

Sculpt Operators

`bpy.ops.sculpt.brush_stroke(*, stroke=None, mode='NORMAL', pen_flip=False, override_location=False, ignore_background_click=False)`

Sculpt a stroke into the geometry

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
- **override_location** (*boolean, (optional)*) – Override Location, Override the given *location* array by recalculating object space positions from the provided *mouse_event* positions
- **ignore_background_click** (*boolean, (optional)*) – Ignore Background Click, Clicks on the background do not start the stroke

`bpy.ops.sculpt.cloth_filter(*, start_mouse=(0, 0), area_normal_radius=0.25, strength=1.0, iteration_count=1, event_history=None, type='GRAVITY', force_axis={'X', 'Y', 'Z'}, orientation='LOCAL', cloth_mass=1.0, cloth_damping=0.0, use_face_sets=False, use_collisions=False)`

Applies a cloth simulation deformation to the entire mesh

PARAMETERS:

- **start_mouse** (*int array of 2 items in [0, 16384], (optional)*) – Starting Mouse
- **area_normal_radius** (*float in [0.001, 5], (optional)*) – Normal Radius, Radius used for calculating area normal on initial click, in percentage of brush radius
- **strength** (*float in [-10, 10], (optional)*) – Strength, Filter strength
- **iteration_count** (*int in [1, 10000], (optional)*) – Repeat, How many times to repeat the filter
- **type** (*enum in ['GRAVITY', 'INFLATE', 'EXPAND', 'PINCH', 'SCALE'], (optional)*) – Filter Type, Operation that is going to be applied to the mesh
 - `GRAVITY` Gravity – Applies gravity to the simulation.
 - `INFLATE` Inflate – Inflates the cloth.
 - `EXPAND` Expand – Expands the cloth's dimensions.
 - `PINCH` Pinch – Pulls the cloth to the cursor's start position.
 - `SCALE` Scale – Scales the mesh as a soft body using the origin of the object as scale.
- **force_axis** (*enum set in {'X', 'Y', 'Z'}, (optional)*) – Force Axis, Apply the force in the selected axis
 - `X` X – Apply force in the X axis.
 - `Y` Y – Apply force in the Y axis.
 - `Z` Z – Apply force in the Z axis.
- **orientation** (*enum in ['LOCAL', 'WORLD', 'VIEW'], (optional)*) – Orientation, Orientation of the axis to limit the filter force
 - `LOCAL` Local – Use the local axis to limit the force and set the gravity direction.
 - `WORLD` World – Use the global axis to limit the force and set the gravity direction.
 - `VIEW` View – Use the view axis to limit the force and set the gravity direction.

- **cloth_mass** (*float in [0, 2], (optional)*) – Cloth Mass, Mass of each simulation particle
- **cloth_damping** (*float in [0, 1], (optional)*) – Cloth Damping, How much the applied forces are propagated through the cloth
- **use_face_sets** (*boolean, (optional)*) – Use Face Sets, Apply the filter only to the Face Set under the cursor
- **use_collisions** (*boolean, (optional)*) – Use Collisions, Collide with other collider objects in the scene

`bpy.ops.sculpt.color_filter(*, start_mouse=(0, 0), area_normal_radius=0.25, strength=1.0, iteration_count=1, event_history=None, type='FILL', fill_color=(1.0, 1.0, 1.0))`

Applies a filter to modify the active color attribute

PARAMETERS:

- **start_mouse** (*int array of 2 items in [0, 16384], (optional)*) – Starting Mouse
- **area_normal_radius** (*float in [0.001, 5], (optional)*) – Normal Radius, Radius used for calculating area normal on initial click, in percentage of brush radius
- **strength** (*float in [-10, 10], (optional)*) – Strength, Filter strength
- **iteration_count** (*int in [1, 10000], (optional)*) – Repeat, How many times to repeat the filter
- **type** (*enum in ['FILL', 'HUE', 'SATURATION', 'VALUE', 'BRIGHTNESS', 'CONTRAST', 'SMOOTH', 'RED', 'GREEN', 'BLUE'], (optional)*) –

