Grease Pencil Operators

bpy.ops.grease_pencil.**active_frame_delete(*, all=False)**

Delete the active Grease Pencil frame(s)

PARAMETERS:

all (*boolean, (optional)*) – Delete all, Delete active keyframes of all layers

bpy.ops.grease_pencil.**bake_grease_pencil_animation(*, frame_start=1, frame_end=250, step=1, only_selected=False, frame_target=1, project_type='KEEP')**

Bake Grease Pencil object transform to Grease Pencil keyframes

PARAMETERS:

- **frame_start** (*int in [1, 100000], (optional)*) – Start Frame, The start frame
- **frame_end** (*int in [1, 100000], (optional)*) – End Frame, The end frame of animation
- **step** (*int in [1, 100], (optional)*) – Step, Step between generated frames
- **only_selected** (*boolean, (optional)*) – Only Selected Keyframes, Convert only selected keyframes
- **frame_target** (*int in [1, 100000], (optional)*) – Target Frame, Destination frame
- **project_type** (*enum in ['KEEP', 'FRONT', 'SIDE', 'TOP', 'VIEW', 'CURSOR'], (optional)*) –
  Projection Type
  - `KEEP` No Reproject.
  - `FRONT` Front – Reproject the strokes using the X-Z plane.
  - `SIDE` Side – Reproject the strokes using the Y-Z plane.
  - `TOP` Top – Reproject the strokes using the X-Y plane.
  - `VIEW` View – Reproject the strokes to end up on the same plane, as if drawn from the current viewpoint using ‘Cursor’ Stroke Placeme
  - `CURSOR` Cursor – Reproject the strokes using the orientation of 3D cursor.

bpy.ops.grease_pencil.**brush_stroke(*, stroke=None, mode='NORMAL', pen_flip=False)**

Draw a new stroke in the active Grease Pencil object

PARAMETERS:

- **stroke** (`bpy_prop_collection` of `OperatorStrokeElement`, (optional)) – Stroke
- **mode** (*enum in ['NORMAL', 'INVERT', 'SMOOTH', 'ERASE'], (optional)*) –
  Stroke Mode, Action taken when a paint stroke is made
  - `NORMAL` Regular – Apply brush normally.
  - `INVERT` Invert – Invert action of brush for duration of stroke.
  - `SMOOTH` Smooth – Switch brush to smooth mode for duration of stroke.
  - `ERASE` Erase – Switch brush to erase mode for duration of stroke.
- **pen_flip** (*boolean, (optional)*) – Pen Flip, Whether a tablet’s eraser mode is being used

bpy.ops.grease_pencil.**caps_set(*, type='ROUND')**

Change curve caps mode (rounded or flat)

PARAMETERS:

type (*enum in ['ROUND', 'FLAT', 'START', 'END'], (optional)*) –

Type

- `ROUND` Rounded – Set as default rounded.
- `FLAT` Flat.
- `START` Toggle Start.
- `END` Toggle End

- END Toggle End.

bpy.ops.grease_pencil.**clean_loose(*, limit=1)**

Remove loose points

PARAMETERS:

limit (*int in [1, inf], (optional)*) – Limit, Number of points to consider stroke as loose

bpy.ops.grease_pencil.**copy()**

Copy the selected Grease Pencil points or strokes to the internal clipboard

bpy.ops.grease_pencil.**cyclical_set(*, type='TOGGLE', subdivide_cyclic_segment=True)**

Close or open the selected stroke adding a segment from last to first point

PARAMETERS:

- **type** (*enum in ['CLOSE', 'OPEN', 'TOGGLE'], (optional)*) – Type
- **subdivide_cyclic_segment** (*boolean, (optional)*) – Match Point Density, Add point in the new segment to keep the same density

bpy.ops.grease_pencil.**delete()**

Delete selected strokes or points

bpy.ops.grease_pencil.**delete_breakdown()**

Remove breakdown frames generated by interpolating between two Grease Pencil frames

bpy.ops.grease_pencil.**delete_frame(*, type='ACTIVE_FRAME')**

Delete Grease Pencil Frame(s)

PARAMETERS:

type (*enum in ['ACTIVE_FRAME', 'ALL_FRAMES'], (optional)*) –

Type, Method used for deleting Grease Pencil frames

- ACTIVE_FRAME Active Frame – Deletes current frame in the active layer.
- ALL_FRAMES All Active Frames – Delete active frames for all layers.

bpy.ops.grease_pencil.**dissolve(*, type='POINTS')**

Delete selected points without splitting strokes

PARAMETERS:

type (*enum in ['POINTS', 'BETWEEN', 'UNSELECT'], (optional)*) –

Type, Method used for dissolving stroke points

- POINTS Dissolve – Dissolve selected points.
- BETWEEN Dissolve Between – Dissolve points between selected points.
- UNSELECT Dissolve Unselect – Dissolve all unselected points.

bpy.ops.grease_pencil.**duplicate()**

Duplicate the selected points

bpy.ops.grease_pencil.**duplicate_move(*, GREASE_PENCIL_OT_duplicate=None, TRANSFORM_OT_translate=None)**

Make copies of the selected Grease Pencil strokes and move them

PARAMETERS:

- **GREASE_PENCIL_OT_duplicate** (GREASE_PENCIL_OT_duplicate, (optional)) – Duplicate, Duplicate the selected points
- **TRANSFORM_OT_translate** (TRANSFORM_OT_translate, (optional)) – Move, Move selected items

bpy.ops.grease_pencil.**erase_box(*, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True)**

