Boolean | 셀프 연결 여부 |

raphaelRenderer.drawLabel(shapeElement, text, style) ⇒ Element

Shape 의 Label 을 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [drawLabel](#)

Returns: Element - DOM Element

| Param | Type | Description |
|--------------|------------------|-------------------------|
| shapeElement | Element String | Shape DOM element or ID |
| text | String | 텍스트 |
| style | Object | 스타일 |

drawLabel~getCenterOfEdge(element) ⇒ OG.Coordinate

라인(꺼은선)의 중심위치를 반환한다.

Kind: inner method of [drawLabel](#)

| Param | Type | Description |
|---------|--------------|-------------|
| element | Element Edge | 엘리먼트 |

raphaelRenderer.drawEdgeLabel(shapeElement, text, type) ⇒ Element

Edge 의 from, to Label 을 캔버스에 위치 및 사이즈 지정하여 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [drawEdgeLabel](#)

Returns: Element - DOM Element

| Param | Type | Description |
|--------------|------------------|-------------------------|
| shapeElement | Element String | Shape DOM element or ID |
| text | String | 텍스트 |
| type | String | 유형(FROM or TO) |

```
raphaelRenderer.redrawShape(element, excludeEdgeId)
```

Element 에 저장된 geom, angle, image, text 정보로 shape 을 redraw 한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [redrawShape](#)

| Param | Type | Description |
|--------------------|----------------|--------------------|
| element | Element | Shape 엘리먼트 |
| excludeEdgeIdArray | Array.<String> | redraw 제외할 Edge ID |

```
raphaelRenderer.redrawConnectedEdge(element)
```

Shape 의 연결된 Edge 를 redraw 한다.(이동 또는 리사이즈시)

Kind: instance method of [RaphaelRenderer](#)

Overrides: [redrawConnectedEdge](#)

| Param | Type |
|---------|---------|
| element | Element |

```
raphaelRenderer.connect(fromTerminal, toTerminal, edge, style, label, preventTrigger) ⇒ Element
```

두개의 터미널을 연결하고, 속성정보에 추가한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [connect](#)

Returns: Element - 연결된 Edge 엘리먼트

| Param | Type | Description |
|----------------|--------------------------------|--------------------|
| fromTerminal | Element Array.<Number> | 시작점 (fromTerminal) |
| toTerminal | Element Array.<Number> | 끝점 (toTerminal) |
| edge | Element | Edge Shape |
| style | Style Object | 스타일 |
| label | String | Label |
| preventTrigger | Boolean | 이벤트 트리거 발생 막기 |

`raphaelRenderer.disconnect(element)`

연결속성정보를 삭제한다. Edge 인 경우는 라인만 삭제하고, 일반 Shape 인 경우는 연결된 모든 Edge 를 삭제한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [disconnect](#)

| Param | Type |
|---------|---------|
| element | Element |

`raphaelRenderer.drawDropOverGuide(element)`

ID에 해당하는 Element 의 Edge 연결시 Drop Over 가이드를 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [drawDropOverGuide](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.drawGuide(element) ⇒ Object`

ID에 해당하는 Element 의 Move & Resize 용 가이드를 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [drawGuide](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.removeGuide(element)`

ID에 해당하는 Element 의 Move & Resize 용 가이드를 제거한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [removeGuide](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.removeAllGuide()`

모든 Move & Resize 용 가이드를 제거한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [removeAllGuide](#)

`raphaelRenderer.drawEdgeGuide(element) ⇒ Object`

ID에 해당하는 Edge Element 의 Move & Resize 용 가이드를 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [drawEdgeGuide](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.drawRubberBand(position, size, style) ⇒ Element`

Rectangle 모양의 마우스 드래그 선택 박스 영역을 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [drawRubberBand](#)

Returns: Element - DOM Element

| Param | Type | Description |
|----------|--------------------------------|---------------------------|
| position | Array.<Number> | 드로잉할 위치 좌표(좌상단) |
| size | Array.<Number> | Text Width, Height, Angle |
| style | Style Object | 스타일 |

`raphaelRenderer.removeRubberBand(root)`

Rectangle 모양의 마우스 드래그 선택 박스 영역을 제거한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [removeRubberBand](#)

| Param | Type | Description |
|-------|---------|--------------------------------|
| root | Element | first, rubberBand 정보를 저장한 엘리먼트 |

`raphaelRenderer.drawDraggableGuide(element) ⇒ Element`

ID에 해당하는 Element 의 Draggable 가이드를 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.drawCollapseGuide(element) ⇒ Element`

ID에 해당하는 Element 의 Collapse 가이드를 드로잉한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.removeCollapseGuide(element)`

ID에 해당하는 Element 의 Collapse 가이드를 제거한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [removeCollapseGuide](#)

| Param | Type |
|---------|---------|
| element | Element |

`raphaelRenderer.group(elements) ⇒ Element`

주어진 Shape 들을 그룹핑한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [group](#)

Returns: Element - Group Shape Element

| Param | Type |
|----------|-----------------|
| elements | Array.<Element> |

`raphaelRenderer.ungroup(groupElements) ⇒ Array.<Element>`

주어진 그룹들을 그룹해제한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [ungroup](#)

Returns: Array.<Element> - ungrouped Elements

| Param | Type |
|---------------|-----------------|
| groupElements | Array.<Element> |

`raphaelRenderer.addToGroup(groupElement, elements)`

주어진 Shape 들을 그룹에 추가한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [addToGroup](#)

| Param | Type |
|-------|------|
|-------|------|

groupElement Element
elements Array.<Element>

raphaelRenderer.collapse(element)

주어진 Shape 이 그룹인 경우 collapse 한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type |
|---------|---------|
| element | Element |

raphaelRenderer.expand(element)

주어진 Shape 이 그룹인 경우 expand 한다.

Kind: instance method of [RaphaelRenderer](#)

| Param | Type |
|---------|---------|
| element | Element |

raphaelRenderer.clear()

드로잉된 모든 오브젝트를 클리어한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [clear](#)

raphaelRenderer.removeShape(element)

Shape 을 캔버스에서 관련된 모듈을 삭제한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [removeShape](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

```
raphaelRenderer.remove(element)
```

ID에 해당하는 Element 를 캔버스에서 제거한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [remove](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

```
raphaelRenderer.removeChild(element)
```

하위 엘리먼트만 제거한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [removeChild](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

```
raphaelRenderer.getRootElement() ⇒ Element
```

랜더러 캔버스 Root Element 를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [getRootElement](#)

Returns: Element - Element

`raphaelRenderer.getRootGroup() ⇒ Element`

랜더러 캔버스 Root Group Element 를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - Element

`raphaelRenderer.getElementByPoint(position) ⇒ Element`

주어진 지점을 포함하는 Top Element 를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [getElementByPoint](#)

Returns: Element - Element

| Param | Type | Description |
|----------|----------------|-------------|
| position | Array.<Number> | 위치 좌표 |

`raphaelRenderer.getElementsByBBox(envelope) ⇒ Array.<Element>`

주어진 Boundary Box 영역에 포함되는 Shape(GEOM, TEXT, IMAGE) Element 를 반환한다.

