Skip to content

Action Operators

bpy.ops.action.**bake_keys()**

Add keyframes on every frame between the selected keyframes

bpy.ops.action.**clean(*, threshold=0.001, channels=False)**

Simplify F-Curves by removing closely spaced keyframes

PARAMETERS:

- **threshold** (*float in [0, inf], (optional)*) – Threshold
- **channels** (*boolean, (optional)*) – Channels

bpy.ops.action.**clickselect(*, wait_to_deselect_others=False, mouse_x=0, mouse_y=0, extend=False, deselect_all=False, column=False, channel=False)**

Select keyframes by clicking on them

PARAMETERS:

- **wait_to_deselect_others** (*boolean, (optional)*) – Wait to Deselect Others
- **mouse_x** (*int in [-inf, inf], (optional)*) – Mouse X
- **mouse_y** (*int in [-inf, inf], (optional)*) – Mouse Y
- **extend** (*boolean, (optional)*) – Extend Select, Toggle keyframe selection instead of leaving newly selected keyframes only
- **deselect_all** (*boolean, (optional)*) – Deselect On Nothing, Deselect all when nothing under the cursor
- **column** (*boolean, (optional)*) – Column Select, Select all keyframes that occur on the same frame as the one under the mouse
- **channel** (*boolean, (optional)*) – Only Channel, Select all the keyframes in the channel under the mouse

bpy.ops.action.**copy()**

Copy selected keyframes to the internal clipboard

bpy.ops.action.**delete(*, confirm=True)**

Remove all selected keyframes

PARAMETERS:

confirm (*boolean, (optional)*) – Confirm, Prompt for confirmation

bpy.ops.action.**duplicate()**

Make a copy of all selected keyframes

bpy.ops.action.**duplicate_move(*, ACTION_OT_duplicate=None, TRANSFORM_OT_transform=None)**

Make a copy of all selected keyframes and move them

PARAMETERS:

- **ACTION_OT_duplicate** (`ACTION_OT_duplicate`, (optional)) – Duplicate Keyframes, Make a copy of all selected keyframes
- **TRANSFORM_OT_transform** (`TRANSFORM_OT_transform`, (optional)) – Transform, Transform selected items by mode type

bpy.ops.action.**easing_type(*, type='AUTO')**

Set easing type for the F-Curve segments starting from the selected keyframes

PARAMETERS:

type (enum in Beztriple Interpolation Easing Items, (optional)) – Type

bpy.ops.action.**extrapolation_type(*, type='CONSTANT')**

Set extrapolation mode for selected F-Curves

PARAMETERS:

type (*enum in ['CONSTANT', 'LINEAR', 'MAKE_CYCLIC', 'CLEAR_CYCLIC'], (optional)*) –

Type

- `CONSTANT` Constant Extrapolation – Values on endpoint keyframes are held.
- `LINEAR` Linear Extrapolation – Straight-line slope of end segments are extended past the endpoint keyframes.
- `MAKE_CYCLIC` Make Cyclic (F-Modifier) – Add Cycles F-Modifier if one doesn't exist already.
- `CLEAR_CYCLIC` Clear Cyclic (F-Modifier) – Remove Cycles F-Modifier if not needed anymore.

bpy.ops.action.**frame_jump()**

Set the current frame to the average frame value of selected keyframes

bpy.ops.action.**handle_type(*, type='FREE')**

Set type of handle for selected keyframes

PARAMETERS:

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

bpy.ops.action.**interpolation_type(*, type='CONSTANT')**

Set interpolation mode for the F-Curve segments starting from the selected keyframes

PARAMETERS:

type (enum in Beztriple Interpolation Mode Items, (optional)) – Type

bpy.ops.action.**keyframe_insert(*, type='ALL')**

Insert keyframes for the specified channels

PARAMETERS:

type (*enum in ['ALL', 'SEL', 'GROUP'], (optional)*) – Type

bpy.ops.action.**keyframe_type(*, type='KEYFRAME')**

Set type of keyframe for the selected keyframes

PARAMETERS:

type (enum in Beztriple Keyframe Type Items, (optional)) – Type

bpy.ops.action.**layer_next()**

Switch to editing action in animation layer above the current action in the NLA Stack

bpy.ops.action.**layer_prev()**

Switch to editing action in animation layer below the current action in the NLA Stack

bpy.ops.action.**markers_make_local()**

Move selected scene markers to the active Action as local 'pose' markers

bpy.ops.action.**mirror(*, type='CFRA')**

Flip selected keyframes over the selected mirror line

PARAMETERS:

type (*enum in ['CFRA', 'XAXIS', 'MARKER'], (optional)*) –

Type

- `CFRA` By Times Over Current Frame – Flip times of selected keyframes using the current frame as the mirror line.
- `XAXIS` By Values Over Zero Value – Flip values of selected keyframes (i.e. negative values become positive, and vice versa).
- `MARKER` By Times Over First Selected Marker – Flip times of selected keyframes using the first selected marker as the reference point.