Erase points in the box region

Erase points in the box region

PARAMETERS:

- **xmin** (*int in [-inf, inf], (optional)*) – X Min
- **xmax** (*int in [-inf, inf], (optional)*) – X Max
- **ymin** (*int in [-inf, inf], (optional)*) – Y Min
- **ymax** (*int in [-inf, inf], (optional)*) – Y Max
- **wait_for_input** (*boolean, (optional)*) – Wait for Input

bpy.ops.grease_pencil.**erase_lasso(*, path=None, use_smooth_stroke=False, smooth_stroke_factor=0.75, smooth_stroke_radius=35)**

Erase points in the lasso region

PARAMETERS:

- **path** (`bpy_prop_collection` of `OperatorMousePath` , (optional)) – Path
- **use_smooth_stroke** (*boolean, (optional)*) – Stabilize Stroke, Selection lags behind mouse and follows a smoother path
- **smooth_stroke_factor** (*float in [0.5, 0.99], (optional)*) – Smooth Stroke Factor, Higher values gives a smoother stroke
- **smooth_stroke_radius** (*int in [10, 200], (optional)*) – Smooth Stroke Radius, Minimum distance from last point before selection continues

bpy.ops.grease_pencil.**extrude()**

Extrude the selected points

bpy.ops.grease_pencil.**extrude_move(*, GREASE_PENCIL_OT_extrude=None, TRANSFORM_OT_translate=None)**

Extrude selected points and move them

PARAMETERS:

- **GREASE_PENCIL_OT_extrude** (`GREASE_PENCIL_OT_extrude` , (optional)) – Extrude Stroke Points, Extrude the selected poi
- **TRANSFORM_OT_translate** (`TRANSFORM_OT_translate` , (optional)) – Move, Move selected items

bpy.ops.grease_pencil.**fill(*, invert=False, precision=False)**

Fill with color the shape formed by strokes

PARAMETERS:

- **invert** (*boolean, (optional)*) – Invert, Find boundary of unfilled instead of filled regions
- **precision** (*boolean, (optional)*) – Precision, Use precision movement for extension lines

bpy.ops.grease_pencil.**frame_clean_duplicate(*, selected=False)**

Remove any keyframe that is a duplicate of the previous one

PARAMETERS:

- **selected** (*boolean, (optional)*) – Selected, Only delete selected keyframes

bpy.ops.grease_pencil.**frame_duplicate(*, all=False)**

Make a copy of the active Grease Pencil frame(s)

PARAMETERS:

- **all** (*boolean, (optional)*) – Duplicate all, Duplicate active keyframes of all layer

bpy.ops.grease_pencil.**insert_blank_frame(*, all_layers=False, duration=0)**

Insert a blank frame on the current scene frame

PARAMETERS:

- **all_layers** (*boolean, (optional)*) – All Layers, Insert a blank frame in all editable layers
- **duration** (*int in [0, 1048574], (optional)*) – Duration

bpy.ops.grease_pencil.**interpolate(*, shift=0.0, layers='ACTIVE', exclude_breakdowns=False, use_selection=False, flip='AUTO', smooth_steps=1, smooth_factor=0.0)**

Interpolate Grease Pencil strokes between frames

PARAMETERS:

- **shift** (*float in [-1, 1], (optional)*) – Shift, Bias factor for which frame has more influence on the interpolated strokes
- **layers** (*enum in ['ACTIVE', 'ALL'], (optional)*) – Layer, Layers included in the interpolation
- **exclude_breakdowns** (*boolean, (optional)*) – Exclude Breakdowns, Exclude existing Breakdowns keyframes as interpolation extremes
- **use_selection** (*boolean, (optional)*) – Use Selection, Use only selected strokes for interpolating
- **flip** (*enum in ['NONE', 'FLIP', 'AUTO'], (optional)*) – Flip Mode, Invert destination stroke to match start and end with source stroke
- **smooth_steps** (*int in [1, 3], (optional)*) – Iterations, Number of times to smooth newly created strokes
- **smooth_factor** (*float in [0, 2], (optional)*) – Smooth, Amount of smoothing to apply to interpolated strokes, to reduce jitter/noise

bpy.ops.grease_pencil.**interpolate_sequence(*, step=1, layers='ACTIVE', exclude_breakdowns=False, use_selection=False, flip='AUTO', smooth_steps=1, smooth_factor=0.0, type='LINEAR', easing='EASE_IN', back=1.702, amplitude=0.15, period=0.15)**

Generate ‘in-betweens’ to smoothly interpolate between Grease Pencil frames

PARAMETERS:

- **step** (*int in [1, 1048574], (optional)*) – Step, Number of frames between generated interpolated frames
- **layers** (*enum in ['ACTIVE', 'ALL'], (optional)*) – Layer, Layers included in the interpolation
- **exclude_breakdowns** (*boolean, (optional)*) – Exclude Breakdowns, Exclude existing Breakdowns keyframes as interpolation extremes
- **use_selection** (*boolean, (optional)*) – Use Selection, Use only selected strokes for interpolating
- **flip** (*enum in ['NONE', 'FLIP', 'AUTO'], (optional)*) – Flip Mode, Invert destination stroke to match start and end with source stroke
- **smooth_steps** (*int in [1, 3], (optional)*) – Iterations, Number of times to smooth newly created strokes
- **smooth_factor** (*float in [0, 2], (optional)*) – Smooth, Amount of smoothing to apply to interpolated strokes, to reduce jitter/noise
- **type** (*enum in ['LINEAR', 'CUSTOM', 'SINE', 'QUAD', 'CUBIC', 'QUART', 'QUINT', 'EXPO', 'CIRC', 'BACK', 'BOUNCE', 'ELASTIC'], (optional)*) –