모든 vertices를 포함한 엘리먼트를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array.<Element> - Element

| Param | Type | Description |
|----------|--------------------------|-----------------|
| envelope | Envelope | Boundary Box 영역 |

raphaelRenderer.setAttr(element, attribute)

엘리먼트에 속성값을 설정한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [setAttr](#)

| Param | Type | Description |
|-----------|------------------|---------------|
| element | Element String | Element 또는 ID |
| attribute | Object | 속성값 |

raphaelRenderer.getAttr(element, attrName) ⇒ Object

엘리먼트 속성값을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [getAttr](#)

Returns: Object - attribute 속성값

| Param | Type | Description |
|----------|------------------|---------------|
| element | Element String | Element 또는 ID |
| attrName | String | 속성이름 |

```
raphaelRenderer.setShapeStyle(element, style)
```

Shape 의 스타일을 변경한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [setShapeStyle](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |
| style | Object | 스타일 |

```
raphaelRenderer.toFront(element)
```

ID에 해당하는 Element 를 최상단 레이어로 이동한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [toFront](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

```
raphaelRenderer.toBack(element)
```

ID에 해당하는 Element 를 최하단 레이어로 이동한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [toBack](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

raphaelRenderer.bringForward(element)

ID에 해당하는 Element 를 앞으로 한단계 이동한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [bringForward](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

raphaelRenderer.sendBackward(element)

ID에 해당하는 Element 를 뒤로 한단계 이동한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [sendBackward](#)

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

raphaelRenderer.getCanvasSize() ⇒ Array.<Number>

랜더러 캔버스의 사이즈(Width, Height)를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [getCanvasSize](#)

Returns: Array.<Number> - Canvas Width, Height

```
raphaelRenderer.setCanvasSize(size)
```

랜더러 캔버스의 사이즈(Width, Height)를 변경한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [setCanvasSize](#)

| Param | Type | Description |
|-------|----------------|----------------------|
| size | Array.<Number> | Canvas Width, Height |

```
raphaelRenderer.fitCanvasSize(minSize, fitScale)
```

랜더러 캔버스의 사이즈(Width, Height)를 실제 존재하는 Shape 의 영역에 맞게 변경한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [fitCanvasSize](#)

| Param | Type | Description |
|----------|----------------|--|
| minSize | Array.<Number> | Canvas 최소 Width, Height |
| fitScale | Boolean | 주어진 minSize 에 맞게 fit 여부(Default:false) |

```
raphaelRenderer.setViewBox(position, size, isFit)
```

새로운 View Box 영역을 설정한다. (ZoomIn & ZoomOut 가능)

Kind: instance method of [RaphaelRenderer](#)

Overrides: [setViewBox](#)

| Param | Type | Description |
|----------|----------------|----------------------|
| position | Array.<Number> | 위치 좌표(좌상단 기준) |
| size | Array.<Number> | Canvas Width, Height |
| isFit | Boolean | Fit 여부 |

`raphaelRenderer.getScale()` ⇒ Number

Scale 을 반환한다. (리얼 사이즈 : Scale = 1)

Kind: instance method of [RaphaelRenderer](#)

Overrides: [getScale](#)

Returns: Number - 스케일값

`raphaelRenderer.setScale(scale)`

Scale 을 설정한다. (리얼 사이즈 : Scale = 1)

Kind: instance method of [RaphaelRenderer](#)

Overrides: [setScale](#)

| Param | Type | Description |
|-------|--------|-------------|
| scale | Number | 스케일값 |

`raphaelRenderer.show(element)`

ID에 해당하는 Element 를 캔버스에서 show 한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [show](#)

| Param | Type | Description |
|---------|-------------------------|-------------|
| element | Element StringElement | 또는 ID |

`raphaelRenderer.hide(element)`

ID에 해당하는 Element 를 캔버스에서 hide 한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [hide](#)

| Param | Type | Description |
|---------|-------------------------|-------------|
| element | Element StringElement | 또는 ID |

`raphaelRenderer.appendChild(srcElement, targetElement) ⇒ Element`

Source Element 를 Target Element 아래에 append 한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [appendChild](#)

Returns: Element - Source Element

| Param | Type | Description |
|---------------|-------------------------|-------------|
| srcElement | Element StringElement | 또는 ID |
| targetElement | Element StringElement | 또는 ID |

`raphaelRenderer.insertAfter(srcElement, targetElement) ⇒ Element`

Source Element 를 Target Element 이후에 insert 한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [insertAfter](#)

Returns: Element - Source Element

| Param | Type | Description |
|---------------|-------------------------|-------------|
| srcElement | Element StringElement | 또는 ID |
| targetElement | Element StringElement | 또는 ID |

raphaelRenderer.insertBefore(srcElement, targetElement) ⇒ Element

Source Element 를 Target Element 이전에 insert 한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [insertBefore](#)

Returns: Element - Source Element

| Param | Type | Description |
|---------------|-------------------------|-------------|
| srcElement | Element StringElement | 또는 ID |
| targetElement | Element StringElement | 또는 ID |

raphaelRenderer.move(element, offset) ⇒ Element

해당 Element 를 가로, 세로 Offset 만큼 이동한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [move](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |
| offset | Array.<Number> | [가로, 세로] |

raphaelRenderer.moveCentroid(element, position) ⇒ Element

주어진 중심좌표로 해당 Element 를 이동한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [moveCentroid](#)

Returns: Element - Element

| Param | Type | Description |
|----------|------------------|---------------|
| element | Element String | Element 또는 ID |
| position | Array.<Number> | [x, y] |

raphaelRenderer.rotate(element, angle) ⇒ Element

중심 좌표를 기준으로 주어진 각도 만큼 회전한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [rotate](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |
| angle | Number | 각도 |

raphaelRenderer.resize(element, offset) ⇒ Element

상, 하, 좌, 우 외곽선을 이동한 만큼 리사이즈 한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [resize](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|-------------------------|
| element | Element String | Element 또는 ID |
| offset | Array.<Number> | [상, 하, 좌, 우] 각 방향으로 + 값 |

raphaelRenderer.resizeBox(element, size) ⇒ Element

중심좌표는 고정된 채 Bounding Box 의 width, height 를 리사이즈 한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [resizeBox](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|-----------------|
| element | Element String | Element 또는 ID |
| size | Array.<Number> | [Width, Height] |

raphaelRenderer.clone(element) ⇒ Element

노드 Element 를 복사한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [clone](#)

Returns: Element - Element

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

raphaelRenderer.getElementById(id) ⇒ Element

ID로 Node Element 를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [getElementById](#)

Returns: Element - Element

| Param | Type |
|-------|--------|
| id | String |

raphaelRenderer.getElementsByType(shapeType, excludeType) ⇒ Array.<Element>

Shape 타입에 해당하는 Node Element 들을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array.<Element> - Element's Array

| Param | Type | Description |
|-------------|--------|---|
| shapeType | String | Shape 타입(GEOM, HTML, IMAGE, EDGE, GROUP), Null 이면 모든 타입 |
| excludeType | String | 제외 할 타입 |

raphaelRenderer.getBBox(element) ⇒ Object

해당 엘리먼트의 BoundingBox 영역 정보를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [getBBox](#)

Returns: Object - {width, height, x, y, x2, y2}

| Param | Type |
|---------|------------------|
| element | Element String |

`raphaelRenderer.getRootBBox()` ⇒ Object

부모노드기준으로 캔버스 루트 엘리먼트의 BoundingBox 영역 정보를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [getRootBBox](#)

Returns: Object - {width, height, x, y, x2, y2}

`raphaelRenderer.getRealRootBBox()` ⇒ Object

부모노드기준으로 캔버스 루트 엘리먼트의 실제 Shape 이 차지하는 BoundingBox 영역 정보를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Object - {width, height, x, y, x2, y2}

`raphaelRenderer.getContainer()` ⇒ Element

캔버스의 컨테이너 DOM element 를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [getContainer](#)

Returns: Element - 컨테이너

`raphaelRenderer.isSVG()` ⇒ Boolean

SVG 인지 여부를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [isSVG](#)

Returns: Boolean - svg 여부

`raphaelRenderer.isVML()` ⇒ Boolean

VML 인지 여부를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [isVML](#)

Returns: Boolean - vml 여부

`raphaelRenderer.getPrevEdges(element)` ⇒ Array.<Element>

연결된 이전 Edge Element 들을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array.<Element> - Previous Element's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.getNextEdges(element) ⇒ Array.<Element>`

연결된 이후 Edge Element 들을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: `Array.<Element>` - Previous Element's Array

| Param | Type | Description |
|----------------------|-------------------------------|---------------|
| <code>element</code> | <code>Element String</code> | Element 또는 ID |

