Skip to content

Transform Operators

bpy.ops.transform.**bbone_resize(*, value=(1.0, 1.0, 1.0), orient_type='GLOBAL', orient_matrix=((0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0)), orient_matrix_type='GLOBAL', constraint_axis=(False, False, False), mirror=False, release_confirm=False, use_accurate=False)**

Scale selected bendy bones display size

PARAMETERS:

- **value** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Display Size
- **orient_type** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **orient_matrix** (`mathutils.Matrix` of 3 * 3 items in [-inf, inf], (optional)) – Matrix
- **orient_matrix_type** (*enum in [], (optional)*) – Matrix Orientation
- **constraint_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**bend(*, value=0.0, mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=False, snap=False, gpencil_strokes=False, center_override=(0.0, 0.0, 0.0), release_confirm=False, use_accurate=False)**

Bend selected items between the 3D cursor and the mouse

PARAMETERS:

- **value** (*float array of 1 items in [-inf, inf], (optional)*) – Angle
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **gpencil_strokes** (*boolean, (optional)*) – Edit Grease Pencil, Edit selected Grease Pencil strokes
- **center_override** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Center Override, Force using this center value (when set)
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**create_orientation(*, name='', use_view=False, use=False, overwrite=False)**

Create transformation orientation from selection

PARAMETERS:

- **name** (*string, (optional, never None)*) – Name, Name of the new custom orientation
- **use_view** (*boolean, (optional)*) – Use View, Use the current view instead of the active object to create the new orientation
- **use** (*boolean, (optional)*) – Use After Creation, Select orientation after its creation
- **overwrite** (*boolean, (optional)*) – Overwrite Previous, Overwrite previously created orientation with same name

bpy.ops.transform.**delete_orientation()**

Delete transformation orientation

bpy.ops.transform.**edge_bevelweight(*, value=0.0, snap=False, release_confirm=False, use_accurate=False)**

Change the bevel weight of edges

PARAMETERS:

- **value** (*float in [-1, 1], (optional)*) – Factor
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**edge_crease(*, value=0.0, snap=False, release_confirm=False, use_accurate=False)**

Change the crease of edges

PARAMETERS:

- **value** (*float in [-1, 1], (optional)*) – Factor
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**edge_slide(*, value=0.0, single_side=False, use_even=False, flipped=False, use_clamp=True, mirror=False, snap=False, snap_elements={'INCREMENT'}, use_snap_project=False, snap_target='CLOSEST', use_snap_self=True, use_snap_edit=True, use_snap_nonedit=True, use_snap_selectable=False, snap_point=(0.0, 0.0, 0.0), correct_uv=True, release_confirm=False, use_accurate=False)**

Slide an edge loop along a mesh

PARAMETERS:

- **value** (*float in [-10, 10], (optional)*) – Factor
- **single_side** (*boolean, (optional)*) – Single Side
- **use_even** (*boolean, (optional)*) – Even, Make the edge loop match the shape of the adjacent edge loop
- **flipped** (*boolean, (optional)*) – Flipped, When Even mode is active, flips between the two adjacent edge loops
- **use_clamp** (*boolean, (optional)*) – Clamp, Clamp within the edge extents
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap_elements** (enum set in Snap Element Items, (optional)) – Snap to Elements
- **use_snap_project** (*boolean, (optional)*) – Project Individual Elements
- **snap_target** (enum in Snap Source Items, (optional)) – Snap Base, Point on source that will snap to target
- **use_snap_self** (*boolean, (optional)*) – Target: Include Active
- **use_snap_edit** (*boolean, (optional)*) – Target: Include Edit
- **use_snap_nonedit** (*boolean, (optional)*) – Target: Include Non-Edited
- **use_snap_selectable** (*boolean, (optional)*) – Target: Exclude Non-Selectable
- **snap_point** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Point
- **correct_uv** (*boolean, (optional)*) – Correct UVs, Correct UV coordinates when transforming
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**from_gizmo()**