Filter Type

- **FILL** Fill – Fill with a specific color.
- **HUE** Hue – Change hue.
- **SATURATION** Saturation – Change saturation.
- **VALUE** Value – Change value.
- **BRIGHTNESS** Brightness – Change brightness.
- **CONTRAST** Contrast – Change contrast.
- **SMOOTH** Smooth – Smooth colors.
- **RED** Red – Change red channel.
- **GREEN** Green – Change green channel.
- **BLUE** Blue – Change blue channel.
- **fill_color** (`mathutils.Color` of 3 items in [0, inf], (optional)) – Fill Color

`bpy.ops.sculpt.detail_flood_fill()`

Flood fill the mesh with the selected detail setting

`bpy.ops.sculpt.dynamic_topology_toggle()`

Dynamic topology alters the mesh topology while sculpting

`bpy.ops.sculpt.dyntopo_detail_size_edit()`

Modify the detail size of dyntopo interactively

`bpy.ops.sculpt.expand(*, target='MASK', falloff_type='GEODESIC', invert=False, use_mask_preserve=False, use_falloff_gradient=False, use_modify_active=False, use_reposition_pivot=True, max_geodesic_move_preview=10000, use_auto_mask=False, normal_falloff_smooth=2)`

Generic sculpt expand operator

PARAMETERS:

- **target** (*enum in ['MASK', 'FACE_SETS', 'COLOR'], (optional)*) – Data Target, Data that is going to be modified in the expand operation
- **falloff_type** (*enum in ['GEODESIC', 'TOPOLOGY', 'TOPOLOGY_DIAGONALS', 'NORMALS', 'SPHERICAL', 'BOUNDARY_TOPOLOGY', 'BOUNDARY_FACE_SET', 'ACTIVE_FACE_SET'], (optional)*) – Falloff Type, Initial falloff of the expand operation
- **invert** (*boolean, (optional)*) – Invert, Invert the expand active elements
- **use_mask_preserve** (*boolean, (optional)*) – Preserve Previous, Preserve the previous state of the target data
- **use_falloff_gradient** (*boolean, (optional)*) – Falloff Gradient, Expand Using a linear falloff

- **use_modify_active** (*boolean, (optional)*) – Modify Active, Modify the active Face Set instead of creating a new one
- **use_reposition_pivot** (*boolean, (optional)*) – Reposition Pivot, Reposition the sculpt transform pivot to the boundary of the expand active area
- **max_geodesic_move_preview** (*int in [0, inf], (optional)*) – Max Vertex Count for Geodesic Move Preview, Maximum number of vertice in the mesh for using geodesic falloff when moving the origin of expand. If the total number of vertices is greater than this value, the falloff will be set to spherical when moving
- **use_auto_mask** (*boolean, (optional)*) – Auto Create, Fill in mask if nothing is already masked
- **normal_falloff_smooth** (*int in [0, 10], (optional)*) – Normal Smooth, Blurring steps for normal falloff

`bpy.ops.sculpt.face_set_box_gesture(*, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True, use_front_faces_only=False)`

Add a face set in a rectangle defined by the cursor

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
- **use_front_faces_only** (*boolean, (optional)*) – Front Faces Only, Affect only faces facing towards the view

`bpy.ops.sculpt.face_set_change_visibility(*, mode='TOGGLE')`

Change the visibility of the Face Sets of the sculpt

PARAMETERS:

mode (*enum in ['TOGGLE', 'SHOW_ACTIVE', 'HIDE_ACTIVE'], (optional)*) –

Mode

- **TOGGLE** Toggle Visibility – Hide all Face Sets except for the active one.
- **SHOW_ACTIVE** Show Active Face Set – Show Active Face Set.
- **HIDE_ACTIVE** Hide Active Face Sets – Hide Active Face Sets.

