Shared function

CreateColPolygon

> **Note:** For this function to work correctly, get/set your bound points in an anti-clockwise fashion.

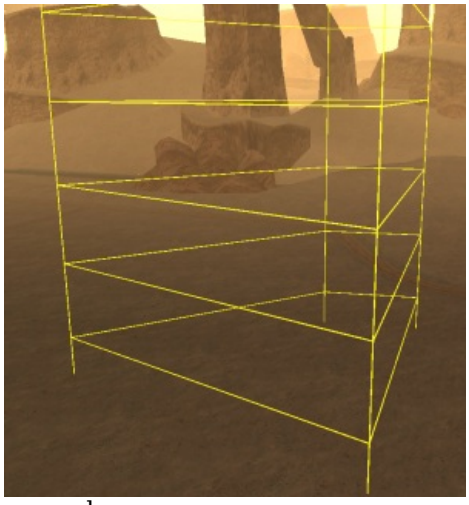
> **Note:** Even though this is a 2D colshape, isElementWithinColShape returns false if an element is in the area but below 0 Z height.

This function creates a collision polygon. See Wikipedia for a definition of a polygon. The first set of X Y of this shape is not part of the colshape bounds, so can set anywhere in the game world, however for performance, place it as close to the centre of the polygon as you can. It should be noted this shape is **2D**. There should be at least 3 bound points set.

> **Tip:** To visualize a colshape when writing scripts, use the client console command **showcol**

Syntax

```
colshape createColPolygon ( float fCenterX, float fCenterY, float fX1, float fY1, float fX2, float fY2, float fX3, float fY3,
 ... )
```



example

OOP Syntax Help! I don't understand this!

Method: *ColShape.Polygon(...)*

Required Arguments

- **fCenterX:** The X position of the collision polygon's position - the position that will be returned from getElementPosition.
- **fCenterY:** The Y position of the collision polygon's position - the position that will be returned from getElementPosition.
- **fX1:** The 1st X position of the collision polygon's bound point
- **fY1:** The 1st Y position of the collision polygon's bound point
- **fX2:** The 2nd X position of the collision polygon's bound point
- **fY2:** The 2nd Y position of the collision polygon's bound point
- **fX3:** The 3rd X position of the collision polygon's bound point
- **fY3:** The 3rd Y position of the collision polygon's bound point
- **... :** From the 3rd position you can have as many points as you require to create the colshape.

Returns

Returns a colshape element if successful, *false* if invalid arguments were passed to the function.