  Type, Interpolation method to use the next time ‘Interpolate Sequence’ is run

  - `LINEAR` Linear – Straight-line interpolation between A and B (i.e. no ease in/out).
  - `CUSTOM` Custom – Custom interpolation defined using a curve map.
  - `SINE` Sinusoidal – Sinusoidal easing (weakest, almost linear but with a slight curvature).
  - `QUAD` Quadratic – Quadratic easing.
  - `CUBIC` Cubic – Cubic easing.
  - `QUART` Quartic – Quartic easing.
  - `QUINT` Quintic – Quintic easing.
  - `EXPO` Exponential – Exponential easing (dramatic).
  - `CIRC` Circular – Circular easing (strongest and most dynamic).
  - `BACK` Back – Cubic easing with overshoot and settle.
  - `BOUNCE` Bounce – Exponentially decaying parabolic bounce, like when objects collide.
  - `ELASTIC` Elastic – Exponentially decaying sine wave, like an elastic band.

- **easing** (enum in Beztriple Interpolation Easing Items, (optional)) – Easing, Which ends of the segment between the preceding and following Grease Pencil frames easing interpolation is applied to
- **back** (*float in [0, inf], (optional)*) – Back, Amount of overshoot for ‘back’ easing
- **amplitude** (*float in [0, inf], (optional)*) – Amplitude, Amount to boost elastic bounces for ‘elastic’ easing
- **period** (*float in [-inf, inf], (optional)*) – Period, Time between bounces for elastic easing

bpy.ops.grease_pencil.**join_selection(*, type='JOIN')**

New stroke from selected points/strokes

PARAMETERS:

type (*enum in ['JOINCOPY', 'JOIN'], (optional)*) –

Type, Defines how the operator will behave on the selection in the active layer

- `JOINCOPY` Join and Copy – Copy the selection in the new stroke.
- `JOIN` Join – Move the selection to the new stroke.

bpy.ops.grease_pencil.**layer_active(*, layer=0)**

Set the active Grease Pencil layer

PARAMETERS:

layer (*int in [0, inf], (optional)*) – Grease Pencil Layer

bpy.ops.grease_pencil.**layer_add(*, new_layer_name='Layer')**

Add a new Grease Pencil layer in the active object

PARAMETERS:

new_layer_name (*string, (optional, never None)*) – Name, Name of the new layer

bpy.ops.grease_pencil.**layer_duplicate(*, empty_keyframes=False)**

Make a copy of the active Grease Pencil layer

PARAMETERS:

empty_keyframes (*boolean, (optional)*) – Empty Keyframes, Add Empty Keyframes

bpy.ops.grease_pencil.**layer_duplicate_object(*, only_active=True, mode='ALL')**

Make a copy of the active Grease Pencil layer to selected object

PARAMETERS:

- **only_active** (*boolean, (optional)*) – Only Active, Copy only active Layer, uncheck to append all layers
- **mode** (*enum in ['ALL', 'ACTIVE'], (optional)*) – Mode

bpy.ops.grease_pencil.**layer_group_add(*, new_layer_group_name='')**

Add a new Grease Pencil layer group in the active object

PARAMETERS:

new_layer_group_name (*string, (optional, never None)*) – Name, Name of the new layer group

bpy.ops.grease_pencil.**layer_group_color_tag(*, color_tag='COLOR1')**

Change layer group icon

PARAMETERS:

color_tag (*enum in ['NONE', 'COLOR1', 'COLOR2', 'COLOR3', 'COLOR4', 'COLOR5', 'COLOR6', 'COLOR7', 'COLOR8'], (optional)*) Color Tag

bpy.ops.grease_pencil.**layer_group_remove(*, keep_children=False)**

Remove Grease Pencil layer group in the active object

PARAMETERS:

keep_children (*boolean, (optional)*) – Keep children nodes, Keep the children nodes of the group and only delete the group itself

bpy.ops.grease_pencil.**layer_hide(*, unselected=False)**

Hide selected/unselected Grease Pencil layers

PARAMETERS:

unselected (*boolean, (optional)*) – Unselected, Hide unselected rather than selected layers

bpy.ops.grease_pencil.**layer_isolate(*, affect_visibility=False)**

Make only active layer visible/editable

PARAMETERS:

affect_visibility (*boolean, (optional)*) – Affect Visibility, Also affect the visibility

affect_visibility (*boolean, (optional)*) – Affect Visibility, Also affect the visibility

bpy.ops.grease_pencil.**layer_lock_all(*, lock=True)**

Lock all Grease Pencil layers to prevent them from being accidentally modified

PARAMETERS:

lock (*boolean, (optional)*) – Lock Value, Lock/Unlock all layers

bpy.ops.grease_pencil.**layer_mask_add(*, name='')**

Add new layer as masking

PARAMETERS:

name (*string, (optional, never None)*) – Layer, Name of the layer

bpy.ops.grease_pencil.**layer_mask_remove()**

Remove Layer Mask

bpy.ops.grease_pencil.**layer_mask_reorder(*, direction='UP')**

Reorder the active Grease Pencil mask layer up/down in the list

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction

bpy.ops.grease_pencil.**layer_merge(*, mode='ACTIVE')**

Combine layers based on the mode into one layer

PARAMETERS:

mode (*enum in ['ACTIVE', 'GROUP', 'ALL'], (optional)*) –

Mode

- `ACTIVE` Active – Combine the active layer with the layer just below (if it exists).
- `GROUP` Group – Combine layers in the active group into a single layer.
- `ALL` All – Combine all layers into a single layer.