`bpy.ops.sculpt.face_set_edit(*, active_face_set=1, mode='GROW', strength=1.0, modify_hidden=False)`

Edits the current active Face Set

PARAMETERS:

- **active_face_set** (*int in [0, inf], (optional)*) – Active Face Set
- **mode** (*enum in ['GROW', 'SHRINK', 'DELETE_GEOMETRY', 'FAIR_POSITIONS', 'FAIR_TANGENCY'], (optional)*) –
Mode
 - **GROW** Grow Face Set – Grows the Face Sets boundary by one face based on mesh topology.
 - **SHRINK** Shrink Face Set – Shrinks the Face Sets boundary by one face based on mesh topology.
 - **DELETE_GEOMETRY** Delete Geometry – Deletes the faces that are assigned to the Face Set.
 - **FAIR_POSITIONS** Fair Positions – Creates a smooth as possible geometry patch from the Face Set minimizing changes in vertex positions.
 - **FAIR_TANGENCY** Fair Tangency – Creates a smooth as possible geometry patch from the Face Set minimizing changes in vertex tangents.
- **strength** (*float in [0, 1], (optional)*) – Strength
- **modify_hidden** (*boolean, (optional)*) – Modify Hidden, Apply the edit operation to hidden geometry

`bpy.ops.sculpt.face_set_lasso_gesture(*, path=None, use_smooth_stroke=False, smooth_stroke_factor=0.75, smooth_stroke_radius=35, use_front_faces_only=False)`

Add a face set in a shape defined by the cursor

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
- **use_front_faces_only** (boolean, (optional)) – Front Faces Only, Affect only faces facing towards the view

bpy.ops.sculpt.face_set_line_gesture(*, xstart=0, xend=0, ystart=0, yend=0, flip=False, cursor=5, use_front_faces_only=False, use_limit_to_segment=False)

Add a face set to one side of a line defined by the cursor

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
- **use_front_faces_only** (boolean, (optional)) – Front Faces Only, Affect only faces facing towards the view
- **use_limit_to_segment** (boolean, (optional)) – Limit to Segment, Apply the gesture action only to the area that is contained within the segment without extending its effect to the entire line

bpy.ops.sculpt.face_set_polyline_gesture(*, path=None, use_front_faces_only=False)

Add a face set in a shape defined by the cursor

PARAMETERS:

- **path** (bpy_prop_collection of OperatorMousePath, (optional)) – Path
- **use_front_faces_only** (boolean, (optional)) – Front Faces Only, Affect only faces facing towards the view

bpy.ops.sculpt.face_sets_create(*, mode='MASKED')

Create a new Face Set

PARAMETERS:

mode (enum in ['MASKED', 'VISIBLE', 'ALL', 'SELECTION'], (optional)) –

Mode

- **MASKED** Face Set from Masked – Create a new Face Set from the masked faces.
- **VISIBLE** Face Set from Visible – Create a new Face Set from the visible vertices.
- **ALL** Face Set Full Mesh – Create an unique Face Set with all faces in the sculpt.
- **SELECTION** Face Set from Edit Mode Selection – Create an Face Set corresponding to the Edit Mode face selection.

bpy.ops.sculpt.face_sets_init(*, mode='LOOSE_PARTS', threshold=0.5)

Initializes all Face Sets in the mesh

PARAMETERS:

- **mode** (enum in ['LOOSE_PARTS', 'MATERIALS', 'NORMALS', 'UV_SEAMS', 'CREASES', 'BEVEL_WEIGHT', 'SHARP_EDGES', 'FACE_SET_BOUNDARIES'], (optional)) –

Mode

- **LOOSE_PARTS** Face Sets from Loose Parts – Create a Face Set per loose part in the mesh.
- **MATERIALS** Face Sets from Material Slots – Create a Face Set per Material Slot.
- **NORMALS** Face Sets from Mesh Normals – Create Face Sets for Faces that have similar normal.
- **UV_SEAMS** Face Sets from UV Seams – Create Face Sets using UV Seams as boundaries.
- **CREASES** Face Sets from Edge Creases – Create Face Sets using Edge Creases as boundaries.
- **BEVEL_WEIGHT** Face Sets from Bevel Weight – Create Face Sets using Bevel Weights as boundaries.