Transform selected items by mode type

bpy.ops.transform.**mirror(*, orient_type='GLOBAL', orient_matrix=((0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0)), orient_matrix_type='GLOBAL', constraint_axis=(False, False, False), gpencil_strokes=False, center_override=(0.0, 0.0, 0.0), release_confirm=False, use_accurate=False)**

Mirror selected items around one or more axes

PARAMETERS:

- **orient_type** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **orient_matrix** (`mathutils.Matrix` of 3 * 3 items in [-inf, inf], (optional)) – Matrix
- **orient_matrix_type** (*enum in [], (optional)*) – Matrix Orientation
- **constraint_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis

- **constraint_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **gpencil_strokes** (*boolean, (optional)*) – Edit Grease Pencil, Edit selected Grease Pencil strokes
- **center_override** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Center Override, Force using this center value (when set)
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**push_pull(*, value=0.0, mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=False, snap=False, center_override=(0.0, 0.0, 0.0), release_confirm=False, use_accurate=False)**

Push/Pull selected items

PARAMETERS:

- **value** (*float in [-inf, inf], (optional)*) – Distance
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **center_override** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Center Override, Force using this center value (when set)
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**resize(*, value=(1.0, 1.0, 1.0), mouse_dir_constraint=(0.0, 0.0, 0.0), orient_type='GLOBAL', orient_matrix=((0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0)), orient_matrix_type='GLOBAL', constraint_axis=(False, False, False), mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=False, snap=False, snap_elements={'INCREMENT'}, use_snap_project=False, snap_target='CLOSEST', use_snap_self=True, use_snap_edit=True, use_snap_nonedit=True, use_snap_selectable=False, snap_point=(0.0, 0.0, 0.0), gpencil_strokes=False, texture_space=False, remove_on_cancel=False, use_duplicated_keyframes=False, center_override=(0.0, 0.0, 0.0), release_confirm=False, use_accurate=False)**

Scale (resize) selected items

PARAMETERS:

- **value** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Scale
- **mouse_dir_constraint** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Mouse Directional Constraint
- **orient_type** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **orient_matrix** (`mathutils.Matrix` of 3 * 3 items in [-inf, inf], (optional)) – Matrix
- **orient_matrix_type** (*enum in [], (optional)*) – Matrix Orientation
- **constraint_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap_elements** (enum set in Snap Element Items, (optional)) – Snap to Elements
- **use_snap_project** (*boolean, (optional)*) – Project Individual Elements
- **snap_target** (enum in Snap Source Items, (optional)) – Snap Base, Point on source that will snap to target
- **use_snap_self** (*boolean, (optional)*) – Target: Include Active
- **use_snap_edit** (*boolean, (optional)*) – Target: Include Edit

- **use_snap_nonedit** (*boolean, (optional)*) – Target: Include Non-Edited
- **use_snap_selectable** (*boolean, (optional)*) – Target: Exclude Non-Selectable
- **snap_point** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Point
- **gpencil_strokes** (*boolean, (optional)*) – Edit Grease Pencil, Edit selected Grease Pencil strokes
- **texture_space** (*boolean, (optional)*) – Edit Texture Space, Edit object data texture space
- **remove_on_cancel** (*boolean, (optional)*) – Remove on Cancel, Remove elements on cancel
- **use_duplicated_keyframes** (*boolean, (optional)*) – Duplicated Keyframes, Transform duplicated keyframes
- **center_override** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Center Override, Force using this center value (when set)
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**rotate(*, value=0.0, orient_axis='Z', orient_type='GLOBAL', orient_matrix=((0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0)), orient_matrix_type='GLOBAL', constraint_axis=(False, False, False), mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=Fals snap=False, snap_elements={'INCREMENT'}, use_snap_project=False, snap_target='CLOSEST', use_snap_self=True, use_snap_edit=True, use_snap_nonedit=True, use_snap_selectable=False, snap_point=(0.0, 0.0, 0.0), gpencil_strokes=False, center_override=(0.0, 0.0, 0.0), release_confirm=False, use_accurate=False)**

Rotate selected items

PARAMETERS:

- **value** (*float in [-inf, inf], (optional)*) – Angle
- **orient_axis** (enum in Axis Xyz Items, (optional)) – Axis
- **orient_type** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **orient_matrix** (`mathutils.Matrix` of 3 * 3 items in [-inf, inf], (optional)) – Matrix
- **orient_matrix_type** (*enum in [], (optional)*) – Matrix Orientation
- **constraint_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap_elements** (enum set in Snap Element Items, (optional)) – Snap to Elements
- **use_snap_project** (*boolean, (optional)*) – Project Individual Elements
- **snap_target** (enum in Snap Source Items, (optional)) – Snap Base, Point on source that will snap to target
- **use_snap_self** (*boolean, (optional)*) – Target: Include Active
- **use_snap_edit** (*boolean, (optional)*) – Target: Include Edit
- **use_snap_nonedit** (*boolean, (optional)*) – Target: Include Non-Edited
- **use_snap_selectable** (*boolean, (optional)*) – Target: Exclude Non-Selectable
- **snap_point** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Point
- **gpencil_strokes** (*boolean, (optional)*) – Edit Grease Pencil, Edit selected Grease Pencil strokes
- **center_override** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Center Override, Force using this center value (when set)
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**rotate_normal(*, value=0.0, orient_axis='Z', orient_type='GLOBAL', orient_matrix=((0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0)), orient_matrix_type='GLOBAL', constraint_axis=(False, False, False), mirror=False, release_confirm=False, use_accurate=False)**

Rotate split normal of selected items

PARAMETERS:

- **value** (*float in [-inf, inf], (optional)*) – Angle
- **orient_axis** (enum in Axis Xyz Items, (optional)) – Axis
- **orient_type** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **orient_matrix** (`mathutils.Matrix` of 3 * 3 items in [-inf, inf], (optional)) – Matrix
- **orient_matrix_type** (*enum in [], (optional)*) – Matrix Orientation
- **constraint_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**select_orientation(*, orientation='GLOBAL')**

Select transformation orientation

PARAMETERS:

orientation (*enum in [], (optional)*) – Orientation, Transformation orientation

bpy.ops.transform.**seq_slide(*, value=(0.0, 0.0), use_restore_handle_selection=False, snap=False, view2d_edge_pan=False, release_confirm=False, use_accurate=False)**

Slide a sequence strip in time

PARAMETERS:

- **value** (`mathutils.Vector` of 2 items in [-inf, inf], (optional)) – Offset
- **use_restore_handle_selection** (*boolean, (optional)*) – Restore Handle Selection, Restore handle selection after tweaking
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **view2d_edge_pan** (*boolean, (optional)*) – Edge Pan, Enable edge panning in 2D view
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**shear(*, value=0.0, orient_axis='Z', orient_axis_ortho='X', orient_type='GLOBAL', orient_matrix=((0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0)), orient_matrix_type='GLOBAL', mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=Fals snap=False, gpencil_strokes=False, release_confirm=False, use_accurate=False)**

Shear selected items along the given axis

PARAMETERS:

- **value** (*float in [-inf, inf], (optional)*) – Offset
- **orient_axis** (enum in Axis Xyz Items, (optional)) – Axis
- **orient_axis_ortho** (enum in Axis Xyz Items, (optional)) – Axis Ortho
- **orient_type** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **orient_matrix** (`mathutils.Matrix` of 3 * 3 items in [-inf, inf], (optional)) – Matrix
- **orient_matrix_type** (*enum in [], (optional)*) – Matrix Orientation
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **gpencil_strokes** (*boolean, (optional)*) – Edit Grease Pencil, Edit selected Grease Pencil strokes
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**shrink_fatten(*, value=0.0, use_even_offset=False, mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=Fals snap=False, release_confirm=False, use_accurate=False)**

Shrink/fatten selected vertices along normals

PARAMETERS:

- **value** (*float in [-inf, inf], (optional)*) – Offset
- **use_even_offset** (*boolean, (optional)*) – Offset Even, Scale the offset to give more even thickness
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**skin_resize(*, value=(1.0, 1.0, 1.0), orient_type='GLOBAL', orient_matrix=((0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0)), orient_matrix_type='GLOBAL', constraint_axis=(False, False, False), mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=Fals snap=False, snap_elements={'INCREMENT'}, use_snap_project=False, snap_target='CLOSEST', use_snap_self=True, use_snap_edit=True, use_snap_nonedit=True, use_snap_selectable=False, snap_point=(0.0, 0.0, 0.0), release_confirm=False, use_accurate=False)**

Scale selected vertices' skin radii

PARAMETERS:

- **value** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Scale
- **orient_type** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **orient_matrix** (`mathutils.Matrix` of 3 * 3 items in [-inf, inf], (optional)) – Matrix
- **orient_matrix_type** (*enum in [], (optional)*) – Matrix Orientation
- **constraint_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap_elements** (enum set in Snap Element Items, (optional)) – Snap to Elements
- **use_snap_project** (*boolean, (optional)*) – Project Individual Elements
- **snap_target** (enum in Snap Source Items, (optional)) – Snap Base, Point on source that will snap to target
- **use_snap_self** (*boolean, (optional)*) – Target: Include Active
- **use_snap_edit** (*boolean, (optional)*) – Target: Include Edit
- **use_snap_nonedit** (*boolean, (optional)*) – Target: Include Non-Edited
- **use_snap_selectable** (*boolean, (optional)*) – Target: Exclude Non-Selectable
- **snap_point** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Point
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**tilt(*, value=0.0, mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH',**

proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=False, snap=False, release_confirm=False, use_accurate=False)

Tilt selected control vertices of 3D curve

PARAMETERS:

- **value** (*float in [-inf, inf], (optional)*) – Angle
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**tosphere(*, value=0.0, mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=False, snap=False, gpencil_strokes=False, center_override=(0.0, 0.0, 0.0), release_confirm=False, use_accurate=False)**

Move selected items outward in a spherical shape around geometric center

PARAMETERS:

- **value** (*float in [0, 1], (optional)*) – Factor
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **gpencil_strokes** (*boolean, (optional)*) – Edit Grease Pencil, Edit selected Grease Pencil strokes
- **center_override** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Center Override, Force using this center value (when set)
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**trackball(*, value=(0.0, 0.0), mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=False, snap=False, gpencil_strokes=False, center_override=(0.0, 0.0, 0.0), release_confirm=False, use_accurate=False)**

Trackball style rotation of selected items

PARAMETERS:

- **value** (*float array of 2 items in [-inf, inf], (optional)*) – Angle
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **gpencil_strokes** (*boolean, (optional)*) – Edit Grease Pencil, Edit selected Grease Pencil strokes
- **center_override** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Center Override, Force using this center value (when set)

- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**transform(*, mode='TRANSLATION', value=(0.0, 0.0, 0.0, 0.0), orient_axis='Z', orient_type='GLOBAL', orient_matrix=((0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0)), orient_matrix_type='GLOBAL', constraint_axis=(False, False, False), mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=False, snap=False, snap_elements={'INCREMENT'}, use_snap_project=False, snap_target='CLOSEST', use_snap_self=True, use_snap_edit=True, use_snap_nonedit=True, use_snap_selectable=False, snap_point=(0.0, 0.0, 0.0), snap_align=False, snap_normal=(0.0, 0.0, 0.0), gpencil_strokes=False, texture_space=False, remove_on_cancel=False, use_duplicated_keyframes=False, center_override=(0.0, 0.0, 0.0), release_confirm=False, use_accurate=False, use_automerge_and_split=False)**

Transform selected items by mode type

PARAMETERS:

- **mode** (enum in Transform Mode Type Items, (optional)) – Mode
- **value** (`mathutils.Vector` of 4 items in [-inf, inf], (optional)) – Values
- **orient_axis** (enum in Axis Xyz Items, (optional)) – Axis
- **orient_type** (enum in Transform Orientation Items, (optional)) – Orientation, Transformation orientation
- **orient_matrix** (`mathutils.Matrix` of 3 * 3 items in [-inf, inf], (optional)) – Matrix
- **orient_matrix_type** (enum in Transform Orientation Items, (optional)) – Matrix Orientation
- **constraint_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap_elements** (enum set in Snap Element Items, (optional)) – Snap to Elements
- **use_snap_project** (*boolean, (optional)*) – Project Individual Elements
- **snap_target** (enum in Snap Source Items, (optional)) – Snap Base, Point on source that will snap to target
- **use_snap_self** (*boolean, (optional)*) – Target: Include Active
- **use_snap_edit** (*boolean, (optional)*) – Target: Include Edit
- **use_snap_nonedit** (*boolean, (optional)*) – Target: Include Non-Edited
- **use_snap_selectable** (*boolean, (optional)*) – Target: Exclude Non-Selectable
- **snap_point** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Point
- **snap_align** (*boolean, (optional)*) – Align with Point Normal
- **snap_normal** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Normal
- **gpencil_strokes** (*boolean, (optional)*) – Edit Grease Pencil, Edit selected Grease Pencil strokes
- **texture_space** (*boolean, (optional)*) – Edit Texture Space, Edit object data texture space
- **remove_on_cancel** (*boolean, (optional)*) – Remove on Cancel, Remove elements on cancel
- **use_duplicated_keyframes** (*boolean, (optional)*) – Duplicated Keyframes, Transform duplicated keyframes
- **center_override** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Center Override, Force using this center value (when set)
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation
- **use_automerge_and_split** (*boolean, (optional)*) – Auto Merge & Split, Forces the use of Auto Merge and Split

bpy.ops.transform.**translate(*, value=(0.0, 0.0, 0.0), orient_type='GLOBAL', orient_matrix=((0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0)), orient_matrix_type='GLOBAL', constraint_axis=(False, False, False), mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1.0, use_proportional_connected=False, use_proportional_projected=Fals snap=False, snap_elements={'INCREMENT'}, use_snap_project=False, snap_target='CLOSEST', use_snap_self=True, use_snap_edit=True, use_snap_nonedit=True, use_snap_selectable=False, snap_point=(0.0, 0.0, 0.0), snap_align=False,**

snap_normal=(0.0, 0.0, 0.0), gpencil_strokes=False, cursor_transform=False, texture_space=False, remove_on_cancel=False, use_duplicated_keyframes=False, view2d_edge_pan=False, release_confirm=False, use_accurate=False, use_automerge_and_split=False)

Move selected items

PARAMETERS:

- **value** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Move
- **orient_type** (enum in Transform Orientation Items, (optional)) – Orientation, Transformation orientation
- **orient_matrix** (`mathutils.Matrix` of 3 * 3 items in [-inf, inf], (optional)) – Matrix
- **orient_matrix_type** (enum in Transform Orientation Items, (optional)) – Matrix Orientation
- **constraint_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **use_proportional_edit** (*boolean, (optional)*) – Proportional Editing
- **proportional_edit_falloff** (enum in Proportional Falloff Items, (optional)) – Proportional Falloff, Falloff type for proportional editing mode
- **proportional_size** (*float in [1e-06, inf], (optional)*) – Proportional Size
- **use_proportional_connected** (*boolean, (optional)*) – Connected
- **use_proportional_projected** (*boolean, (optional)*) – Projected (2D)
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap_elements** (enum set in Snap Element Items, (optional)) – Snap to Elements
- **use_snap_project** (*boolean, (optional)*) – Project Individual Elements
- **snap_target** (enum in Snap Source Items, (optional)) – Snap Base, Point on source that will snap to target
- **use_snap_self** (*boolean, (optional)*) – Target: Include Active
- **use_snap_edit** (*boolean, (optional)*) – Target: Include Edit
- **use_snap_nonedit** (*boolean, (optional)*) – Target: Include Non-Edited
- **use_snap_selectable** (*boolean, (optional)*) – Target: Exclude Non-Selectable
- **snap_point** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Point
- **snap_align** (*boolean, (optional)*) – Align with Point Normal
- **snap_normal** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Normal
- **gpencil_strokes** (*boolean, (optional)*) – Edit Grease Pencil, Edit selected Grease Pencil strokes
- **cursor_transform** (*boolean, (optional)*) – Transform Cursor
- **texture_space** (*boolean, (optional)*) – Edit Texture Space, Edit object data texture space
- **remove_on_cancel** (*boolean, (optional)*) – Remove on Cancel, Remove elements on cancel
- **use_duplicated_keyframes** (*boolean, (optional)*) – Duplicated Keyframes, Transform duplicated keyframes
- **view2d_edge_pan** (*boolean, (optional)*) – Edge Pan, Enable edge panning in 2D view
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation
- **use_automerge_and_split** (*boolean, (optional)*) – Auto Merge & Split, Forces the use of Auto Merge and Split

