

James Fry's Quick Reference for GCLC Language

| | |
|---|---|
| number name constant | |
| point p_new x y | based on "dim" coordinates |
| point p_new x_initial y_initial x_final y_final | |
| line L_new p_end p_end | |
| circle name p_cen point | |
| getx name point | |
| gety name point | |
| distance name p1 p2 | |
| angle name p1 p_vertex p2 | |
| random id between 0 and 1 | |
| expression n_id { formula } | <p>Use the following operators:</p> <p>+, -, *, /, ==, !=, <, <=, >, >=, &&, , abs, ceil, floor, sin, asin, cos, acos, tan, atan, sinh, cosh, tanh, sqrt, exp, pow, log, log10, min (two arguments), max (two arguments), ite(comparison, true_const, false_const)</p> <p>example: expression size { 3 + ite(divs<2, 2, 3) }</p> |
| while {comparison} { statements } | |
| intersec p_new line line | |
| intersec2 p_new1 p_new2 circle line | |
| intersec2 p_new1 p_new2 line circle | |
| midpoint p_new p_id1 p_id2 | |
| perp L_new p line | |
| bis L_new p1 p_vertex p3 | line bisecting a 3-point angle |
| med L_new p_id1 p_id2 | perpendicular line between points |
| parallel L_new p_thru line | |
| translate p_new point_from point_to p_from | |
| rotate p_new p_center angle p_from | |
| sim p_new point p_from | |
| sim p_new line p_from | |
| sim p_new circle p_from | |
| turtle p_new p_from p_vertex angle dist_vertex | |
| towards p_new p_from p_to fraction_to_travel | |
| drawpoint p_id | |
| drawsegment p_id1 p_id2 | |
| drawline L_id | clipped to defined area |
| drawline p_id1 p_id2 | |
| drawdashsegment p_id1 p_id2 | use dash |

| | |
|--|--|
| drawvector p_id1 p_id2 | |
| drawcircle c_id | use circleprecision |
| drawcircle center p | |
| drawdashcircle p_id1 p_id2 | |
| drawarc center p angle | |
| drawdasharc center p angle | |
| drawellipse center p_axis p | |
| drawdashellipse center p_axis p | use circleprecision |
| drawellipsearc center p_axis p angle | |
| drawdashellipsearc center p_axis p angle | |
| drawpolygon center point n_sides | |
| | |
| cmark_lt p_id also _l, _lb, _t, _b, _rt, _r, _rb | draw the point and name it |
| mark_lt p_id | name the point but don't draw its circle |
| printat_lt p_id text | |
| area n1 n2 n3 n4 | visible area |
| dim width height | |
| color red green blue | |
| fontsize n | |
| circleprecision num_segments | |
| linethickness mm | |
| double thickness | |
| normal normal thickness | |
| dash mm | |
| dmc mm distance between point and its name | |
| mcr mm radius of circle marking point | |
| | |
| ang_picture xlb ylb xrt yrt | |
| ang_origin x y | |
| ang_unit mm | |
| ang_point p_new x y | |
| ang_line l_new a b c | |
| ang_conic new a b c d e f | where $ax^2+2bxy+cy^2+2dx+2ey+f=0$ |
| ang_intersec2 p1_new p2_new line conic | |
| ang_intersec2 p1_new p2_new conic line | |
| ang_tangent L_new p conic | |
| ang_drawline line | |
| ang_drawline_p p1 p2 | |
| ang_drawconic conic_id | |
| ang_drawdashconic conic_id | use conicprecision |
| ang_drawsystem | |
| ang_drawsystem0 | no integer points |
| ang_drawsystem1 | small dashes |
| ang_drawsystem_a | denotes the axes by x and y |
| ang_drawsystem0_a | denotes the axes by x and y |
| ang_drawsystem1_a | denotes the axes by x and y |
| ang_drawparametric_curve n_input | |
| { initial_exp; condition_exp; recalc } | |
| { x_expression; y_expression } | |
| ang_conicprecision n_segments | |
| | |
| animation_frames n_frames frames_per_second | |
| trace p red green blue | |