- **DEVEL_WEIGHT** Face Sets from Devel Weight – Create Face Sets using Devel Weights as boundaries.
- **SHARP_EDGES** Face Sets from Sharp Edges – Create Face Sets using Sharp Edges as boundaries.
- **FACE_SET_BOUNDARIES** Face Sets from Face Set Boundaries – Create a Face Set per isolated Face Set.
- **threshold** (*float in [0, 1], (optional)*) – Threshold, Minimum value to consider a certain attribute a boundary when creating the Face Sets

`bpy.ops.sculpt.face_sets_randomize_colors()`

Generates a new set of random colors to render the Face Sets in the viewport

`bpy.ops.sculpt.mask_by_color(*, contiguous=False, invert=False, preserve_previous_mask=False, threshold=0.35)`

Creates a mask based on the active color attribute

PARAMETERS:

- **contiguous** (*boolean, (optional)*) – Contiguous, Mask only contiguous color areas
- **invert** (*boolean, (optional)*) – Invert, Invert the generated mask
- **preserve_previous_mask** (*boolean, (optional)*) – Preserve Previous Mask, Preserve the previous mask and add or subtract the new one generated by the colors
- **threshold** (*float in [0, 1], (optional)*) – Threshold, How much changes in color affect the mask generation

`bpy.ops.sculpt.mask_filter(*, filter_type='SMOOTH', iterations=1, auto_iteration_count=True)`

Applies a filter to modify the current mask

PARAMETERS:

- **filter_type** (*enum in ['SMOOTH', 'SHARPEN', 'GROW', 'SHRINK', 'CONTRAST_INCREASE', 'CONTRAST_DECREASE'], (optional)*) – Type, Filter that is going to be applied to the mask
- **iterations** (*int in [1, 100], (optional)*) – Iterations, Number of times that the filter is going to be applied
- **auto_iteration_count** (*boolean, (optional)*) – Auto Iteration Count, Use an automatic number of iterations based on the number of vertices the sculpt

`bpy.ops.sculpt.mask_from_boundary(*, mix_mode='MIX', mix_factor=1.0, settings_source='OPERATOR', boundary_mode='MESH', propagation_steps=1)`

Creates a mask based on the boundaries of the surface

PARAMETERS:

- **mix_mode** (*enum in ['MIX', 'MULTIPLY', 'DIVIDE', 'ADD', 'SUBTRACT'], (optional)*) – Mode, Mix mode
- **mix_factor** (*float in [0, 5], (optional)*) – Mix Factor
- **settings_source** (*enum in ['OPERATOR', 'BRUSH', 'SCENE'], (optional)*) – Settings, Use settings from here
 - **OPERATOR** Operator – Use settings from operator properties.
 - **BRUSH** Brush – Use settings from brush.
 - **SCENE** Scene – Use settings from scene.
- **boundary_mode** (*enum in ['MESH', 'FACE_SETS'], (optional)*) – Mode, Boundary type to mask
 - **MESH** Mesh – Calculate the boundary mask based on disconnected mesh topology islands.
 - **FACE_SETS** Face Sets – Calculate the boundary mask between face sets.
- **propagation_steps** (*int in [1, 20], (optional)*) – Propagation Steps

`bpy.ops.sculpt.mask_from_cavity(*, mix_mode='MIX', mix_factor=1.0, settings_source='OPERATOR', factor=0.5, blur_steps=2, use_curve=False, invert=False)`

Creates a mask based on the curvature of the surface

PARAMETERS:

- **mix_mode** (*enum in ['MIX', 'MULTIPLY', 'DIVIDE', 'ADD', 'SUBTRACT'], (optional)*) – Mode, Mix mode