`raphaelRenderer.getPrevShapes(element) ⇒ Array.<Element>`

연결된 이전 노드 Element 들을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: `Array.<Element>` - Previous Element's Array

| Param | Type | Description |
|----------------------|-------------------------------|---------------|
| <code>element</code> | <code>Element String</code> | Element 또는 ID |

`raphaelRenderer.getPrevShapeIds(element) ⇒ Array.<String>`

연결된 이전 노드 Element ID들을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: `Array.<String>` - Previous Element Id's Array

| Param | Type | Description |
|----------------------|-------------------------------|---------------|
| <code>element</code> | <code>Element String</code> | Element 또는 ID |

`raphaelRenderer.getNextShapes(element) ⇒ Array.<Element>`

연결된 이후 노드 Element 들을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array.<Element> - Previous Element's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.getNextShapeIds(element) ⇒ Array.<String>`

연결된 이후 노드 Element ID들을 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array.<String> - Previous Element Id's Array

| Param | Type | Description |
|---------|------------------|---------------|
| element | Element String | Element 또는 ID |

`raphaelRenderer.isTopGroup(Element) ⇒ boolean`

최상위 그룹 엘리먼트인지 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: boolean - true false

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

Element Element 엘리먼트

`raphaelRenderer.getParent(element) ⇒ Element`

부모 엘리먼트를 반환한다. 부모가 루트일때는 반환하지 않는다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Element - element 엘리먼트

| Param | Type | Description |
|---------|---------|-------------|
| element | Element | 엘리먼트 |

`raphaelRenderer.getChilds(element) ⇒ Array`

그룹의 하위 엘리먼트를 반환한다.

Kind: instance method of [RaphaelRenderer](#)

Returns: Array - Elements

| Param | Type | Description |
|---------|---------|-------------|
| element | Element | 엘리먼트 |

`raphaelRenderer.isGroup(element) ⇒ boolean`

그룹의 Shape 인지 반환한다. RootGroup 일 경우는 제외.

Kind: instance method of [RaphaelRenderer](#)

Returns: boolean - true false

| Param | Type | Description |
|---------|---------|-------------|
| element | Element | 엘리먼트 |

`raphaelRenderer.getAllShapes() ⇒ Array`

캔버스의 모든 Shape 들을 리턴

Kind: instance method of [RaphaelRenderer](#)

Returns: Array - Elements

`raphaelRenderer.getAllEdges() ⇒ Array`

캔버스의 모든 Edge를 리턴

Kind: instance method of [RaphaelRenderer](#)

Returns: Array - Edge Elements

`raphaelRenderer.getAllNotEdges() ⇒ Array`

캔버스의 모든 Edge 가 아닌 shpaes 를 리턴

Kind: instance method of [RaphaelRenderer](#)

Returns: Array - Edge Elements

`raphaelRenderer.isEdge() ⇒ boolean`

Edge 여부를 판단.

Kind: instance method of [RaphaelRenderer](#)

Returns: boolean - true false

`raphaelRenderer.isShape()` ⇒ boolean

Shape 여부를 판단.

Kind: instance method of [RaphaelRenderer](#)

Returns: boolean - true false

`raphaelRenderer.initHistory()`

캔버스의 히스토리를 초기화한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [initHistory](#)

`raphaelRenderer.addHistory()`

캔버스에 히스토리를 추가한다.

Kind: instance method of [RaphaelRenderer](#)

Overrides: [addHistory](#)

`raphaelRenderer.undo()`

캔버스의 Undo

Kind: instance method of [RaphaelRenderer](#)

Overrides: [undo](#)

```
raphaelRenderer.redo()
```

캔버스의 Redo

Kind: instance method of [RaphaelRenderer](#)

Overrides: [redo](#)

OG.marker : object

Kind: static namespace of [OG](#)

- [.marker](#) : object
 - [.ArrowMarker](#) \leftarrow [IMarker](#)
 - [new OG.marker.ArrowMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) \Rightarrow *
 - [.CircleMarker](#) \leftarrow [IMarker](#)

- [new OG.marker.CircleMarker\(\)](#)
- [.MARKER_ID](#) : String
- [.geom](#) : [Geometry](#)
- [.createMarker\(\)](#) ⇒ *
- [.IMarker](#)
 - [new OG.marker.IMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *
- [.RectangleMarker](#) ⇐ [IMarker](#)
 - [new OG.marker.RectangleMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *
- [.SwitchLMarker](#) ⇐ [IMarker](#)
 - [new OG.marker.SwitchLMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *
- [.SwitchRMarker](#) ⇐ [IMarker](#)
 - [new OG.marker.SwitchRMarker\(\)](#)

- [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *
- [.SwitchXMarker](#) ⇐ [IMarker](#)
 - [new OG.marker.SwitchXMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *

marker.ArrowMarker ⇐ [IMarker](#)

Kind: static class of [marker](#)

Extends: [IMarker](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.ArrowMarker](#) ⇐ [IMarker](#)
 - [new OG.marker.ArrowMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *

`new OG.marker.ArrowMarker()`

Rectangle Maker

`arrowMarker.MARKER_ID` : String

marker 을 구분하는 marker ID(marker 클래스명과 일치)

Kind: instance property of [ArrowMarker](#)

Overrides: [MARKER_ID](#)

`arrowMarker.geom` : [Geometry](#)

marker 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [ArrowMarker](#)

Overrides: [geom](#)

`arrowMarker.createMarker()` ⇒ *

드로잉할 marker 를 생성하여 반환한다.

Kind: instance method of [ArrowMarker](#)

Overrides: [createMarker](#)

Returns: * - Marker 정보

marker.CircleMarker \Leftarrow [IMarker](#)

Kind: static class of [marker](#)

Extends: [IMarker](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.CircleMarker](#) \Leftarrow [IMarker](#)
 - [new OG.marker.CircleMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) \Rightarrow *

`new OG.marker.CircleMarker()`

Circle Marker

`circleMarker.MARKER_ID` : String

marker 을 구분하는 marker ID(marker 클래스명과 일치)

Kind: instance property of [CircleMarker](#)

Overrides: [MARKER_ID](#)

circleMarker.geom : [Geometry](#)

marker 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [CircleMarker](#)

Overrides: [geom](#)

circleMarker.createMarker() ⇒ *

드로잉할 marker 를 생성하여 반환한다.

Kind: instance method of [CircleMarker](#)

Overrides: [createMarker](#)

Returns: * - Marker 정보

marker.IMarker

Kind: static class of [marker](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.IMarker](#)
 - [new OG.marker.IMarker\(\)](#)

- [.MARKER_ID](#) : String
- [.geom](#) : [Geometry](#)
- [.createMarker\(\)](#) ⇒ *

new OG.marker.IMarker()

도형 Path 의 Marker 정보 최상위 인터페이스

iMarker.MARKER_ID : String

marker 을 구분하는 marker ID(marker 클래스명과 일치)

Kind: instance property of [IMarker](#)

iMarker.geom : [Geometry](#)

marker 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [IMarker](#)

iMarker.createMarker() ⇒ *

드로잉할 marker 를 생성하여 반환한다.

Kind: instance abstract method of [IMarker](#)

Returns: * - Marker 정보

marker.RectangleMarker \Leftarrow [IMarker](#)

Kind: static class of [marker](#)

Extends: [IMarker](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.RectangleMarker](#) \Leftarrow [IMarker](#)
 - [new OG.marker.RectangleMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) \Rightarrow *

[new OG.marker.RectangleMarker\(\)](#)

Rectangle Maker

[rectangleMarker.MARKER_ID](#) : String

marker 을 구분하는 marker ID(marker 클래스명과 일치)

Kind: instance property of [RectangleMarker](#)

Overrides: [MARKER_ID](#)

rectangleMarker.geom : [Geometry](#)

marker 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [RectangleMarker](#)

Overrides: [geom](#)

rectangleMarker.createMarker() ⇒ *

드로잉할 marker 를 생성하여 반환한다.