bpy.ops.grease_pencil.**layer_move(*, direction='UP')**

Move the active Grease Pencil layer or Group

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction

bpy.ops.grease_pencil.**layer_remove()**

Remove the active Grease Pencil layer

bpy.ops.grease_pencil.**layer_reveal()**

Show all Grease Pencil layers

bpy.ops.grease_pencil.**material_copy_to_object(*, only_active=True)**

Append Materials of the active Grease Pencil to other object

PARAMETERS:

only_active (*boolean, (optional)*) – Only Active, Append only active material, uncheck to append all materials

bpy.ops.grease_pencil.**material_hide(*, invert=False)**

Hide active/inactive Grease Pencil material(s)

PARAMETERS:

invert (*boolean, (optional)*) – Invert, Hide inactive materials instead of the active one

bpy.ops.grease_pencil.**material_isolate(*, affect_visibility=False)**

Toggle whether the active material is the only one that is editable and/or visible

PARAMETERS:

affect_visibility (*boolean, (optional)*) – Affect Visibility, In addition to toggling the editability, also affect the visibility

bpy.ops.grease_pencil.**material_lock_all()**

Lock all Grease Pencil materials to prevent them from being accidentally modified

bpy.ops.grease_pencil.**material_lock_unselected()**

Lock any material not used in any selected stroke

bpy.ops.grease_pencil.**material_lock_unused()**

Lock and hide any material not used

bpy.ops.grease_pencil.**material_reveal()**

Unhide all hidden Grease Pencil materials

bpy.ops.grease_pencil.**material_select(*, deselect=False)**

Select/Deselect all Grease Pencil strokes using current material

PARAMETERS:

deselect (*boolean, (optional)*) – Deselect, Unselect strokes

bpy.ops.grease_pencil.**material_unlock_all()**

Unlock all Grease Pencil materials so that they can be edited

bpy.ops.grease_pencil.**move_to_layer(*, target_layer_name='', add_new_layer=False)**

Move selected strokes to another layer

PARAMETERS:

- **target_layer_name** (*string, (optional, never None)*) – Name, Target Grease Pencil Layer
- **add_new_layer** (*boolean, (optional)*) – New Layer, Move selection to a new layer

bpy.ops.grease_pencil.**paintmode_toggle(*, back=False)**

Enter/Exit paint mode for Grease Pencil strokes

PARAMETERS:

back (*boolean, (optional)*) – Return to Previous Mode, Return to previous mode

bpy.ops.grease_pencil.**paste(*, type='ACTIVE', paste_back=False, keep_world_transform=False)**

Paste Grease Pencil points or strokes from the internal clipboard to the active layer

PARAMETERS:

- **type** (*enum in ['ACTIVE', 'LAYER'], (optional)*) – Type
- **paste_back** (*boolean, (optional)*) – Paste on Back, Add pasted strokes behind all strokes
- **keep_world_transform** (*boolean, (optional)*) – Keep World Transform, Keep the world transform of strokes from the clipboard unchange

bpy.ops.grease_pencil.**primitive_arc(*, subdivision=62, type='ARC')**

Create predefined Grease Pencil stroke arcs

PARAMETERS:

- **subdivision** (*int in [0, inf], (optional)*) – Subdivisions, Number of subdivisions per segment
- **type** (*enum in ['BOX', 'LINE', 'POLYLINE', 'CIRCLE', 'ARC', 'CURVE'], (optional)*) – Type, Type of shape

bpy.ops.grease_pencil.**primitive_box(*, subdivision=3, type='BOX')**

Create predefined Grease Pencil stroke boxes

PARAMETERS:

- **subdivision** (*int in [0, inf], (optional)*) – Subdivisions, Number of subdivisions per segment
- **type** (*enum in ['BOX', 'LINE', 'POLYLINE', 'CIRCLE', 'ARC', 'CURVE'], (optional)*) – Type, Type of shape

bpy.ops.grease_pencil.**primitive_circle(*, subdivision=94, type='CIRCLE')**

Create predefined Grease Pencil stroke circles

PARAMETERS:

- **subdivision** (*int in [0, inf], (optional)*) – Subdivisions, Number of subdivisions per segment
- **type** (*enum in ['BOX', 'LINE', 'POLYLINE', 'CIRCLE', 'ARC', 'CURVE'], (optional)*) – Type, Type of shape

bpy.ops.grease_pencil.**primitive_curve(*, subdivision=62, type='CURVE')**

Create predefined Grease Pencil stroke curve shapes

PARAMETERS:

- **subdivision** (*int in [0, inf], (optional)*) – Subdivisions, Number of subdivisions per segment
- **type** (*enum in ['BOX', 'LINE', 'POLYLINE', 'CIRCLE', 'ARC', 'CURVE'], (optional)*) – Type, Type of shape

bpy.ops.grease_pencil.**primitive_line(*, subdivision=6, type='LINE')**

Create predefined Grease Pencil stroke lines

PARAMETERS:

- **subdivision** (*int in [0, inf], (optional)*) – Subdivisions, Number of subdivisions per segment
- **type** (*enum in ['BOX', 'LINE', 'POLYLINE', 'CIRCLE', 'ARC', 'CURVE'], (optional)*) – Type, Type of shape

bpy.ops.grease_pencil.**primitive_polyline(*, subdivision=6, type='POLYLINE')**

Create predefined Grease Pencil stroke polylines

PARAMETERS:

- **subdivision** (*int in [0, inf], (optional)*) – Subdivisions, Number of subdivisions per segment
- **type** (*enum in ['BOX', 'LINE', 'POLYLINE', 'CIRCLE', 'ARC', 'CURVE'], (optional)*) – Type, Type of shape

bpy.ops.grease_pencil.**reorder(*, direction='TOP')**

Change the display order of the selected strokes

PARAMETERS:

direction (*enum in ['TOP', 'UP', 'DOWN', 'BOTTOM'], (optional)*) – Direction

bpy.ops.grease_pencil.**reproject(*, type='VIEW', keep_original=False, offset=0.0)**

Reproject the selected strokes from the current viewpoint as if they had been newly drawn (e.g. to fix problems from accidental 3D cursor movem or accidental viewport changes, or for matching deforming geometry)

PARAMETERS:

- **type** (*enum in ['FRONT', 'SIDE', 'TOP', 'VIEW', 'SURFACE', 'CURSOR'], (optional)*) –
  Projection Type
  - `FRONT` Front – Reproject the strokes using the X-Z plane.
  - `SIDE` Side – Reproject the strokes using the Y-Z plane.
  - `TOP` Top – Reproject the strokes using the X-Y plane.
  - `VIEW` View – Reproject the strokes to end up on the same plane, as if drawn from the current viewpoint using 'Cursor' Stroke Placeme
  - `SURFACE` Surface – Reproject the strokes on to the scene geometry, as if drawn using 'Surface' placement.
  - `CURSOR` Cursor – Reproject the strokes using the orientation of 3D cursor.
- **keep_original** (*boolean, (optional)*) – Keep Original, Keep original strokes and create a copy before reprojecting

- **offset** (*float in [0, 10], (optional)*) – Surface Offset

bpy.ops.grease_pencil.**reset_uvs()**

Reset UV transformation to default values

bpy.ops.grease_pencil.**sculpt_paint(*, stroke=None, mode='NORMAL', pen_flip=False)**

Sculpt strokes in the active Grease Pencil object

PARAMETERS:

- **stroke** (`bpy_prop_collection` of `OperatorStrokeElement`, (optional)) – Stroke
- **mode** (*enum in ['NORMAL', 'INVERT', 'SMOOTH', 'ERASE'], (optional)*) –
  Stroke Mode, Action taken when a paint stroke is made
  - `NORMAL` Regular – Apply brush normally.
  - `INVERT` Invert – Invert action of brush for duration of stroke.
  - `SMOOTH` Smooth – Switch brush to smooth mode for duration of stroke.
  - `ERASE` Erase – Switch brush to erase mode for duration of stroke.
- **pen_flip** (*boolean, (optional)*) – Pen Flip, Whether a tablet's eraser mode is being used

bpy.ops.grease_pencil.**sculptmode_toggle(*, back=False)**

Enter/Exit sculpt mode for Grease Pencil strokes

PARAMETERS:

back (*boolean, (optional)*) – Return to Previous Mode, Return to previous mode

bpy.ops.grease_pencil.**select_all(*, action='TOGGLE')**

(De)select all visible strokes

PARAMETERS:

action (*enum in ['TOGGLE', 'SELECT', 'DESELECT', 'INVERT'], (optional)*) –

Action, Selection action to execute

- `TOGGLE` Toggle – Toggle selection for all elements.
- `SELECT` Select – Select all elements.
- `DESELECT` Deselect – Deselect all elements.
- `INVERT` Invert – Invert selection of all elements.

bpy.ops.grease_pencil.**select_alternate(*, deselect_ends=False)**

Select alternated points in strokes with already selected points

PARAMETERS:

deselect_ends (*boolean, (optional)*) – Deselect Ends, (De)select the first and last point of each stroke

bpy.ops.grease_pencil.**select_ends(*, amount_start=0, amount_end=1)**

Select end points of strokes

PARAMETERS:

- **amount_start** (*int in [0, inf], (optional)*) – Amount Start, Number of points to select from the start
- **amount_end** (*int in [0, inf], (optional)*) – Amount End, Number of points to select from the end

bpy.ops.grease_pencil.**select_less()**

Shrink the selection by one point

bpy.ops.grease_pencil.**select_linked()**

Select all points in curves with any point selection

bpy.ops.grease_pencil.**select_more()**

Grow the selection by one point

bpy.ops.grease_pencil.**select_random(*, ratio=0.5, seed=0, action='SELECT')**

Selects random points from the current strokes selection

PARAMETERS:

- **ratio** (*float in [0, 1], (optional)*) – Ratio, Portion of items to select randomly
- **seed** (*int in [0, inf], (optional)*) – Random Seed, Seed for the random number generator
- **action** (*enum in ['SELECT', 'DESELECT'], (optional)*) –

  Action, Selection action to execute

  - `SELECT` Select – Select all elements.
  - `DESELECT` Deselect – Deselect all elements.

bpy.ops.grease_pencil.**select_similar(*, mode='LAYER', threshold=0.1)**

Select all strokes with similar characteristics

PARAMETERS:

- **mode** (*enum in ['LAYER', 'MATERIAL', 'VERTEX_COLOR', 'RADIUS', 'OPACITY'], (optional)*) – Mode
- **threshold** (*float in [0, inf], (optional)*) – Threshold

bpy.ops.grease_pencil.**separate(*, mode='SELECTED')**

Separate the selected geometry into a new Grease Pencil object

PARAMETERS:

mode (*enum in ['SELECTED', 'MATERIAL', 'LAYER'], (optional)*) –

Mode

- `SELECTED` Selection – Separate selected geometry.
- `MATERIAL` By Material – Separate by material.
- `LAYER` By Layer – Separate by layer.

bpy.ops.grease_pencil.**set_active_material()**

Set the selected stroke material as the active material

bpy.ops.grease_pencil.**set_curve_resolution(*, resolution=12)**

Set resolution of selected curves

PARAMETERS:

resolution (*int in [0, 10000], (optional)*) – Resolution, The resolution to use for each curve segment

bpy.ops.grease_pencil.**set_curve_type(*, type='POLY', use_handles=False)**

Set type of selected curves

PARAMETERS:

- **type** (enum in Curves Type Items, (optional)) – Type, Curve type
- **use_handles** (*boolean, (optional)*) – Handles, Take handle information into account in the conversion

bpy.ops.grease_pencil.**set_handle_type(*, type='AUTO')**

Set the handle type for bezier curves

PARAMETERS:

type (enum in Curves Handle Type Items, (optional)) – Type

bpy.ops.grease_pencil.**set_material(*, slot='DEFAULT')**

bpy.ops.grease_pencil.**set_material('', slot='DEFAULT')**

Set active material

PARAMETERS:

slot (*enum in ['DEFAULT'], (optional)*) – Material Slot

bpy.ops.grease_pencil.**set_selection_mode(*, mode='POINT')**

Change the selection mode for Grease Pencil strokes

PARAMETERS:

mode (enum in Grease Pencil Selectmode Items, (optional)) – Mode

bpy.ops.grease_pencil.**set_start_point()**

Select which point is the beginning of the curve

bpy.ops.grease_pencil.**set_uniform_opacity(*, opacity_stroke=1.0, opacity_fill=0.5)**

Set all stroke points to same opacity

PARAMETERS:

- **opacity_stroke** (*float in [0, 1], (optional)*) – Stroke Opacity
- **opacity_fill** (*float in [0, 1], (optional)*) – Fill Opacity

bpy.ops.grease_pencil.**set_uniform_thickness(*, thickness=0.1)**

Set all stroke points to same thickness

PARAMETERS:

thickness (*float in [0, 1000], (optional)*) – Thickness, Thickness

bpy.ops.grease_pencil.**snap_cursor_to_selected()**

Snap cursor to center of selected points

bpy.ops.grease_pencil.**snap_to_cursor(*, use_offset=True)**

Snap selected points/strokes to the cursor

PARAMETERS:

use_offset (*boolean, (optional)*) – With Offset, Offset the entire stroke instead of selected points only

bpy.ops.grease_pencil.**snap_to_grid()**

Snap selected points to the nearest grid points

bpy.ops.grease_pencil.**stroke_material_set(*, material='')**

Assign the active material slot to the selected strokes

PARAMETERS:

material (*string, (optional, never None)*) – Material, Name of the material

bpy.ops.grease_pencil.**stroke_merge_by_distance(*, threshold=0.001, use_unselected=False)**

Merge points by distance

PARAMETERS:

- **threshold** (*float in [0, 100], (optional)*) – Threshold
- **use_unselected** (*boolean, (optional)*) – Unselected, Use whole stroke, not only selected points

bpy.ops.grease_pencil.**stroke_reset_vertex_color(*, mode='BOTH')**

Reset vertex color for all or selected strokes

PARAMETERS:

mode (*enum in ['STROKE', 'FILL', 'BOTH'], (optional)*) – Mode

mode (*enum in ['STROKE', 'FILL', 'BOTH'], (optional)*) – Mode

bpy.ops.grease_pencil.**stroke_simplify(*, factor=0.01, length=0.05, distance=0.01, steps=1, mode='FIXED')**

Simplify selected strokes

PARAMETERS:

- **factor** (*float in [0, 100], (optional)*) – Factor
- **length** (*float in [0.01, 100], (optional)*) – Length
- **distance** (*float in [0, 100], (optional)*) – Distance
- **steps** (*int in [0, 50], (optional)*) – Steps
- **mode** (*enum in ['FIXED', 'ADAPTIVE', 'SAMPLE', 'MERGE'], (optional)*) –
  Mode, Method used for simplifying stroke points
  - `FIXED` Fixed – Delete alternating vertices in the stroke, except extremes.
  - `ADAPTIVE` Adaptive – Use a Ramer-Douglas-Peucker algorithm to simplify the stroke preserving main shape.
  - `SAMPLE` Sample – Re-sample the stroke with segments of the specified length.
  - `MERGE` Merge – Simplify the stroke by merging vertices closer than a given distance.