- **mix_factor** (*float in [0, 5], (optional)*) – Mix Factor
- **settings_source** (*enum in ['OPERATOR', 'BRUSH', 'SCENE'], (optional)*) – Settings, Use settings from here
 - **OPERATOR** Operator – Use settings from operator properties.
 - **BRUSH** Brush – Use settings from brush.
 - **SCENE** Scene – Use settings from scene.
- **factor** (*float in [0, 5], (optional)*) – Factor, The contrast of the cavity mask
- **blur_steps** (*int in [0, 25], (optional)*) – Blur, The number of times the cavity mask is blurred
- **use_curve** (*boolean, (optional)*) – Custom Curve
- **invert** (*boolean, (optional)*) – Cavity (Inverted)

`bpy.ops.sculpt.mask_init(*, mode='RANDOM_PER_VERTEX')`

Creates a new mask for the entire mesh

PARAMETERS:

mode (*enum in ['RANDOM_PER_VERTEX', 'RANDOM_PER_FACE_SET', 'RANDOM_PER_LOOSE_PART'], (optional)*) – Mode

`bpy.ops.sculpt.mesh_filter(*, start_mouse=(0, 0), area_normal_radius=0.25, strength=1.0, iteration_count=1, event_history=None, type='INFLATE', deform_axis={'X', 'Y', 'Z'}, orientation='LOCAL', surface_smooth_shape_preservation=0.5, surface_smooth_current_vertex=0.5, sharpen_smooth_ratio=0.35, sharpen_intensify_detail_strength=0.0, sharpen_curvature_smooth_iterations=0)`

Applies a filter to modify the current mesh

PARAMETERS:

- **start_mouse** (*int array of 2 items in [0, 16384], (optional)*) – Starting Mouse
- **area_normal_radius** (*float in [0.001, 5], (optional)*) – Normal Radius, Radius used for calculating area normal on initial click, in percentage of brush radius
- **strength** (*float in [-10, 10], (optional)*) – Strength, Filter strength
- **iteration_count** (*int in [1, 10000], (optional)*) – Repeat, How many times to repeat the filter
- **type** (*enum in ['SMOOTH', 'SCALE', 'INFLATE', 'SPHERE', 'RANDOM', 'RELAX', 'RELAX_FACE_SETS', 'SURFACE_SMOOTH', 'SHARPEN', 'ENHANCE_DETAILS', 'ERASE_DISPLACEMENT'], (optional)*) – Filter Type, Operation that is going to be applied to the mesh
 - **SMOOTH** Smooth – Smooth mesh.
 - **SCALE** Scale – Scale mesh.
 - **INFLATE** Inflate – Inflate mesh.
 - **SPHERE** Sphere – Morph into sphere.
 - **RANDOM** Random – Randomize vertex positions.
 - **RELAX** Relax – Relax mesh.
 - **RELAX_FACE_SETS** Relax Face Sets – Smooth the edges of all the Face Sets.
 - **SURFACE_SMOOTH** Surface Smooth – Smooth the surface of the mesh, preserving the volume.
 - **SHARPEN** Sharpen – Sharpen the cavities of the mesh.
 - **ENHANCE_DETAILS** Enhance Details – Enhance the high frequency surface detail.
 - **ERASE_DISPLACEMENT** Erase Displacement – Deletes the displacement of the Multires Modifier.
- **deform_axis** (*enum set in {'X', 'Y', 'Z'}, (optional)*) – Deform Axis, Apply the deformation in the selected axis
 - **X** X – Deform in the X axis.
 - **Y** Y – Deform in the Y axis.
 - **Z** Z – Deform in the Z axis.
- **orientation** (*enum in ['LOCAL', 'WORLD', 'VIEW'], (optional)*) – Orientation Orientation of the axis to limit the filter displacement

SHARPEN, SMOOTH AND DISPLACEMENT

- **LOCAL** Local – Use the local axis to limit the displacement.
- **WORLD** World – Use the global axis to limit the displacement.
- **VIEW** View – Use the view axis to limit the displacement.
- **surface_smooth_shape_preservation** (*float in [0, 1], (optional)*) – Shape Preservation, How much of the original shape is preserved when smoothing
- **surface_smooth_current_vertex** (*float in [0, 1], (optional)*) – Per Vertex Displacement, How much the position of each individual vertex influences the final result
- **sharpen_smooth_ratio** (*float in [0, 1], (optional)*) – Smooth Ratio, How much smoothing is applied to polished surfaces
- **sharpen_intensify_detail_strength** (*float in [0, 10], (optional)*) – Intensify Details, How much creases and valleys are intensified
- **sharpen_curvature_smooth_iterations** (*int in [0, 10], (optional)*) – Curvature Smooth Iterations, How much smooth the resulting shape : ignoring high frequency details

bpy.ops.sculpt.optimize()