Kind: instance method of [RectangleMarker](#)

Overrides: [createMarker](#)

Returns: * - Marker 정보

marker.SwitchLMarker ⇐ [IMarker](#)

Kind: static class of [marker](#)

Extends: [IMarker](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.SwitchLMarker](#) \Leftarrow [IMarker](#)
 - [new OG.marker.SwitchLMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) \Rightarrow *

`new OG.marker.SwitchLMarker()`

Rectangle Maker

`switchLMarker.MARKER_ID` : String

marker 을 구분하는 marker ID(marker 클래스명과 일치)

Kind: instance property of [SwitchLMarker](#)

Overrides: [MARKER_ID](#)

`switchLMarker.geom` : [Geometry](#)

marker 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [SwitchLMarker](#)

Overrides: [geom](#)

`switchLMarker.createMarker()` ⇒ *

드로잉할 `marker` 를 생성하여 반환한다.

Kind: instance method of [SwitchLMarker](#)

Overrides: [createMarker](#)

Returns: * - Marker 정보

`marker.SwitchRMarker` ⇐ [IMarker](#)

Kind: static class of [marker](#)

Extends: [IMarker](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.SwitchRMarker](#) ⇐ [IMarker](#)
 - [new OG.marker.SwitchRMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) ⇒ *

`new OG.marker.SwitchRMarker()`

Rectangle Maker

switchRMarker.MARKER_ID : String

marker 을 구분하는 marker ID(marker 클래스명과 일치)

Kind: instance property of [SwitchRMarker](#)

Overrides: [MARKER_ID](#)

switchRMarker.geom : [Geometry](#)

marker 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [SwitchRMarker](#)

Overrides: [geom](#)

switchRMarker.createMarker() ⇒ *

드로잉할 marker 를 생성하여 반환한다.

Kind: instance method of [SwitchRMarker](#)

Overrides: [createMarker](#)

Returns: * - Marker 정보

marker.SwitchXMarker ⇐ [IMarker](#)

Kind: static class of [marker](#)

Extends: [IMarker](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.SwitchXMarker](#) \Leftarrow [IMarker](#)
 - [new OG.marker.SwitchXMarker\(\)](#)
 - [.MARKER_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createMarker\(\)](#) \Rightarrow *

`new OG.marker.SwitchXMarker()`

Rectangle Maker

`switchXMarker.MARKER_ID` : String

marker 을 구분하는 marker ID(marker 클래스명과 일치)

Kind: instance property of [SwitchXMarker](#)

Overrides: [MARKER_ID](#)

`switchXMarker.geom` : [Geometry](#)

marker 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [SwitchXMarker](#)

Overrides: [geom](#)

`switchXMarker.createMarker()` ⇒ *

드로잉할 marker 를 생성하여 반환한다.

Kind: instance method of [SwitchXMarker](#)

Overrides: [createMarker](#)

Returns: * - Marker 정보

OG.pattern : object

Kind: static namespace of [OG](#)

- [.pattern](#) : object
 - [.HatchedPattern](#) ⇐ [IPattern](#)
 - [new OG.pattern.HatchedPattern\(\)](#)
 - [.PATTERN_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createPattern\(\)](#) ⇒ *
 - [.IPattern](#)

- [new OG.pattern.IPattern\(\)](#)
 - [.PATTERN_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createPattern\(\)](#) ⇒ *
- [.RectPattern](#) ⇐ [IPattern](#)
 - [new OG.pattern.RectPattern\(\)](#)
 - [.PATTERN_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createPattern\(\)](#) ⇒ *

pattern.HatchedPattern ⇐ [IPattern](#)

Kind: static class of [pattern](#)

Extends: [IPattern](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.HatchedPattern](#) ⇐ [IPattern](#)
 - [new OG.pattern.HatchedPattern\(\)](#)
 - [.PATTERN_ID](#) : String
 - [.geom](#) : [Geometry](#)

- [.createPattern\(\)](#) ⇒ *

new OG.pattern.HatchedPattern()

Hatched Pattern

hatchedPattern.PATTERN_ID : String

pattern 을 구분하는 pattern ID(pattern 클래스명과 일치)

Kind: instance property of [HatchedPattern](#)

Overrides: [PATTERN_ID](#)

hatchedPattern.geom : [Geometry](#)

pattern 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [HatchedPattern](#)

Overrides: [geom](#)

hatchedPattern.createPattern() ⇒ *

드로잉할 pattern 를 생성하여 반환한다.

Kind: instance method of [HatchedPattern](#)

Overrides: [createPattern](#)

Returns: * - pattern 정보

pattern.IPattern

Kind: static class of [pattern](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.IPattern](#)
 - [new OG.pattern.IPattern\(\)](#)
 - [.PATTERN_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.createPattern\(\)](#) ⇒ *

`new OG.pattern.IPattern()`

도형 Pattern 정보 최상위 인터페이스

`iPattern.PATTERN_ID` : String

pattern 을 구분하는 pattern ID(pattern 클래스명과 일치)

Kind: instance property of [IPattern](#)

iPattern.geom : [Geometry](#)

pattern 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [IPattern](#)

iPattern.createPattern() ⇒ *

드로잉할 pattern 를 생성하여 반환한다.

Kind: instance abstract method of [IPattern](#)

Returns: * - pattern 정보

pattern.RectPattern ⇐ [IPattern](#)

Kind: static class of [pattern](#)

Extends: [IPattern](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.RectPattern](#) ⇐ [IPattern](#)
 - [new OG.pattern.RectPattern\(\)](#)
 - [.PATTERN_ID](#) : String

- [.geom](#) : [Geometry](#)
- [.createPattern\(\)](#) ⇒ *

new OG.pattern.RectPattern()

Rect Pattern

rectPattern.PATTERN_ID : String

pattern 을 구분하는 pattern ID(pattern 클래스명과 일치)

Kind: instance property of [RectPattern](#)

Overrides: [PATTERN_ID](#)

rectPattern.geom : [Geometry](#)

pattern 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [RectPattern](#)

Overrides: [geom](#)

rectPattern.createPattern() ⇒ *

드로잉할 pattern 를 생성하여 반환한다.

Kind: instance method of [RectPattern](#)

Overrides: [createPattern](#)

Returns: * - pattern 정보

OG.shape : object

Kind: static namespace of [OG](#)

- [.shape](#) : object
 - [.CircleShape](#) \Leftarrow [GeomShape](#)
 - [new OG.shape.CircleShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean

- [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.clone\(\)](#) ⇒ [IShape](#)
 - [.createShape\(\)](#) ⇒ *
- [.EdgeShape](#) ⇐ [IShape](#)
 - [new OG.shape.EdgeShape\(from, to, label, fromLabel, toLabel\)](#)
 - [.from](#) : Array.<Number>
 - [.to](#) : Array.<Number>
 - [.fromLabel](#) : String
 - [.toLabel](#) : String
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean

- [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.EllipseShape](#) ⇐ [GeomShape](#)
 - [new OG.shape.EllipseShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean

- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.clone\(\)](#) ⇒ [IShape](#)
- [.createShape\(\)](#) ⇒ *
- [.GeomShape](#) ⇐ [IShape](#)
 - [new OG.shape.GeomShape\(\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean

- [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.GroupShape](#) ⇐ [IShape](#)
 - [new OG.shape.GroupShape\(label\)](#)
 - [.GROUP_DROPABLE](#) : Boolean
 - [.GROUP_COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean

- [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.HorizontalLaneShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.HorizontalLaneShape\(label\)](#)
 - [.GROUP_DROPABLE](#) : Boolean
 - [.GROUP_COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean

- [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.HorizontalPoolShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.HorizontalPoolShape\(label\)](#)
 - [.GROUP_DROPABLE](#) : Boolean
 - [.GROUP_COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean

- [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.HtmlShape](#) ⇐ [IShape](#)
 - [new OG.shape.HtmlShape\(html, label\)](#)
 - [.html](#) : String
 - [.angle](#) : Number
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean

- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

- [.IShape](#)