bpy.ops.grease_pencil.**stroke_smooth(*, iterations=10, factor=1.0, smooth_ends=False, keep_shape=False, smooth_position=True, smooth_radius=True, smooth_opacity=False)**

Smooth selected strokes

PARAMETERS:

- **iterations** (*int in [1, 100], (optional)*) – Iterations
- **factor** (*float in [0, 1], (optional)*) – Factor
- **smooth_ends** (*boolean, (optional)*) – Smooth Endpoints
- **keep_shape** (*boolean, (optional)*) – Keep Shape
- **smooth_position** (*boolean, (optional)*) – Position
- **smooth_radius** (*boolean, (optional)*) – Radius
- **smooth_opacity** (*boolean, (optional)*) – Opacity

bpy.ops.grease_pencil.**stroke_subdivide(*, number_cuts=1, only_selected=True)**

Subdivide between continuous selected points of the stroke adding a point half way between them

PARAMETERS:

- **number_cuts** (*int in [1, 32], (optional)*) – Number of Cuts
- **only_selected** (*boolean, (optional)*) – Selected Points, Smooth only selected points in the stroke

bpy.ops.grease_pencil.**stroke_subdivide_smooth(*, GREASE_PENCIL_OT_stroke_subdivide=None, GREASE_PENCIL_OT_stroke_smooth=None)**

Subdivide strokes and smooth them

PARAMETERS:

- **GREASE_PENCIL_OT_stroke_subdivide** (`GREASE_PENCIL_OT_stroke_subdivide`, (optional)) – Subdivide Stroke, Subdivide between continuous selected points of the stroke adding a point half way between them
- **GREASE_PENCIL_OT_stroke_smooth** (`GREASE_PENCIL_OT_stroke_smooth`, (optional)) – Smooth Stroke, Smooth selected strokes

bpy.ops.grease_pencil.**stroke_switch_direction()**

Change direction of the points of the selected strokes

bpy.ops.grease_pencil.**stroke_trim(*, path=None, use_smooth_stroke=False, smooth_stroke_factor=0.75, smooth_stroke_radius=35)**

Delete stroke points in between intersecting strokes

PARAMETERS:

- **path** (`bpy_prop_collection` of `OperatorMousePath`, (optional)) – Path
- **use_smooth_stroke** (*boolean, (optional)*) – Stabilize Stroke, Selection lags behind mouse and follows a smoother path
- **smooth_stroke_factor** (*float in [0.5, 0.99], (optional)*) – Smooth Stroke Factor, Higher values gives a smoother stroke
- **smooth_stroke_radius** (*int in [10, 200], (optional)*) – Smooth Stroke Radius, Minimum distance from last point before selection continues

bpy.ops.grease_pencil.**texture_gradient(*, xstart=0, xend=0, ystart=0, yend=0, flip=False, cursor=5)**

Draw a line to set the fill material gradient for the selected strokes

PARAMETERS:

- **xstart** (*int in [-inf, inf], (optional)*) – X Start
- **xend** (*int in [-inf, inf], (optional)*) – X End
- **ystart** (*int in [-inf, inf], (optional)*) – Y Start
- **yend** (*int in [-inf, inf], (optional)*) – Y End
- **flip** (*boolean, (optional)*) – Flip
- **cursor** (*int in [0, inf], (optional)*) – Cursor, Mouse cursor style to use during the modal operator

bpy.ops.grease_pencil.**trace_image(*, target='NEW', radius=0.01, threshold=0.5, turnpolicy='MINORITY', mode='SINGLE', use_current_frame=True, frame_number=0)**

Extract Grease Pencil strokes from image

PARAMETERS:

- **target** (*enum in ['NEW', 'SELECTED'], (optional)*) – Target Object, Target Grease Pencil
- **radius** (*float in [0.001, 1], (optional)*) – Radius
- **threshold** (*float in [0, 1], (optional)*) – Color Threshold, Determine the lightness threshold above which strokes are generated
- **turnpolicy** (*enum in ['FOREGROUND', 'BACKGROUND', 'LEFT', 'RIGHT', 'MINORITY', 'MAJORITY', 'RANDOM'], (optional)*) – Turn Policy, Determines how to resolve ambiguities during decomposition of bitmaps into paths
  - `FOREGROUND` Foreground – Prefers to connect foreground components.
  - `BACKGROUND` Background – Prefers to connect background components.
  - `LEFT` Left – Always take a left turn.
  - `RIGHT` Right – Always take a right turn.
  - `MINORITY` Minority – Prefers to connect the color that occurs least frequently in the local neighborhood of the current position.
  - `MAJORITY` Majority – Prefers to connect the color that occurs most frequently in the local neighborhood of the current position.
  - `RANDOM` Random – Choose pseudo-randomly.
- **mode** (*enum in ['SINGLE', 'SEQUENCE'], (optional)*) – Mode, Determines if trace simple image or full sequence
  - `SINGLE` Single – Trace the current frame of the image.
  - `SEQUENCE` Sequence – Trace full sequence.
- **use_current_frame** (*boolean, (optional)*) – Start At Current Frame, Trace Image starting in current image frame
- **frame_number** (*int in [0, 9999], (optional)*) – Trace Frame, Used to trace only one frame of the image sequence, set to zero to trace all

bpy.ops.grease_pencil.**vertex_brush_stroke(*, stroke=None, mode='NORMAL', pen_flip=False)**

Draw on vertex colors in the active Grease Pencil object

PARAMETERS:

- **stroke** (`bpy_prop_collection` of `OperatorStrokeElement`, (optional)) – Stroke
- **mode** (*enum in ['NORMAL', 'INVERT', 'SMOOTH', 'ERASE'], (optional)*) – Stroke Mode, Action taken when a paint stroke is made
  - `NORMAL` Regular – Apply brush normally.
  - `INVERT` Invert – Invert action of brush for duration of stroke.
  - `SMOOTH` Smooth – Switch brush to smooth mode for duration of stroke.