Recalculate the sculpt BVH to improve performance

bpy.ops.sculpt.project_line_gesture(*, xstart=0, xend=0, ystart=0, yend=0, flip=False, cursor=5, use_front_faces_only=False, use_limit_to_segment=False)

Project the geometry onto a plane defined by a line

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
- **use_front_faces_only** (*boolean, (optional)*) – Front Faces Only, Affect only faces facing towards the view
- **use_limit_to_segment** (*boolean, (optional)*) – Limit to Segment, Apply the gesture action only to the area that is contained within the segment without extending its effect to the entire line

bpy.ops.sculpt.sample_color()

Sample the vertex color of the active vertex

bpy.ops.sculpt.sample_detail_size(*, location=(0, 0), mode='DYNTPOPO')

Sample the mesh detail on clicked point

PARAMETERS:

- **location** (*int array of 2 items in [0, 32767], (optional)*) – Location, Screen coordinates of sampling
- **mode** (*enum in ['DYNTPOPO', 'VOXEL'], (optional)*) – Detail Mode, Target sculpting workflow that is going to use the sampled size
 - **DYNTPOPO** Dyntopo – Sample dyntopo detail.
 - **VOXEL** Voxel – Sample mesh voxel size.

bpy.ops.sculpt.sculptmode_toggle()

Toggle sculpt mode in 3D view

bpy.ops.sculpt.set_persistent_base()

Reset the copy of the mesh that is being sculpted on

bpy.ops.sculpt.set_pivot_position(*, mode='UNMASKED', mouse_x=0.0, mouse_y=0.0)

Sets the sculpt transform pivot position

PARAMETERS:

- **mode** (enum in ['ORIGIN', 'UNMASKED', 'BORDER', 'ACTIVE', 'SURFACE'], (optional)) – Mode
 - ORIGIN Origin – Sets the pivot to the origin of the sculpt.
 - UNMASKED Unmasked – Sets the pivot position to the average position of the unmasked vertices.
 - BORDER Mask Border – Sets the pivot position to the center of the border of the mask.
 - ACTIVE Active Vertex – Sets the pivot position to the active vertex position.
 - SURFACE Surface – Sets the pivot position to the surface under the cursor.
- **mouse_x** (float in [0, inf], (optional)) – Mouse Position X, Position of the mouse used for “Surface” mode
- **mouse_y** (float in [0, inf], (optional)) – Mouse Position Y, Position of the mouse used for “Surface” mode

bpy.ops.sculpt.symmetrize(*, merge_tolerance=0.0005)

Symmetrize the topology modifications

PARAMETERS:

merge_tolerance (float in [0, inf], (optional)) – Merge Distance, Distance within which symmetrical vertices are merged

bpy.ops.sculpt.trim_box_gesture(*, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True, use_front_faces_only=False, location=(0, 0), trim_mode='DIFFERENCE', use_cursor_depth=False, trim_orientation='VIEW', trim_extrude_mode='FIXED', trim_solver='FAST')