- [new OG.shape.IShape\(\)](#)
- [.TYPE](#) : String
- [.SHAPE_ID](#) : String
- [.geom](#) : [Geometry](#)
- [.label](#) : String
- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean

- [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.ImageShape](#) ⇐ [IShape](#)
 - [new OG.shape.ImageShape\(image, label\)](#)
 - [.image](#) : String
 - [.angle](#) : Number
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean

- [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.RectangleShape](#) ⇐ [GeomShape](#)
 - [new OG.shape.RectangleShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean

- [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.clone\(\)](#) ⇒ [IShape](#)
 - [.createShape\(\)](#) ⇒ *
- [.SpotShape](#) ⇐ [GeomShape](#)
 - [new OG.shape.SpotShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean

- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.clone\(\)](#) ⇒ [IShape](#)
- [.createShape\(\)](#) ⇒ *
- [.TextShape](#) ⇐ [IShape](#)
 - [new OG.shape.TextShape\(text\)](#)
 - [.text](#) : String
 - [.angle](#) : Number
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean

- [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.VerticalLaneShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.VerticalLaneShape\(label\)](#)
 - [.GROUP_DROPABLE](#) : Boolean
 - [.GROUP_COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean

- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)
- [.VerticalPoolShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.VerticalPoolShape\(label\)](#)
 - [.GROUP_DROPABLE](#) : Boolean
 - [.GROUP_COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean

- [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) ⇒ *
 - [.clone\(\)](#) ⇒ [IShape](#)
- [.bpmn](#) : object
 - [.elec](#) : object

shape.CircleShape ⇐ [GeomShape](#)

Kind: static class of [shape](#)

Extends: [GeomShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.CircleShape](#) ⇐ [GeomShape](#)
 - [new OG.shape.CircleShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String

- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.clone\(\)](#) ⇒ [IShape](#)
- [.createShape\(\)](#) ⇒ *

```
new OG.shape.CircleShape(label)
```

Circle Shape

| Param | Type | Description |
|-------|--------|---------------|
| label | String | 라벨 [Optional] |

circleShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [CircleShape](#)

circleShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [CircleShape](#)

Overrides: [SHAPE_ID](#)

circleShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [CircleShape](#)

Overrides: [geom](#)

circleShape.label : String

Shape 라벨 텍스트

Kind: instance property of [CircleShape](#)

Overrides: [label](#)

circleShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [CircleShape](#)

circleShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [CircleShape](#)

circleShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [CircleShape](#)

circleShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [CircleShape](#)

circleShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [CircleShape](#)

circleShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [CircleShape](#)

circleShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [CircleShape](#)

circleShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [CircleShape](#)

circleShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [CircleShape](#)

circleShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [CircleShape](#)

circleShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [CircleShape](#)

circleShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [CircleShape](#)

circleShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [CircleShape](#)

circleShape.data : Object

도형의 데이터

Kind: instance property of [CircleShape](#)

circleShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [CircleShape](#)

circleShape.clone() ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [CircleShape](#)

Returns: [IShape](#) - 복사된 인스턴스

circleShape.createShape() ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [CircleShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

shape.EdgeShape ⇐ [IShape](#)

Kind: static class of [shape](#)

Extends: [IShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.EdgeShape](#) \Leftarrow [IShape](#)
 - [new OG.shape.EdgeShape\(from, to, label, fromLabel, toLabel\)](#)
 - [.from](#) : Array.<Number>
 - [.to](#) : Array.<Number>
 - [.fromLabel](#) : String
 - [.toLabel](#) : String
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean

- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

```
new OG.shape.EdgeShape(from, to, label, fromLabel, toLabel)
```

Edge Shape

| Param | Type | Description |
|-----------|----------------|-------------------|
| from | Array.<Number> | 와이어 시작 좌표 |
| to | Array.<Number> | 와이어 끝 좌표 |
| label | String | 라벨 [Optional] |
| fromLabel | String | 시작점 라벨 [Optional] |
| toLabel | String | 끝점 라벨 [Optional] |

```
edgeShape.from : Array.<Number>
```

Edge 시작 좌표

Kind: instance property of [EdgeShape](#)

```
edgeShape.to : Array.<Number>
```

Edge 끝 좌표

Kind: instance property of [EdgeShape](#)

edgeShape.fromLabel : String

Edge 시작점 라벨

Kind: instance property of [EdgeShape](#)

edgeShape.toLabel : String

Edge 끝점 라벨

Kind: instance property of [EdgeShape](#)

edgeShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [EdgeShape](#)

Overrides: [TYPE](#)

edgeShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [EdgeShape](#)

Overrides: [SHAPE_ID](#)

edgeShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [EdgeShape](#)

Overrides: [geom](#)

edgeShape.label : String

Shape 라벨 텍스트

Kind: instance property of [EdgeShape](#)

Overrides: [label](#)

edgeShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [EdgeShape](#)

edgeShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [EdgeShape](#)

edgeShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [EdgeShape](#)

edgeShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [EdgeShape](#)

edgeShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [EdgeShape](#)

edgeShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [EdgeShape](#)

edgeShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [EdgeShape](#)

edgeShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [EdgeShape](#)

edgeShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [EdgeShape](#)

edgeShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [EdgeShape](#)

edgeShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [EdgeShape](#)

edgeShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [EdgeShape](#)

edgeShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [EdgeShape](#)

edgeShape.data : Object

도형의 데이터

Kind: instance property of [EdgeShape](#)

edgeShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [EdgeShape](#)

edgeShape.createShape() ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [EdgeShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

`edgeShape.clone()` ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [EdgeShape](#)

Overrides: [clone](#)

Returns: [IShape](#) - 복사된 인스턴스

`shape.EllipseShape` ⇐ [GeomShape](#)

Kind: static class of [shape](#)

Extends: [GeomShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.EllipseShape](#) ⇐ [GeomShape](#)
 - [new OG.shape.EllipseShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String

- [.geom](#) : [Geometry](#)
- [.label](#) : String
- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.clone\(\)](#) ⇒ [IShape](#)
- [.createShape\(\)](#) ⇒ *

```
new OG.shape.EllipseShape(label)
```

Ellipse Shape

| Param | Type | Description |
|-------|--------|---------------|
| label | String | 라벨 [Optional] |

ellipseShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [EllipseShape](#)

ellipseShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [EllipseShape](#)

Overrides: [SHAPE_ID](#)

ellipseShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [EllipseShape](#)

Overrides: [geom](#)

ellipseShape.label : String

Shape 라벨 텍스트

Kind: instance property of [EllipseShape](#)

Overrides: [label](#)

ellipseShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [EllipseShape](#)

ellipseShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [EllipseShape](#)

ellipseShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [EllipseShape](#)

ellipseShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [EllipseShape](#)

ellipseShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [EllipseShape](#)

ellipseShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [EllipseShape](#)

ellipseShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [EllipseShape](#)

ellipseShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [EllipseShape](#)

ellipseShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [EllipseShape](#)

ellipseShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [EllipseShape](#)

ellipseShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [EllipseShape](#)

ellipseShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [EllipseShape](#)

ellipseShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [EllipseShape](#)

ellipseShape.data : Object

도형의 데이터

Kind: instance property of [EllipseShape](#)

ellipseShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [EllipseShape](#)

ellipseShape.clone() ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [EllipseShape](#)

Returns: [IShape](#) - 복사된 인스턴스

ellipseShape.createShape() ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [EllipseShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

shape.GeomShape \Leftarrow [IShape](#)

Kind: static class of [shape](#)

Extends: [IShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.GeomShape](#) \Leftarrow [IShape](#)
 - [new OG.shape.GeomShape\(\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean

- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

new OG.shape.GeoShape()

Geometry Shape

geomShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [GeoShape](#)

