Skip to content

View3D Operators

bpy.ops.view3d.**bone_select_menu(*, name='', extend=False, deselect=False, toggle=False)**

Menu bone selection

PARAMETERS:

- **name** (*enum in [], (optional)*) – Bone Name
- **extend** (*boolean, (optional)*) – Extend
- **deselect** (*boolean, (optional)*) – Deselect
- **toggle** (*boolean, (optional)*) – Toggle

bpy.ops.view3d.**camera_background_image_add(*, filepath='', relative_path=True, name='', session_uid=0)**

Add a new background image to the active camera

PARAMETERS:

- **filepath** (*string, (optional, never None)*) – Filepath, Path to image file
- **relative_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file
- **name** (*string, (optional, never None)*) – Name, Name of the data-block to use by the operator
- **session_uid** (*int in [-inf, inf], (optional)*) – Session UID, Session UID of the data-block to use by the operator

bpy.ops.view3d.**camera_background_image_remove(*, index=0)**

Remove a background image from the camera

PARAMETERS:

index (*int in [0, inf], (optional)*) – Index, Background image index to remove

bpy.ops.view3d.**camera_to_view()**

Set camera view to active view

bpy.ops.view3d.**camera_to_view_selected()**

Move the camera so selected objects are framed

bpy.ops.view3d.**clear_render_border()**

Clear the boundaries of the border render and disable border render

bpy.ops.view3d.**clip_border(*, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True)**

Set the view clipping 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.view3d.**copybuffer()**

Copy the selected objects to the internal clipboard

bpy.ops.view3d.**cursor3d(*, use_depth=True, orientation='VIEW')**

Set the location of the 3D cursor

PARAMETERS:

- **use_depth** (*boolean, (optional)*) – Surface Project, Project onto the surface

- **orientation** (*enum in ['NONE', 'VIEW', 'XFORM', 'GEOM'], (optional)*) – Orientation, Preset viewpoint to use
  - `NONE` None – Leave orientation unchanged.
  - `VIEW` View – Orient to the viewport.
  - `XFORM` Transform – Orient to the current transform setting.
  - `GEOM` Geometry – Match the surface normal.

bpy.ops.view3d.**dolly(*, mx=0, my=0, delta=0, use_cursor_init=True)**

Dolly in/out in the view

PARAMETERS:

- **mx** (*int in [0, inf], (optional)*) – Region Position X
- **my** (*int in [0, inf], (optional)*) – Region Position Y
- **delta** (*int in [-inf, inf], (optional)*) – Delta
- **use_cursor_init** (*boolean, (optional)*) – Use Mouse Position, Allow the initial mouse position to be used

bpy.ops.view3d.**drop_world(*, name='', session_uid=0)**

Drop a world into the scene

PARAMETERS:

- **name** (*string, (optional, never None)*) – Name, Name of the data-block to use by the operator
- **session_uid** (*int in [-inf, inf], (optional)*) – Session UID, Session UID of the data-block to use by the operator

bpy.ops.view3d.**edit_mesh_extrude_individual_move()**

Extrude each individual face separately along local normals

FILE:

startup/bl_operators/view3d.py:31

bpy.ops.view3d.**edit_mesh_extrude_manifold_normal()**

Extrude manifold region along normals

FILE:

startup/bl_operators/view3d.py:202

bpy.ops.view3d.**edit_mesh_extrude_move_normal(*, dissolve_and_intersect=False)**

Extrude region together along the average normal

PARAMETERS:

dissolve_and_intersect (*boolean, (optional)*) – dissolve_and_intersect, Dissolves adjacent faces and intersects new geometry

FILE:

startup/bl_operators/view3d.py:168

bpy.ops.view3d.**edit_mesh_extrude_move_shrink_fatten()**

Extrude region together along local normals

FILE:

startup/bl_operators/view3d.py:185

bpy.ops.view3d.**fly()**

Interactively fly around the scene

bpy.ops.view3d.**interactive_add(*, primitive_type='CUBE', plane_origin_base='EDGE', plane_origin_depth='EDGE', plane_aspect_base='FREE', plane_aspect_depth='FREE', wait_for_input=True)**

Interactively add an object

Interactively add an object

PARAMETERS:

- **primitive_type** (*enum in ['CUBE', 'CYLINDER', 'CONE', 'SPHERE_UV', 'SPHERE_ICO'], (optional)*) – Primitive
- **plane_origin_base** (*enum in ['EDGE', 'CENTER'], (optional)*) –
  Origin, The initial position for placement
  - `EDGE` Edge – Start placing the edge position.
  - `CENTER` Center – Start placing the center position.
- **plane_origin_depth** (*enum in ['EDGE', 'CENTER'], (optional)*) –
  Origin, The initial position for placement
  - `EDGE` Edge – Start placing the edge position.
  - `CENTER` Center – Start placing the center position.
- **plane_aspect_base** (*enum in ['FREE', 'FIXED'], (optional)*) –
  Aspect, The initial aspect setting
  - `FREE` Free – Use an unconstrained aspect.
  - `FIXED` Fixed – Use a fixed 1:1 aspect.
- **plane_aspect_depth** (*enum in ['FREE', 'FIXED'], (optional)*) –
  Aspect, The initial aspect setting
  - `FREE` Free – Use an unconstrained aspect.
  - `FIXED` Fixed – Use a fixed 1:1 aspect.
- **wait_for_input** (*boolean, (optional)*) – Wait for Input

bpy.ops.view3d.**localview(*, frame_selected=True)**

Toggle display of selected object(s) separately and centered in view

PARAMETERS:

frame_selected (*boolean, (optional)*) – Frame Selected, Move the view to frame the selected objects

bpy.ops.view3d.**localview_remove_from()**

Move selected objects out of local view

bpy.ops.view3d.**move(*, use_cursor_init=True)**

Move the view

PARAMETERS:

use_cursor_init (*boolean, (optional)*) – Use Mouse Position, Allow the initial mouse position to be used

bpy.ops.view3d.**navigate()**

Interactively navigate around the scene (uses the mode (walk/fly) preference)

bpy.ops.view3d.**ndof_all()**

Pan and rotate the view with the 3D mouse

bpy.ops.view3d.**ndof_orbit()**

Orbit the view using the 3D mouse

bpy.ops.view3d.**ndof_orbit_zoom()**

Orbit and zoom the view using the 3D mouse

bpy.ops.view3d.**ndof_pan()**

Pan the view with the 3D mouse

bpy.ops.view3d.**object_as_camera()**

Set the active object as the active camera for this view or scene

bpy.ops.view3d.**object_mode_pie_or_toggle()**

Undocumented, consider contributing.

bpy.ops.view3d.**pastebuffer(*, autoselect=True, active_collection=True)**

Paste objects from the internal clipboard

PARAMETERS:

- **autoselect** (*boolean, (optional)*) – Select, Select pasted objects
- **active_collection** (*boolean, (optional)*) – Active Collection, Put pasted objects in the active collection

bpy.ops.view3d.**render_border(*, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True)**

Set the boundaries of the border render and enable border render

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.view3d.**rotate(*, use_cursor_init=True)**

Rotate the view

PARAMETERS:

use_cursor_init (*boolean, (optional)*) – Use Mouse Position, Allow the initial mouse position to be used

bpy.ops.view3d.**ruler_add()**

Add ruler

bpy.ops.view3d.**ruler_remove()**

Undocumented, consider contributing.

bpy.ops.view3d.**select(*, extend=False, deselect=False, toggle=False, deselect_all=False, select_passthrough=False, center=False, enumerate=False, object=False, location=(0, 0))**

Select and activate item(s)

PARAMETERS:

- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- **deselect** (*boolean, (optional)*) – Deselect, Remove from selection
- **toggle** (*boolean, (optional)*) – Toggle Selection, Toggle the selection
- **deselect_all** (*boolean, (optional)*) – Deselect On Nothing, Deselect all when nothing under the cursor
- **select_passthrough** (*boolean, (optional)*) – Only Select Unselected, Ignore the select action when the element is already selected
- **center** (*boolean, (optional)*) – Center, Use the object center when selecting, in edit mode used to extend object selection
- **enumerate** (*boolean, (optional)*) – Enumerate, List objects under the mouse (object mode only)
- **object** (*boolean, (optional)*) – Object, Use object selection (edit mode only)
- **location** (*int array of 2 items in [-inf, inf], (optional)*) – Location, Mouse location

bpy.ops.view3d.**select_box(*, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True, mode='SET')**

Select items using box selection

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
- **mode** (*enum in ['SET', 'ADD', 'SUB', 'XOR', 'AND'], (optional)*) –