Execute a boolean operation on the mesh and a rectangle defined by the cursor

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
- **use_front_faces_only** (boolean, (optional)) – Front Faces Only, Affect only faces facing towards the view
- **location** (int array of 2 items in [-inf, inf], (optional)) – Location, Mouse location
- **trim_mode** (enum in ['DIFFERENCE', 'UNION', 'JOIN'], (optional)) – Trim Mode
 - DIFFERENCE Difference – Use a difference boolean operation.
 - UNION Union – Use a union boolean operation.
 - JOIN Join – Join the new mesh as separate geometry, without performing any boolean operation.
- **use_cursor_depth** (boolean, (optional)) – Use Cursor for Depth, Use cursor location and radius for the dimensions and position of the trimming shape
- **trim_orientation** (enum in ['VIEW', 'SURFACE'], (optional)) – Shape Orientation
 - VIEW View – Use the view to orientate the trimming shape.
 - SURFACE Surface – Use the surface normal to orientate the trimming shape.
- **trim_extrude_mode** (enum in ['PROJECT', 'FIXED'], (optional)) – Extrude Mode
 - PROJECT Project – Align trim geometry with the perspective of the current view for a tapered shape.
 - FIXED Fixed – Align trim geometry orthogonally for a shape with 90 degree angles.
- **trim_solver** (enum in ['EXACT', 'FAST'], (optional)) – Solver
 - EXACT Exact – Use the exact boolean solver.

- **FAST** Fast – Use the fast float boolean solver.

```
bpy.ops.sculpt.trim_lasso_gesture(*, path=None, use_smooth_stroke=False, smooth_stroke_factor=0.75, smooth_stroke_radius=35,
    use_front_faces_only=False, location=(0, 0), trim_mode='DIFFERENCE', use_cursor_depth=False, trim_orientation='VIEW',
    trim_extrude_mode='FIXED', trim_solver='FAST')
```

Execute a boolean operation on the mesh and a shape defined by the cursor

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
- **use_front_faces_only** (*boolean*, (optional)) – Front Faces Only, Affect only faces facing towards the view
- **location** (*int* array of 2 items in *[-inf, inf]*, (optional)) – Location, Mouse location
- **trim_mode** (*enum* in *['DIFFERENCE', 'UNION', 'JOIN']*, (optional)) – Trim Mode
 - **DIFFERENCE** Difference – Use a difference boolean operation.
 - **UNION** Union – Use a union boolean operation.
 - **JOIN** Join – Join the new mesh as separate geometry, without performing any boolean operation.
- **use_cursor_depth** (*boolean*, (optional)) – Use Cursor for Depth, Use cursor location and radius for the dimensions and position of the trimming shape
- **trim_orientation** (*enum* in *['VIEW', 'SURFACE']*, (optional)) – Shape Orientation
 - **VIEW** View – Use the view to orientate the trimming shape.
 - **SURFACE** Surface – Use the surface normal to orientate the trimming shape.
- **trim_extrude_mode** (*enum* in *['PROJECT', 'FIXED']*, (optional)) – Extrude Mode
 - **PROJECT** Project – Align trim geometry with the perspective of the current view for a tapered shape.
 - **FIXED** Fixed – Align trim geometry orthogonally for a shape with 90 degree angles.
- **trim_solver** (*enum* in *['EXACT', 'FAST']*, (optional)) – Solver
 - **EXACT** Exact – Use the exact boolean solver.
 - **FAST** Fast – Use the fast float boolean solver.

```
bpy.ops.sculpt.trim_line_gesture(*, xstart=0, xend=0, ystart=0, yend=0, flip=False, cursor=5, use_front_faces_only=False,
    use_limit_to_segment=False, location=(0, 0), trim_mode='DIFFERENCE', use_cursor_depth=False, trim_orientation='VIEW',
    trim_extrude_mode='FIXED', trim_solver='FAST')
```

Remove a portion of the mesh on one side of a line

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
- **use_front_faces_only** (*boolean*, (optional)) – Front Faces Only, Affect only faces facing towards the view
- **use_limit_to_segment** (*boolean*, (optional)) – Limit to Segment, Apply the gesture action only to the area that is contained within the segment without extending its effect to the entire line