Overrides: [TYPE](#)

geomShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [GeoShape](#)

geomShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [GeomShape](#)

geomShape.label : String

Shape 라벨 텍스트

Kind: instance property of [GeomShape](#)

geomShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [GeomShape](#)

geomShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [GeomShape](#)

geomShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [GeomShape](#)

geomShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [GeomShape](#)

geomShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [GeomShape](#)

geomShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [GeomShape](#)

geomShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [GeomShape](#)

geomShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [GeomShape](#)

geomShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [GeomShape](#)

geomShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [GeomShape](#)

geomShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [GeomShape](#)

geomShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [GeomShape](#)

`geomShape.LABEL_EDITABLE` : Boolean

라벨 수정여부

Kind: instance property of [GeomShape](#)

`geomShape.data` : Object

도형의 데이터

Kind: instance property of [GeomShape](#)

`geomShape.textList` : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [GeomShape](#)

`geomShape.createShape()` ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance abstract method of [GeomShape](#)

Returns: * - Shape 정보

`geomShape.clone()` ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [GeomShape](#)

Overrides: [clone](#)

Returns: [IShape](#) - 복사된 인스턴스

`shape.GroupShape` ⇐ [IShape](#)

Kind: static class of [shape](#)

Extends: [IShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.GroupShape](#) ⇐ [IShape](#)
 - [new OG.shape.GroupShape\(label\)](#)
 - [.GROUP_DROPABLE](#) : Boolean
 - [.GROUP_COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean

- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

```
new OG.shape.GroupShape(label)
```

Group Shape

| Param | Type | Description |
|-------|--------|---------------|
| label | String | 라벨 [Optional] |

groupShape.GROUP_DROPABLE : Boolean

그룹핑 가능여부

Kind: instance property of [GroupShape](#)

groupShape.GROUP_COLLAPSIBLE : Boolean

최소화 가능여부

Kind: instance property of [GroupShape](#)

groupShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [GroupShape](#)

Overrides: [TYPE](#)

groupShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [GroupShape](#)

Overrides: [SHAPE_ID](#)

groupShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [GroupShape](#)

Overrides: [geom](#)

groupShape.label : String

Shape 라벨 텍스트

Kind: instance property of [GroupShape](#)

Overrides: [label](#)

groupShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [GroupShape](#)

groupShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [GroupShape](#)

groupShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [GroupShape](#)

groupShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [GroupShape](#)

groupShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [GroupShape](#)

Overrides: [CONNECTABLE](#)

groupShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [GroupShape](#)

groupShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [GroupShape](#)

groupShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [GroupShape](#)

Overrides: [SELF_CONNECTABLE](#)

groupShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [GroupShape](#)

groupShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [GroupShape](#)

groupShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [GroupShape](#)

groupShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [GroupShape](#)

groupShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [GroupShape](#)

groupShape.data : Object

도형의 데이터

Kind: instance property of [GroupShape](#)

groupShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [GroupShape](#)

groupShape.createShape() ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [GroupShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

`groupShape.clone()` ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [GroupShape](#)

Overrides: [clone](#)

Returns: [IShape](#) - 복사된 인스턴스

`shape.HorizontalLaneShape` ⇐ [GroupShape](#)

Kind: static class of [shape](#)

Extends: [GroupShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.HorizontalLaneShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.HorizontalLaneShape\(label\)](#)
 - [.GROUP_DROPABLE](#) : Boolean
 - [.GROUP_COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String

- [.SHAPE_ID](#) : String
- [.geom](#) : [Geometry](#)
- [.label](#) : String
- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

new OG.shape.HorizontalLaneShape(label)

Horizontal Swimlane Shape

| Param | Type | Description |
|-------|--------|---------------|
| label | String | 라벨 [Optional] |

horizontalLaneShape.GROUP_DROPABLE : Boolean

그룹핑 가능여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.GROUP_COLLAPSIBLE : Boolean

최소화 가능여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [HorizontalLaneShape](#)

Overrides: [SHAPE_ID](#)

horizontalLaneShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [HorizontalLaneShape](#)

Overrides: [geom](#)

horizontalLaneShape.label : String

Shape 라벨 텍스트

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.data : Object

도형의 데이터

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [HorizontalLaneShape](#)

horizontalLaneShape.createShape() ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [HorizontalLaneShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

`horizontalLaneShape.clone()` ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [HorizontalLaneShape](#)

Returns: [IShape](#) - 복사된 인스턴스

`shape.HorizontalPoolShape` ⇐ [GroupShape](#)

Kind: static class of [shape](#)

Extends: [GroupShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.HorizontalPoolShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.HorizontalPoolShape\(label\)](#)
 - [.GROUP_DROPABLE](#) : Boolean
 - [.GROUP_COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)

- [.label](#) : String
- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

```
new OG.shape.HorizontalPoolShape(label)
```

Horizontal Pool Shape

| Param | Type | Description |
|-------|--------|---------------|
| label | String | 라벨 [Optional] |

horizontalPoolShape.GROUP_DROPABLE : Boolean

그룹핑 가능여부

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.GROUP_COLLAPSIBLE : Boolean

최소화 가능여부

Kind: instance property of [HorizontalPoolShape](#)

Overrides: [GROUP_COLLAPSIBLE](#)

horizontalPoolShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [HorizontalPoolShape](#)

Overrides: [SHAPE_ID](#)

horizontalPoolShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [HorizontalPoolShape](#)

Overrides: [geom](#)

horizontalPoolShape.label : String

Shape 라벨 텍스트

Kind: instance property of [HorizontalPoolShape](#)

Overrides: [label](#)

horizontalPoolShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [HorizontalPoolShape](#)

Overrides: [CONNECTABLE](#)

horizontalPoolShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.data : Object

도형의 데이터

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [HorizontalPoolShape](#)

horizontalPoolShape.createShape() ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [HorizontalPoolShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

`horizontalPoolShape.clone()` ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [HorizontalPoolShape](#)

Returns: [IShape](#) - 복사된 인스턴스

`shape.HtmlShape` ⇐ [IShape](#)

Kind: static class of [shape](#)

Extends: [IShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.HtmlShape](#) ⇐ [IShape](#)
 - [new OG.shape.HtmlShape\(html, label\)](#)
 - [.html](#) : String
 - [.angle](#) : Number
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String

- [.geom](#) : [Geometry](#)
- [.label](#) : String
- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

```
new OG.shape.HtmlShape(html, label)
```

ForeignObject HTML Shape

| Param | Type | Description |
|-------|--------|-----------------|
| html | String | 임베드 HTML String |
| label | String | 라벨 [Optional] |

htmlShape.html : String

드로잉할 임베드 HTML String

Kind: instance property of [HtmlShape](#)

htmlShape.angle : Number

회전각도

Kind: instance property of [HtmlShape](#)

htmlShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [HtmlShape](#)

Overrides: [TYPE](#)

htmlShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [HtmlShape](#)

Overrides: [SHAPE_ID](#)

htmlShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [HtmlShape](#)

htmlShape.label : String

Shape 라벨 텍스트

Kind: instance property of [HtmlShape](#)

Overrides: [label](#)

htmlShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [HtmlShape](#)

htmlShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [HtmlShape](#)

htmlShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [HtmlShape](#)

htmlShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [HtmlShape](#)

htmlShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [HtmlShape](#)

htmlShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [HtmlShape](#)

htmlShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [HtmlShape](#)

htmlShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [HtmlShape](#)

htmlShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [HtmlShape](#)

htmlShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [HtmlShape](#)

htmlShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [HtmlShape](#)

htmlShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [HtmlShape](#)

htmlShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [HtmlShape](#)

htmlShape.data : Object

도형의 데이터

Kind: instance property of [HtmlShape](#)

htmlShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [HtmlShape](#)

htmlShape.createShape() ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [HtmlShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

htmlShape.clone() ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [HtmlShape](#)