  Mode

  - `SET` Set – Set a new selection.
  - `ADD` Extend – Extend existing selection.
  - `SUB` Subtract – Subtract existing selection.
  - `XOR` Difference – Invert existing selection.
  - `AND` Intersect – Intersect existing selection.

bpy.ops.view3d.**select_circle(*, x=0, y=0, radius=25, wait_for_input=True, mode='SET')**

Select items using circle selection

PARAMETERS:

- **x** (*int in [-inf, inf], (optional)*) – X
- **y** (*int in [-inf, inf], (optional)*) – Y
- **radius** (*int in [1, inf], (optional)*) – Radius
- **wait_for_input** (*boolean, (optional)*) – Wait for Input
- **mode** (*enum in ['SET', 'ADD', 'SUB'], (optional)*) –

  Mode

  - `SET` Set – Set a new selection.
  - `ADD` Extend – Extend existing selection.
  - `SUB` Subtract – Subtract existing selection.

bpy.ops.view3d.**select_lasso(*, path=None, use_smooth_stroke=False, smooth_stroke_factor=0.75, smooth_stroke_radius=35, mode='SET')**

Select items using lasso selection

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
- **mode** (*enum in ['SET', 'ADD', 'SUB', 'XOR', 'AND'], (optional)*) –

  Mode

  - `SET` Set – Set a new selection.
  - `ADD` Extend – Extend existing selection.
  - `SUB` Subtract – Subtract existing selection.
  - `XOR` Difference – Invert existing selection.
  - `AND` Intersect – Intersect existing selection.

bpy.ops.view3d.**select_menu(*, name='', extend=False, deselect=False, toggle=False)**

Menu object selection

PARAMETERS:

- **name** (*enum in [], (optional)*) – Object Name
- **extend** (*boolean, (optional)*) – Extend
- **deselect** (*boolean, (optional)*) – Deselect

- **deselect** (*boolean, (optional)*) – Deselect
- **toggle** (*boolean, (optional)*) – Toggle

bpy.ops.view3d.**smoothview()**

Undocumented, consider contributing.

bpy.ops.view3d.**snap_cursor_to_active()**

Snap 3D cursor to the active item

bpy.ops.view3d.**snap_cursor_to_center()**

Snap 3D cursor to the world origin

bpy.ops.view3d.**snap_cursor_to_grid()**

Snap 3D cursor to the nearest grid division

bpy.ops.view3d.**snap_cursor_to_selected()**

Snap 3D cursor to the middle of the selected item(s)

bpy.ops.view3d.**snap_selected_to_active()**

Snap selected item(s) to the active item

bpy.ops.view3d.**snap_selected_to_cursor(*, use_offset=True)**

Snap selected item(s) to the 3D cursor

PARAMETERS:

use_offset (*boolean, (optional)*) – Offset, If the selection should be snapped as a whole or by each object center

bpy.ops.view3d.**snap_selected_to_grid()**

Snap selected item(s) to their nearest grid division

bpy.ops.view3d.**toggle_matcap_flip()**

Flip MatCap

bpy.ops.view3d.**toggle_shading(*, type='WIREFRAME')**

Toggle shading type in 3D viewport

PARAMETERS:

type (*enum in ['WIREFRAME', 'SOLID', 'MATERIAL', 'RENDERED'], (optional)*) –

Type, Shading type to toggle

- `WIREFRAME` Wireframe – Toggle wireframe shading.
- `SOLID` Solid – Toggle solid shading.
- `MATERIAL` Material Preview – Toggle material preview shading.
- `RENDERED` Rendered – Toggle rendered shading.

bpy.ops.view3d.**toggle_xray()**

Transparent scene display. Allow selecting through items

bpy.ops.view3d.**transform_gizmo_set(*, extend=False, type={})**

Set the current transform gizmo

PARAMETERS:

- **extend** (*boolean, (optional)*) – Extend
- **type** (*enum set in {'TRANSLATE', 'ROTATE', 'SCALE'}, (optional)*) – Type

FILE:

startup/bl_operators/view3d.py:240

startup/bl_operators/view3d.py:249

bpy.ops.view3d.**view_all(*, use_all_regions=False, center=False)**

View all objects in scene

PARAMETERS:

- **use_all_regions** (*boolean, (optional)*) – All Regions, View selected for all regions
- **center** (*boolean, (optional)*) – Center

bpy.ops.view3d.**view_axis(*, type='LEFT', align_active=False, relative=False)**

Use a preset viewpoint

PARAMETERS:

- **type** (*enum in ['LEFT', 'RIGHT', 'BOTTOM', 'TOP', 'FRONT', 'BACK'], (optional)*) – View, Preset viewpoint to use
  - `LEFT` Left – View from the left.
  - `RIGHT` Right – View from the right.
  - `BOTTOM` Bottom – View from the bottom.
  - `TOP` Top – View from the top.
  - `FRONT` Front – View from the front.
  - `BACK` Back – View from the back.
- **align_active** (*boolean, (optional)*) – Align Active, Align to the active object's axis
- **relative** (*boolean, (optional)*) – Relative, Rotate relative to the current orientation

bpy.ops.view3d.**view_camera()**

Toggle the camera view

bpy.ops.view3d.**view_center_camera()**

Center the camera view, resizing the view to fit its bounds

bpy.ops.view3d.**view_center_cursor()**

Center the view so that the cursor is in the middle of the view

bpy.ops.view3d.**view_center_lock()**

Center the view lock offset

bpy.ops.view3d.**view_center_pick()**

Center the view to the Z-depth position under the mouse cursor

bpy.ops.view3d.**view_lock_clear()**

Clear all view locking

bpy.ops.view3d.**view_lock_to_active()**

Lock the view to the active object/bone

bpy.ops.view3d.**view_orbit(*, angle=0.0, type='ORBITLEFT')**

Orbit the view

PARAMETERS:

- **angle** (*float in [-inf, inf], (optional)*) – Roll
- **type** (*enum in ['ORBITLEFT', 'ORBITRIGHT', 'ORBITUP', 'ORBITDOWN'], (optional)*) – Orbit, Direction of View Orbit
  - `ORBITLEFT` Orbit Left – Orbit the view around to the left.

    - `ORBITRIGHT` Orbit Right – Orbit the view around to the right.
    - `ORBITUP` Orbit Up – Orbit the view up.
    - `ORBITDOWN` Orbit Down – Orbit the view down.

bpy.ops.view3d.**view_pan(*, type='PANLEFT')**

Pan the view in a given direction

PARAMETERS:

type (*enum in ['PANLEFT', 'PANRIGHT', 'PANUP', 'PANDOWN'], (optional)*) –

Pan, Direction of View Pan

- `PANLEFT` Pan Left – Pan the view to the left.
- `PANRIGHT` Pan Right – Pan the view to the right.
- `PANUP` Pan Up – Pan the view up.
- `PANDOWN` Pan Down – Pan the view down.

bpy.ops.view3d.**view_persportho()**

Switch the current view from perspective/orthographic projection

bpy.ops.view3d.**view_roll(*, angle=0.0, type='ANGLE')**

Roll the view

PARAMETERS:

- **angle** (*float in [-inf, inf], (optional)*) – Roll
- **type** (*enum in ['ANGLE', 'LEFT', 'RIGHT'], (optional)*) –
  Roll Angle Source, How roll angle is calculated
    - `ANGLE` Roll Angle – Roll the view using an angle value.
    - `LEFT` Roll Left – Roll the view around to the left.
    - `RIGHT` Roll Right – Roll the view around to the right.

bpy.ops.view3d.**view_selected(*, use_all_regions=False)**

Move the view to the selection center

PARAMETERS:

use_all_regions (*boolean, (optional)*) – All Regions, View selected for all regions

bpy.ops.view3d.**walk()**

Interactively walk around the scene

bpy.ops.view3d.**zoom(*, mx=0, my=0, delta=0, use_cursor_init=True)**

Zoom in/out in the view

PARAMETERS:

- **mx** (*int in [0, inf], (optional)*) – Region Position X
- **my** (*int in [0, inf], (optional)*) – Region Position Y
- **delta** (*int in [-inf, inf], (optional)*) – Delta
- **use_cursor_init** (*boolean, (optional)*) – Use Mouse Position, Allow the initial mouse position to be used

bpy.ops.view3d.**zoom_border(*, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True, zoom_out=False)**

Zoom in the view to the nearest object contained in the border

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
- **zoom_out** (*boolean, (optional)*) – Zoom Out

bpy.ops.view3d.**zoom_camera_1_to_1()**

Match the camera to 1:1 to the render output