bpy.ops.action.**new()**

Create new action

bpy.ops.action.**paste(*, offset='START', merge='MIX', flipped=False)**

Paste keyframes from the internal clipboard for the selected channels, starting on the current frame

PARAMETERS:

- **offset** (enum in Keyframe Paste Offset Items, (optional)) – Offset, Paste time offset of keys
- **merge** (enum in Keyframe Paste Merge Items, (optional)) – Type, Method of merging pasted keys and existing
- **flipped** (*boolean, (optional)*) – Flipped, Paste keyframes from mirrored bones if they exist

bpy.ops.action.**previewrange_set()**

Set Preview Range based on extents of selected Keyframes

bpy.ops.action.**push_down()**

Push action down on to the NLA stack as a new strip

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

Toggle selection of all keyframes

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.action.**select_box(*, axis_range=False, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True, mode='SET', tweak=False)**

Select all keyframes within the specified region

PARAMETERS:

- **axis_range** (*boolean, (optional)*) – Axis Range
- **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'], (optional)*) –

  Mode

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

- **tweak** (*boolean, (optional)*) – Tweak, Operator has been activated using a click-drag event

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

Select keyframe points 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

- **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.action.**select_column(*, mode='KEYS')**

Select all keyframes on the specified frame(s)

PARAMETERS:

mode (*enum in ['KEYS', 'CFRA', 'MARKERS_COLUMN', 'MARKERS_BETWEEN'], (optional)*) – Mode

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

Select keyframe points 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'], (optional)*) –

  Mode

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

bpy.ops.action.**select_leftright(*, mode='CHECK', extend=False)**

Select keyframes to the left or the right of the current frame

PARAMETERS:

- **mode** (*enum in ['CHECK', 'LEFT', 'RIGHT'], (optional)*) – Mode
- **extend** (*boolean, (optional)*) – Extend Select

bpy.ops.action.**select_less()**

Deselect keyframes on ends of selection islands

bpy.ops.action.**select_linked()**

Select keyframes occurring in the same F-Curves as selected ones

bpy.ops.action.**select_more()**

Select keyframes beside already selected ones

bpy.ops.action.**snap(*, type='CFRA')**

Snap selected keyframes to the times specified

PARAMETERS:

type (*enum in ['CFRA', 'NEAREST_FRAME', 'NEAREST_SECOND', 'NEAREST_MARKER'], (optional)*) –

Type

- `CFRA` Selection to Current Frame – Snap selected keyframes to the current frame.
- `NEAREST_FRAME` Selection to Nearest Frame – Snap selected keyframes to the nearest (whole) frame (use to fix accidental subframe offsets).

- `NEAREST_SECOND` Selection to Nearest Second – Snap selected keyframes to the nearest second.
- `NEAREST_MARKER` Selection to Nearest Marker – Snap selected keyframes to the nearest marker.

bpy.ops.action.**stash(*, create_new=True)**

Store this action in the NLA stack as a non-contributing strip for later use

PARAMETERS:

create_new (*boolean, (optional)*) – Create New Action, Create a new action once the existing one has been safely stored

bpy.ops.action.**stash_and_create()**

Store this action in the NLA stack as a non-contributing strip for later use, and create a new action

bpy.ops.action.**unlink(*, force_delete=False)**

Unlink this action from the active action slot (and/or exit Tweak Mode)

PARAMETERS:

force_delete (*boolean, (optional)*) – Force Delete, Clear Fake User and remove copy stashed in this data-block's NLA stack

bpy.ops.action.**view_all()**

Reset viewable area to show full keyframe range

bpy.ops.action.**view_frame()**

Move the view to the current frame

bpy.ops.action.**view_selected()**

Reset viewable area to show selected keyframes range