Overrides: [clone](#)

Returns: [IShape](#) - 복사된 인스턴스

shape.IShape

Kind: static class of [shape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.IShape](#)
 - [new OG.shape.IShape\(\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)

- [.label](#) : String
- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

new OG.shape.IShape()

도형, 텍스트, 이미지 등의 드로잉 될 Object 의 정보를 저장하는 Shape 정보 최상위 인터페이스

iShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [IShape](#)

iShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [IShape](#)

iShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [IShape](#)

iShape.label : String

Shape 라벨 텍스트

Kind: instance property of [IShape](#)

iShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [IShape](#)

iShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [IShape](#)

iShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [IShape](#)

iShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [IShape](#)

iShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [IShape](#)

iShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [IShape](#)

iShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [IShape](#)

iShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [IShape](#)

iShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [IShape](#)

iShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [IShape](#)

iShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [IShape](#)

iShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [IShape](#)

iShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [IShape](#)

iShape.data : Object

도형의 데이터

Kind: instance property of [IShape](#)

iShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [IShape](#)

iShape.createShape() ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance abstract method of [IShape](#)

Returns: * - Shape 정보

iShape.clone() ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance abstract method of [IShape](#)

Returns: [IShape](#) - 복사된 인스턴스

shape.ImageShape ⇐ [IShape](#)

Kind: static class of [shape](#)

Extends: [IShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.ImageShape](#) \Leftarrow [IShape](#)
 - [new OG.shape.ImageShape\(image, label\)](#)
 - [.image](#) : String
 - [.angle](#) : Number
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.createShape\(\)](#) \Rightarrow *
 - [.clone\(\)](#) \Rightarrow [IShape](#)

```
new OG.shape.ImageShape(image, label)
```

Image Shape

| Param | Type | Description |
|-------|--------|---------------|
| image | String | 이미지 URL |
| label | String | 라벨 [Optional] |

```
imageShape.image : String
```

드로잉할 이미지 URL

Kind: instance property of [ImageShape](#)

```
imageShape.angle : Number
```

회전각도

Kind: instance property of [ImageShape](#)

```
imageShape.TYPE : String
```

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [ImageShape](#)

Overrides: [TYPE](#)

imageShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [ImageShape](#)

Overrides: [SHAPE_ID](#)

imageShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [ImageShape](#)

imageShape.label : String

Shape 라벨 텍스트

Kind: instance property of [ImageShape](#)

Overrides: [label](#)

imageShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [ImageShape](#)

imageShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [ImageShape](#)

imageShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [ImageShape](#)

imageShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [ImageShape](#)

imageShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [ImageShape](#)

imageShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [ImageShape](#)

imageShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [ImageShape](#)

imageShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [ImageShape](#)

imageShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [ImageShape](#)

imageShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [ImageShape](#)

imageShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [ImageShape](#)

imageShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [ImageShape](#)

imageShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [ImageShape](#)

imageShape.data : Object

도형의 데이터

Kind: instance property of [ImageShape](#)

imageShape.textList : Array

Kind: instance property of [ImageShape](#)

`imageShape.createShape()` ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [ImageShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

`imageShape.clone()` ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [ImageShape](#)

Overrides: [clone](#)

Returns: [IShape](#) - 복사된 인스턴스

`shape.RectangleShape` ⇐ [GeomShape](#)

Kind: static class of [shape](#)

Extends: [GeomShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.RectangleShape](#) \Leftarrow [GeomShape](#)
 - [new OG.shape.RectangleShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean
 - [.SELF_CONNECTABLE](#) : Boolean
 - [.CONNECT_CLONEABLE](#) : Boolean
 - [.CONNECT_REQUIRED](#) : Boolean
 - [.CONNECT_STYLE_CHANGE](#) : Boolean
 - [.DELETABLE](#) : Boolean
 - [.LABEL_EDITABLE](#) : Boolean
 - [.data](#) : Object
 - [.textList](#) : Array
 - [.clone\(\)](#) \Rightarrow [IShape](#)
 - [.createShape\(\)](#) \Rightarrow *

```
new OG.shape.RectangleShape(label)
```

Rectangle Shape

| Param | Type | Description |
|-------|--------|---------------|
| label | String | 라벨 [Optional] |

```
rectangleShape.TYPE : String
```

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [RectangleShape](#)

```
rectangleShape.SHAPE_ID : String
```

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [RectangleShape](#)

Overrides: [SHAPE_ID](#)

```
rectangleShape.geom : Geometry
```

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [RectangleShape](#)

Overrides: [geom](#)

rectangleShape.label : String

Shape 라벨 텍스트

Kind: instance property of [RectangleShape](#)

Overrides: [label](#)

rectangleShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [RectangleShape](#)

rectangleShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [RectangleShape](#)

rectangleShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [RectangleShape](#)

rectangleShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [RectangleShape](#)

rectangleShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [RectangleShape](#)

rectangleShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [RectangleShape](#)

rectangleShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [RectangleShape](#)

rectangleShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [RectangleShape](#)

rectangleShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [RectangleShape](#)

rectangleShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [RectangleShape](#)

rectangleShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [RectangleShape](#)

rectangleShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [RectangleShape](#)

rectangleShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [RectangleShape](#)

rectangleShape.data : Object

도형의 데이터

Kind: instance property of [RectangleShape](#)

rectangleShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [RectangleShape](#)

rectangleShape.clone() ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [RectangleShape](#)

Returns: [IShape](#) - 복사된 인스턴스

rectangleShape.createShape() ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [RectangleShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

shape.SpotShape \Leftarrow [GeomShape](#)

Kind: static class of [shape](#)

Extends: [GeomShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.SpotShape](#) \Leftarrow [GeomShape](#)
 - [new OG.shape.SpotShape\(label\)](#)
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String
 - [.geom](#) : [Geometry](#)
 - [.label](#) : String
 - [.isCollapsed](#) : Boolean
 - [.SELECTABLE](#) : Boolean
 - [.MOVABLE](#) : Boolean
 - [.RESIZABLE](#) : Boolean
 - [.CONNECTABLE](#) : Boolean
 - [.ENABLE_FROM](#) : Boolean
 - [.ENABLE_TO](#) : Boolean

- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.clone\(\)](#) ⇒ [IShape](#)
- [.createShape\(\)](#) ⇒ *

new OG.shape.SpotShape(label)

SpotShape Shape

| Param | Type | Description |
|-------|--------|---------------|
| label | String | 라벨 [Optional] |

spotShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [SpotShape](#)

spotShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [SpotShape](#)

Overrides: [SHAPE_ID](#)

spotShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [SpotShape](#)

Overrides: [geom](#)

spotShape.label : String

Shape 라벨 텍스트

Kind: instance property of [SpotShape](#)

Overrides: [label](#)

spotShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [SpotShape](#)

spotShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [SpotShape](#)

spotShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [SpotShape](#)

spotShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [SpotShape](#)

spotShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [SpotShape](#)

spotShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [SpotShape](#)

spotShape.ENABLE_T0 : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [SpotShape](#)

spotShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [SpotShape](#)

spotShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [SpotShape](#)

spotShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [SpotShape](#)

spotShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [SpotShape](#)

spotShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [SpotShape](#)

spotShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [SpotShape](#)

spotShape.data : Object

도형의 데이터

Kind: instance property of [SpotShape](#)

spotShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [SpotShape](#)

`spotShape.clone()` ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [SpotShape](#)

Returns: [IShape](#) - 복사된 인스턴스

`spotShape.createShape()` ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [SpotShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

`shape.TextShape` ⇐ [IShape](#)

Kind: static class of [shape](#)

Extends: [IShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.TextShape](#) ⇐ [IShape](#)
 - [new OG.shape.TextShape\(text\)](#)