- `ERASE` Erase – Switch brush to erase mode for duration of stroke.

- **pen_flip** (*boolean, (optional)*) – Pen Flip, Whether a tablet's eraser mode is being used

bpy.ops.grease_pencil.**vertex_color_brightness_contrast(*, mode='BOTH', brightness=0.0, contrast=0.0)**

Adjust vertex color brightness/contrast

PARAMETERS:

- **mode** (*enum in ['STROKE', 'FILL', 'BOTH'], (optional)*) – Mode
- **brightness** (*float in [-1, 1], (optional)*) – Brightness
- **contrast** (*float in [-1, 1], (optional)*) – Contrast

bpy.ops.grease_pencil.**vertex_color_hsv(*, mode='BOTH', h=0.5, s=1.0, v=1.0)**

Adjust vertex color HSV values

PARAMETERS:

- **mode** (*enum in ['STROKE', 'FILL', 'BOTH'], (optional)*) – Mode
- **h** (*float in [0, 1], (optional)*) – Hue
- **s** (*float in [0, 2], (optional)*) – Saturation
- **v** (*float in [0, 2], (optional)*) – Value

bpy.ops.grease_pencil.**vertex_color_invert(*, mode='BOTH')**

Invert RGB values

PARAMETERS:

mode (*enum in ['STROKE', 'FILL', 'BOTH'], (optional)*) – Mode

bpy.ops.grease_pencil.**vertex_color_levels(*, mode='BOTH', offset=0.0, gain=1.0)**

Adjust levels of vertex colors

PARAMETERS:

- **mode** (*enum in ['STROKE', 'FILL', 'BOTH'], (optional)*) – Mode
- **offset** (*float in [-1, 1], (optional)*) – Offset, Value to add to colors
- **gain** (*float in [0, inf], (optional)*) – Gain, Value to multiply colors by

bpy.ops.grease_pencil.**vertex_color_set(*, mode='BOTH', factor=1.0)**

Set active color to all selected vertex

PARAMETERS:

- **mode** (*enum in ['STROKE', 'FILL', 'BOTH'], (optional)*) – Mode
- **factor** (*float in [0, 1], (optional)*) – Factor, Mix Factor

bpy.ops.grease_pencil.**vertex_group_normalize()**

Normalize weights of the active vertex group

bpy.ops.grease_pencil.**vertex_group_normalize_all(*, lock_active=True)**

Normalize the weights of all vertex groups, so that for each vertex, the sum of all weights is 1.0

PARAMETERS:

lock_active (*boolean, (optional)*) – Lock Active, Keep the values of the active group while normalizing others

bpy.ops.grease_pencil.**vertex_group_smooth(*, factor=0.5, repeat=1)**

Smooth the weights of the active vertex group

PARAMETERS:

- **factor** (*float in [0, 1], (optional)*) – Factor

- **repeat** (*int in [1, 10000], (optional)*) – Iterations

bpy.ops.grease_pencil.**vertexmode_toggle(*, back=False)**

Enter/Exit vertex paint mode for Grease Pencil strokes

PARAMETERS:

back (*boolean, (optional)*) – Return to Previous Mode, Return to previous mode

bpy.ops.grease_pencil.**weight_brush_stroke(*, stroke=None, mode='NORMAL', pen_flip=False)**

Draw weight on stroke points in the active Grease Pencil object

PARAMETERS:

- **stroke** (`bpy_prop_collection` of `OperatorStrokeElement`, (optional)) – Stroke
- **mode** (*enum in ['NORMAL', 'INVERT', 'SMOOTH', 'ERASE'], (optional)*) –
  Stroke Mode, Action taken when a paint stroke is made
  - `NORMAL` Regular – Apply brush normally.
  - `INVERT` Invert – Invert action of brush for duration of stroke.
  - `SMOOTH` Smooth – Switch brush to smooth mode for duration of stroke.
  - `ERASE` Erase – Switch brush to erase mode for duration of stroke.
- **pen_flip** (*boolean, (optional)*) – Pen Flip, Whether a tablet's eraser mode is being used

bpy.ops.grease_pencil.**weight_invert()**

Invert the weight of active vertex group

bpy.ops.grease_pencil.**weight_sample()**

Set the weight of the Draw tool to the weight of the vertex under the mouse cursor

bpy.ops.grease_pencil.**weight_toggle_direction()**

Toggle Add/Subtract for the weight paint draw tool

bpy.ops.grease_pencil.**weightmode_toggle(*, back=False)**

Enter/Exit weight paint mode for Grease Pencil strokes

PARAMETERS:

back (*boolean, (optional)*) – Return to Previous Mode, Return to previous mode