- **location** (*int array of 2 items in [-inf, inf], (optional)*) – Location, Mouse location
- **trim_mode** (*enum in ['DIFFERENCE', 'UNION', 'JOIN'], (optional)*) – Trim Mode
 - **DIFFERENCE** Difference – Use a difference boolean operation.
 - **UNION** Union – Use a union boolean operation.
 - **JOIN** Join – Join the new mesh as separate geometry, without performing any boolean operation.
- **use_cursor_depth** (*boolean, (optional)*) – Use Cursor for Depth, Use cursor location and radius for the dimensions and position of the trimming shape
- **trim_orientation** (*enum in ['VIEW', 'SURFACE'], (optional)*) – Shape Orientation
 - **VIEW** View – Use the view to orientate the trimming shape.
 - **SURFACE** Surface – Use the surface normal to orientate the trimming shape.
- **trim_extrude_mode** (*enum in ['PROJECT', 'FIXED'], (optional)*) – Extrude Mode
 - **PROJECT** Project – Align trim geometry with the perspective of the current view for a tapered shape.
 - **FIXED** Fixed – Align trim geometry orthogonally for a shape with 90 degree angles.
- **trim_solver** (*enum in ['EXACT', 'FAST'], (optional)*) – Solver
 - **EXACT** Exact – Use the exact boolean solver.
 - **FAST** Fast – Use the fast float boolean solver.

`bpy.ops.sculpt.trim_polyline_gesture(*, path=None, use_front_faces_only=False, location=(0, 0), trim_mode='DIFFERENCE', use_cursor_depth=False, trim_orientation='VIEW', trim_extrude_mode='FIXED', trim_solver='FAST')`

Execute a boolean operation on the mesh and a polygonal shape defined by the cursor

PARAMETERS:

- **path** (*bpy_prop_collection of OperatorMousePath, (optional)*) – Path
- **use_front_faces_only** (*boolean, (optional)*) – Front Faces Only, Affect only faces facing towards the view
- **location** (*int array of 2 items in [-inf, inf], (optional)*) – Location, Mouse location
- **trim_mode** (*enum in ['DIFFERENCE', 'UNION', 'JOIN'], (optional)*) – Trim Mode
 - **DIFFERENCE** Difference – Use a difference boolean operation.
 - **UNION** Union – Use a union boolean operation.
 - **JOIN** Join – Join the new mesh as separate geometry, without performing any boolean operation.
- **use_cursor_depth** (*boolean, (optional)*) – Use Cursor for Depth, Use cursor location and radius for the dimensions and position of the trimming shape
- **trim_orientation** (*enum in ['VIEW', 'SURFACE'], (optional)*) – Shape Orientation
 - **VIEW** View – Use the view to orientate the trimming shape.
 - **SURFACE** Surface – Use the surface normal to orientate the trimming shape.
- **trim_extrude_mode** (*enum in ['PROJECT', 'FIXED'], (optional)*) – Extrude Mode
 - **PROJECT** Project – Align trim geometry with the perspective of the current view for a tapered shape.
 - **FIXED** Fixed – Align trim geometry orthogonally for a shape with 90 degree angles.
- **trim_solver** (*enum in ['EXACT', 'FAST'], (optional)*) – Solver

- `EXACT` Exact – Use the exact boolean solver.
- `FAST` Fast – Use the fast float boolean solver.

`bpy.ops.sculpt.uv_sculpt_grab(*, use_invert=False)`

Grab UVs

PARAMETERS:

`use_invert` (*boolean, (optional)*) – Invert, Invert action for the duration of the stroke

`bpy.ops.sculpt.uv_sculpt_pinch(*, use_invert=False)`

Pinch UVs

PARAMETERS:

`use_invert` (*boolean, (optional)*) – Invert, Invert action for the duration of the stroke

`bpy.ops.sculpt.uv_sculpt_relax(*, use_invert=False, relax_method='COTAN')`

Relax UVs

PARAMETERS:

- **`use_invert`** (*boolean, (optional)*) – Invert, Invert action for the duration of the stroke
- **`relax_method`** (*enum in ['LAPLACIAN', 'HC', 'COTAN'], (optional)*) – Relax Method, Algorithm used for UV relaxation
 - `LAPLACIAN` Laplacian – Use Laplacian method for relaxation.
 - `HC` HC – Use HC method for relaxation.
 - `COTAN` Geometry – Use Geometry (cotangent) relaxation, making UVs follow the underlying 3D geometry.