bpy.ops.transform.**vert_crease(*, value=0.0, snap=False, release_confirm=False, use_accurate=False)**

Change the crease of vertices

PARAMETERS:

- **value** (*float in [-1, 1], (optional)*) – Factor
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**vert_slide(*, value=0.0, use_even=False, flipped=False, use_clamp=True, mirror=False, snap=False, snap_elements={'INCREMENT'}, use_snap_project=False, snap_target='CLOSEST', use_snap_self=True, use_snap_edit=True, use_snap_nonedit=True, use_snap_selectable=False, snap_point=(0.0, 0.0, 0.0), correct_uv=True, release_confirm=False, use_accurate=False)**

Slide a vertex along a mesh

PARAMETERS:

- **value** (*float in [-10, 10], (optional)*) – Factor
- **use_even** (*boolean, (optional)*) – Even, Make the edge loop match the shape of the adjacent edge loop
- **flipped** (*boolean, (optional)*) – Flipped, When Even mode is active, flips between the two adjacent edge loops
- **use_clamp** (*boolean, (optional)*) – Clamp, Clamp within the edge extents
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap_elements** (enum set in Snap Element Items, (optional)) – Snap to Elements
- **use_snap_project** (*boolean, (optional)*) – Project Individual Elements
- **snap_target** (enum in Snap Source Items, (optional)) – Snap Base, Point on source that will snap to target
- **use_snap_self** (*boolean, (optional)*) – Target: Include Active
- **use_snap_edit** (*boolean, (optional)*) – Target: Include Edit
- **use_snap_nonedit** (*boolean, (optional)*) – Target: Include Non-Edited
- **use_snap_selectable** (*boolean, (optional)*) – Target: Exclude Non-Selectable
- **snap_point** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Point
- **correct_uv** (*boolean, (optional)*) – Correct UVs, Correct UV coordinates when transforming
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button
- **use_accurate** (*boolean, (optional)*) – Accurate, Use accurate transformation

bpy.ops.transform.**vertex_random(*, offset=0.0, uniform=0.0, normal=0.0, seed=0, wait_for_input=True)**

Randomize vertices

PARAMETERS:

- **offset** (*float in [-inf, inf], (optional)*) – Amount, Distance to offset
- **uniform** (*float in [0, 1], (optional)*) – Uniform, Increase for uniform offset distance
- **normal** (*float in [0, 1], (optional)*) – Normal, Align offset direction to normals
- **seed** (*int in [0, 10000], (optional)*) – Random Seed, Seed for the random number generator
- **wait_for_input** (*boolean, (optional)*) – Wait for Input

bpy.ops.transform.**vertex_warp(*, warp_angle=6.28319, offset_angle=0.0, min=-1.0, max=1.0, viewmat=((0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0)), center=(0.0, 0.0, 0.0))**

Warp vertices around the cursor

PARAMETERS:

- **warp_angle** (*float in [-inf, inf], (optional)*) – Warp Angle, Amount to warp about the cursor
- **offset_angle** (*float in [-inf, inf], (optional)*) – Offset Angle, Angle to use as the basis for warping
- **min** (*float in [-inf, inf], (optional)*) – Min
- **max** (*float in [-inf, inf], (optional)*) – Max
- **viewmat** (`mathutils.Matrix` of 4 * 4 items in [-inf, inf], (optional)) – Matrix
- **center** (`mathutils.Vector` of 3 items in [-inf, inf], (optional)) – Center