

## Módulo TQSGEO

Contém rotinas comuns de geometria analítica 2D comumente usadas na geração de desenhos.

`TQSGeo .SetPrecision (delx)`

Define a precisão para igualar pontos

`TQSGeo .SetAnglePrecision (delang)`

Define a precisão para igualar ângulos

`TQSGeo .Equals (x1, y1, x2, y2)`

Retorna (1) se dois pontos são iguais dentro da precisão atual.

`TQSGeo .Distance (x1, y1, x2, y2)`

Retorna a distância entre dois pontos

`TQSGeo .LineCoefficients (x1, y1, x2, y2)`

Retorna os coeficientes a,b,c da reta 1-2.

`TQSGeo .DistancePointLine (x11,y11,x21,y21,xpoint,ypoint)`

Retorna a distância entre uma reta e um ponto

`TQSGeo .Angle2p (x1, y1, x2, y2)`

Retorna o ângulo entre dois pontos entre 0..360 graus

`TQSGeo .Angle2r (x1, y1, x2, y2, x3, y3, x4, y4)`

Retorna o ângulo entre as retas 1-2 e 3-4 entre 0..180 graus

`TQSGeo .NormalizeAngle (angle)`

Retorna um ângulo normalizado entre 0 e 360 graus

`TQSGeo .Area (vecx, vecy)`

Retorna a área dos vetores `vecx[]` e `vecy []` com sinal positivo no sentido horário.

`TQSGeo .Perimeter (vecx, vecy)`

Retorna o perímetro de `vecx[]` e `vecy []`

`TQSGeo .GravityCenter (vecx, vecy)`

Retorna `x,y` do centro de gravidade e a área (com sinal) de `vecx[]` e `vecy[]`

`TQSGeo .PointInSegment (x, y, x1, y1, x2, y2)`

Retorna (1) se o ponto `x,y`, contido na reta 1-2, está contido no segmento de reta 1-2

O ponto `x,y` é projetado em 1-2 antes do teste

`TQSGeo .IntersectionByCoefs (a, b, c, d, e, f)`

Retorna `x, y, istat` da intersecção da reta 1 (coeficientes `a, b e c`) e a reta 2 (coeficientes `d, e, f`).

`istat != 0` se não houve intersecção.

TQSGeo **.Intersection2r** (x1, y1, x2, y2, x3, y3, x4, y4)

Retorna x, y, istat da intersecção da reta 1 ( x1, y1, x2, y2) e a reta 2 ( x3, y3, x4, y4).  
istat != 0 se não houve intersecção.

TQSGeo **.IntersectionSegment** (x1, y1, x2, y2, x3, y3, x4, y4)

Retorna x, y, istat da intersecção da reta 1 ( x1, y1, x2, y2) e a reta 2 ( x3, y3, x4, y4). A intersecção tem que acontecer dentro dos segmentos 1-2 e 3-4. istat != 0 se não houve intersecção.

TQSGeo **.Projection** (x1, y1, x2, y2, xp, yp)

Projeta o ponto xp, yp na reta 1-2. Retorna x, y, o ponto projetado

TQSGeo **.ParallelPoint** (x1, y1, x2, y2, dist)

Retorna o ponto xp, yp sobre uma reta ortogonal a 1-2 passando por 2 e uma distância positiva à direita de 1-2.

TQSGeo **.ParallelLine** (x1, y1, x2, y2, dist)

Calcula o a reta paralela a a 1-2 a uma distância positiva à direita de 1-2. Retorna a reta xp1, yp1, xp2, yp2

TQSGeo **.ParallelPolyline** (vecx, vecy, dist)

Modifica os vetores vecx [] e vecy [] por uma paralela a uma distância fornecida, positiva à direita das retas

TQSGeo **.Limits** (vecx, vecy)

Retorna os limites xmin, ymin, xmax, ymax dos vetores vecx [], vecy []

TQSGeo **.VectorProduct** (x1, y1, x2, y2, x3, y3, x4, y4)

Retorna o produto vetorial do vetor 1-2 por 3-4

TQSGeo **.PointProduct** (x1, y1, x2, y2, x3, y3, x4, y4)

Retorna o produto pontual do vetor 1-2 por 3-4

TQSGeo **.Versor** (x1, y1, x2, y2)

Retorna o versor vx, vy do vetor 1-2

TQSGeo **.PointInArea** (x, y, vecx, vecy)

Verifica se o ponto x, y está dentro da poligonal vecx [], vecy []. Retorna (1) se estiver.

TQSGeo **.MirrorInit** (x1, y1, x2, y2)

Define a linha de espelhamento 1-2 que será usada pela função Mirror. Retorna o ângulo original e o final de espelhamento da reta 1-2 em graus.

TQSGeo **.Mirror** (xe, ye)

Retorna o ponto xe, ye espelhado pela linha definida em MirrorInit.

TQSGeo **.Rotate** (x, y, ang, orgx, orgy)

Retorna o ponto x, y rodado por ang graus em torno de orgx, orgy.

## Programa TSTGeo.py de teste de TQSGeo

O programa TSTGeo.py testa as diversas funções do módulo TQSGeo, e imprime os resultados em uma janela TQS.