- [.text](#) : String
- [.angle](#) : Number
- [.TYPE](#) : String
- [.SHAPE_ID](#) : String
- [.geom](#) : [Geometry](#)
- [.label](#) : String
- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

```
new OG.shape.TextShape(text)
```

Text Shape

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|------|--------|-----|
| text | String | 텍스트 |
|------|--------|-----|

```
textShape.text : String
```

드로잉할 텍스트

Kind: instance property of [TextShape](#)

```
textShape.angle : Number
```

회전각도

Kind: instance property of [TextShape](#)

```
textShape.TYPE : String
```

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [TextShape](#)

Overrides: [TYPE](#)

```
textShape.SHAPE_ID : String
```

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [TextShape](#)

Overrides: [SHAPE_ID](#)

textShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [TextShape](#)

textShape.label : String

Shape 라벨 텍스트

Kind: instance property of [TextShape](#)

textShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [TextShape](#)

textShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [TextShape](#)

textShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [TextShape](#)

textShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [TextShape](#)

textShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [TextShape](#)

textShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [TextShape](#)

textShape.ENABLE_T0 : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [TextShape](#)

textShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [TextShape](#)

textShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [TextShape](#)

textShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [TextShape](#)

textShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [TextShape](#)

textShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [TextShape](#)

textShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [TextShape](#)

textShape.data : Object

도형의 데이터

Kind: instance property of [TextShape](#)

textShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [TextShape](#)

textShape.createShape() ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [TextShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

textShape.clone() ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [TextShape](#)

Overrides: [clone](#)

Returns: [IShape](#) - 복사된 인스턴스

shape.VerticalLaneShape ⇐ [GroupShape](#)

Kind: static class of [shape](#)

Extends: [GroupShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [.VerticalLaneShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.VerticalLaneShape\(label\)](#)
 - [.GROUP_DROPABLE](#) : Boolean

- [.GROUP_COLLAPSIBLE](#) : Boolean
- [.TYPE](#) : String
- [.SHAPE_ID](#) : String
- [.geom](#) : [Geometry](#)
- [.label](#) : String
- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

Vertical Swimlane Shape

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|-------|--------|----|
| label | String | 라벨 |
|-------|--------|----|

verticalLaneShape.GROUP_DROPABLE : Boolean

그룹핑 가능여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.GROUP_COLLAPSIBLE : Boolean

최소화 가능여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [VerticalLaneShape](#)

Overrides: [SHAPE_ID](#)

verticalLaneShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [VerticalLaneShape](#)

Overrides: [geom](#)

verticalLaneShape.label : String

Shape 라벨 텍스트

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.data : Object

도형의 데이터

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [VerticalLaneShape](#)

verticalLaneShape.createShape() ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [VerticalLaneShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

`verticalLaneShape.clone()` ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [VerticalLaneShape](#)

Returns: [IShape](#) - 복사된 인스턴스

`shape.VerticalPoolShape` ⇐ [GroupShape](#)

Kind: static class of [shape](#)

Extends: [GroupShape](#)

Requires: module:OG.common.*, module:OG.geometry.*

- [.VerticalPoolShape](#) ⇐ [GroupShape](#)
 - [new OG.shape.VerticalPoolShape\(label\)](#)
 - [.GROUP_DROPABLE](#) : Boolean
 - [.GROUP_COLLAPSIBLE](#) : Boolean
 - [.TYPE](#) : String
 - [.SHAPE_ID](#) : String

- [.geom](#) : [Geometry](#)
- [.label](#) : String
- [.isCollapsed](#) : Boolean
- [.SELECTABLE](#) : Boolean
- [.MOVABLE](#) : Boolean
- [.RESIZABLE](#) : Boolean
- [.CONNECTABLE](#) : Boolean
- [.ENABLE_FROM](#) : Boolean
- [.ENABLE_TO](#) : Boolean
- [.SELF_CONNECTABLE](#) : Boolean
- [.CONNECT_CLONEABLE](#) : Boolean
- [.CONNECT_REQUIRED](#) : Boolean
- [.CONNECT_STYLE_CHANGE](#) : Boolean
- [.DELETABLE](#) : Boolean
- [.LABEL_EDITABLE](#) : Boolean
- [.data](#) : Object
- [.textList](#) : Array
- [.createShape\(\)](#) ⇒ *
- [.clone\(\)](#) ⇒ [IShape](#)

```
new OG.shape.VerticalPoolShape(label)
```

Vertical Pool Shape

| Param | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|-------|--------|----|
| label | String | 라벨 |
|-------|--------|----|

verticalPoolShape.GROUP_DROPABLE : Boolean

그룹핑 가능여부

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.GROUP_COLLAPSIBLE : Boolean

최소화 가능여부

Kind: instance property of [VerticalPoolShape](#)

Overrides: [GROUP_COLLAPSIBLE](#)

verticalPoolShape.TYPE : String

Shape 유형(GEOM, TEXT, HTML, IMAGE, EDGE, GROUP)

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.SHAPE_ID : String

Shape 을 구분하는 Shape ID(Shape 클래스명과 일치)

Kind: instance property of [VerticalPoolShape](#)

Overrides: [SHAPE_ID](#)

verticalPoolShape.geom : [Geometry](#)

Shape 모양을 나타내는 공간기하객체(Geometry)

Kind: instance property of [VerticalPoolShape](#)

Overrides: [geom](#)

verticalPoolShape.label : String

Shape 라벨 텍스트

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.isCollapsed : Boolean

Shape 의 Collapse 여부

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.SELECTABLE : Boolean

선택 가능여부

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.MOVABLE : Boolean

이동 가능여부

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.RESIZABLE : Boolean

리사이즈 가능여부

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.CONNECTABLE : Boolean

연결 가능여부

Kind: instance property of [VerticalPoolShape](#)

Overrides: [CONNECTABLE](#)

verticalPoolShape.ENABLE_FROM : Boolean

From 연결 가능여부 (From(Shape) => To)

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.ENABLE_TO : Boolean

To 연결 가능여부 (From => To(Shape))

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.SELF_CONNECTABLE : Boolean

Self 연결 가능여부

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.CONNECT_CLONEABLE : Boolean

가이드에 자기자신을 복사하는 컨트롤러 여부.

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.CONNECT_REQUIRED : Boolean

드래그하여 연결시 연결대상 있는 경우에만 Edge 드로잉 처리 여부

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.CONNECT_STYLE_CHANGE : Boolean

드래그하여 연결시 그룹을 건너뛸때 스타일 변경 여부

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.DELETABLE : Boolean

가이드에 삭제 컨트롤러 여부

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.LABEL_EDITABLE : Boolean

라벨 수정여부

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.data : Object

도형의 데이터

Kind: instance property of [VerticalPoolShape](#)

verticalPoolShape.textList : Array

도형 선연결시 선연결 컨트롤러 목록

Kind: instance property of [VerticalPoolShape](#)

`verticalPoolShape.createShape()` ⇒ *

드로잉할 Shape 를 생성하여 반환한다.

Kind: instance method of [VerticalPoolShape](#)

Overrides: [createShape](#)

Returns: * - Shape 정보

`verticalPoolShape.clone()` ⇒ [IShape](#)

Shape 을 복사하여 새로운 인스턴스로 반환한다.

Kind: instance method of [VerticalPoolShape](#)

Returns: [IShape](#) - 복사된 인스턴스

`shape.bpmn` : object

Kind: static namespace of [shape](#)

`shape.elec` : object

Kind: static namespace of [shape](#)

`override(origclass, overrides)`

Adds a list of functions to the prototype of an existing class, overwriting any existing methods with the same name.

Usage:

```
Ext.override(MyClass, {  
  newMethod1: function(){  
    // etc.  
  },  
  newMethod2: function(foo){  
    // etc.  
  }  
});
```

Kind: global function

| Param | Type | Description |
|-----------|--------|--|
| origClass | Object | The class to override |
| overrides | Object | The list of functions to add to origClass. This should be specified as an object literal containing one or more methods. |