Graph Operators

bpy.ops.graph.**bake_keys()**

Add keyframes on every frame between the selected keyframes

bpy.ops.graph.**blend_offset(*, factor=0.0)**

Shift selected keys to the value of the neighboring keys as a block

PARAMETERS:

factor (*float in [-inf, inf], (optional)*) – Offset Factor, Control which key to offset towards and how far

bpy.ops.graph.**blend_to_default(*, factor=0.0)**

Blend selected keys to their default value from their current position

PARAMETERS:

factor (*float in [-inf, inf], (optional)*) – Factor, How much to blend to the default value

bpy.ops.graph.**blend_to_ease(*, factor=0.0)**

Blends keyframes from current state to an ease-in or ease-out curve

PARAMETERS:

factor (*float in [-inf, inf], (optional)*) – Blend, Favor either original data or ease curve

bpy.ops.graph.**blend_to_neighbor(*, factor=0.0)**

Blend selected keyframes to their left or right neighbor

PARAMETERS:

factor (*float in [-inf, inf], (optional)*) – Blend, The blend factor with 0 being the current frame

bpy.ops.graph.**breakdown(*, factor=0.0)**

Move selected keyframes to an inbetween position relative to adjacent keys

PARAMETERS:

factor (*float in [-inf, inf], (optional)*) – Factor, Favor either the left or the right key

bpy.ops.graph.**butterworth_smooth(*, cutoff_frequency=3.0, filter_order=4, samples_per_frame=1, blend=1.0, blend_in_out=1)**

Smooth an F-Curve while maintaining the general shape of the curve

PARAMETERS:

- **cutoff_frequency** (*float in [0, inf], (optional)*) – Frequency Cutoff (Hz), Lower values give a smoother curve
- **filter_order** (*int in [1, 32], (optional)*) – Filter Order, Higher values produce a harder frequency cutoff
- **samples_per_frame** (*int in [1, 64], (optional)*) – Samples per Frame, How many samples to calculate per frame, helps with subframe data
- **blend** (*float in [0, inf], (optional)*) – Blend, How much to blend to the smoothed curve
- **blend_in_out** (*int in [0, inf], (optional)*) – Blend In/Out, Linearly blend the smooth data to the border frames of the selection

bpy.ops.graph.**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.graph.**click_insert(*, frame=1.0, value=1.0, extend=False)**

Insert new keyframe at the cursor position for the active F-Curve

PARAMETERS:

- **frame** (*float in [-inf, inf], (optional)*) – Frame Number, Frame to insert keyframe on
- **value** (*float in [-inf, inf], (optional)*) – Value, Value for keyframe on
- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first

bpy.ops.graph.**clickselect(*, wait_to_deselect_others=False, mouse_x=0, mouse_y=0, extend=False, deselect_all=False, column=False, curves=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
- **curves** (*boolean, (optional)*) – Only Curves, Select all the keyframes in the curve

bpy.ops.graph.**copy()**

Copy selected keyframes to the internal clipboard

bpy.ops.graph.**cursor_set(*, frame=0.0, value=0.0)**

Interactively set the current frame and value cursor

PARAMETERS:

- **frame** (*float in [-1.04857e+06, 1.04857e+06], (optional)*) – Frame
- **value** (*float in [-inf, inf], (optional)*) – Value

bpy.ops.graph.**decimate(*, mode='RATIO', factor=0.333333, remove_error_margin=0.0)**

Decimate F-Curves by removing keyframes that influence the curve shape the least

PARAMETERS:

- **mode** (*enum in ['RATIO', 'ERROR'], (optional)*) –
  Mode, Which mode to use for decimation
  - `RATIO` Ratio – Use a percentage to specify how many keyframes you want to remove.
  - `ERROR` Error Margin – Use an error margin to specify how much the curve is allowed to deviate from the original path.
- **factor** (*float in [0, 1], (optional)*) – Factor, The ratio of keyframes to remove
- **remove_error_margin** (*float in [0, inf], (optional)*) – Max Error Margin, How much the new decimated curve is allowed to deviate from t original

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

Remove all selected keyframes

PARAMETERS:

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

bpy.ops.graph.**driver_delete_invalid()**

Delete all visible drivers considered invalid

bpy.ops.graph.**driver_variables_copy()**

Copy the driver variables of the active driver

bpy.ops.graph.**driver_variables_paste(*, replace=False)**

Add variables to the active driver

Add copied driver variables to the active driver

PARAMETERS:

replace (*boolean, (optional)*) – Replace Existing, Replace existing driver variables, instead of just appending to the end of the existing list

bpy.ops.graph.**duplicate(*, mode='TRANSLATION')**

Make a copy of all selected keyframes

PARAMETERS:

mode (enum in Transform Mode Type Items, (optional)) – Mode

bpy.ops.graph.**duplicate_move(*, GRAPH_OT_duplicate=None, TRANSFORM_OT_translate=None)**

Make a copy of all selected keyframes and move them

PARAMETERS:

- **GRAPH_OT_duplicate** (`GRAPH_OT_duplicate`, (optional)) – Duplicate Keyframes, Make a copy of all selected keyframes
- **TRANSFORM_OT_translate** (`TRANSFORM_OT_translate`, (optional)) – Move, Move selected items

bpy.ops.graph.**ease(*, factor=0.0, sharpness=2.0)**

Align keyframes on a ease-in or ease-out curve

PARAMETERS:

- **factor** (*float in [-inf, inf], (optional)*) – Curve Bend, Defines if the keys should be aligned on an ease-in or ease-out curve
- **sharpness** (*float in [0.001, inf], (optional)*) – Sharpness, Higher values make the change more abrupt

bpy.ops.graph.**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.graph.**equalize_handles(*, side='LEFT', handle_length=5.0, flatten=False)**

Ensure selected keyframes' handles have equal length, optionally making them horizontal. Automatic, Automatic Clamped, or Vector handle types will be converted to Aligned

PARAMETERS:

- **side** (*enum in ['LEFT', 'RIGHT', 'BOTH'], (optional)*) –

  Side, Side of the keyframes' Bézier handles to affect

  - `LEFT` Left – Equalize selected keyframes' left handles.
  - `RIGHT` Right – Equalize selected keyframes' right handles.
  - `BOTH` Both – Equalize both of a keyframe's handles.

- **handle_length** (*float in [0.1, inf], (optional)*) – Handle Length, Length to make selected keyframes' Bézier handles
- **flatten** (*boolean, (optional)*) – Flatten, Make the values of the selected keyframes' handles the same as their respective keyframes

bpy.ops.graph.**euler_filter()**

Fix large jumps and flips in the selected Euler Rotation F-Curves arising from rotation values being clipped when baking physics

bpy.ops.graph.**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.

- 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.graph.**fmodifier_add(*, type='NULL', only_active=False)**

Add F-Modifier to the active/selected F-Curves

PARAMETERS:

- **type** (enum in Fmodifier Type Items, (optional)) – Type
- **only_active** (*boolean, (optional)*) – Only Active, Only add F-Modifier to active F-Curve

bpy.ops.graph.**fmodifier_copy()**

Copy the F-Modifier(s) of the active F-Curve

bpy.ops.graph.**fmodifier_paste(*, only_active=False, replace=False)**

Add copied F-Modifiers to the selected F-Curves

PARAMETERS:

- **only_active** (*boolean, (optional)*) – Only Active, Only paste F-Modifiers on active F-Curve
- **replace** (*boolean, (optional)*) – Replace Existing, Replace existing F-Modifiers, instead of just appending to the end of the existing list

bpy.ops.graph.**frame_jump()**

Place the cursor on the midpoint of selected keyframes

bpy.ops.graph.**gaussian_smooth(*, factor=1.0, sigma=0.33, filter_width=6)**

Smooth the curve using a Gaussian filter

PARAMETERS:

- **factor** (*float in [0, inf], (optional)*) – Factor, How much to blend to the default value
- **sigma** (*float in [0.001, inf], (optional)*) – Sigma, The shape of the gaussian distribution, lower values make it sharper
- **filter_width** (*int in [1, 64], (optional)*) – Filter Width, How far to each side the operator will average the key values

bpy.ops.graph.**ghost_curves_clear()**

Clear F-Curve snapshots (Ghosts) for active Graph Editor

bpy.ops.graph.**ghost_curves_create()**

Create snapshot (Ghosts) of selected F-Curves as background aid for active Graph Editor

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

Set type of handle for selected keyframes

PARAMETERS:

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

bpy.ops.graph.**hide(*, unselected=False)**

Hide selected curves from Graph Editor view

PARAMETERS:

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

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

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

Insert keyframes for the specified channels

PARAMETERS:

type (*enum in ['ALL', 'SEL', 'ACTIVE', 'CURSOR_ACTIVE', 'CURSOR_SEL'], (optional)*) –

Type

- `ALL` All Channels – Insert a keyframe on all visible and editable F-Curves using each curve's current value.
- `SEL` Only Selected Channels – Insert a keyframe on selected F-Curves using each curve's current value.
- `ACTIVE` Only Active F-Curve – Insert a keyframe on the active F-Curve using the curve's current value.
- `CURSOR_ACTIVE` Active Channels at Cursor – Insert a keyframe for the active F-Curve at the cursor point.
- `CURSOR_SEL` Selected Channels at Cursor – Insert a keyframe for selected F-Curves at the cursor point.

bpy.ops.graph.**keyframe_jump(*, next=True)**

Jump to previous/next keyframe

PARAMETERS:

next (*boolean, (optional)*) – Next Keyframe

bpy.ops.graph.**keys_to_samples()**

Convert selected channels to an uneditable set of samples to save storage space

bpy.ops.graph.**match_slope(*, factor=0.0)**

Blend selected keys to the slope of neighboring ones

PARAMETERS:

factor (*float in [-inf, inf], (optional)*) – Factor, Defines which keys to use as slope and how much to blend towards them

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

Flip selected keyframes over the selected mirror line

PARAMETERS:

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

Type

- `CFRA` By Times Over Current Frame – Flip times of selected keyframes using the current frame as the mirror line.
- `VALUE` By Values Over Cursor Value – Flip values of selected keyframes using the cursor value (Y/Horizontal component) as the mirror lin
- `YAXIS` By Times Over Zero Time – Flip times of selected keyframes, effectively reversing the order they appear in.
- `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.graph.**paste(*, offset='START', value_offset='NONE', 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)) – Frame Offset, Paste time offset of keys
- **value_offset** (enum in Keyframe Paste Offset Value Items, (optional)) – Value Offset, Paste keys with a value offset
- **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.graph.**previewrange_set()**

Set Preview Range based on range of selected keyframes

bpy.ops.graph.**push_pull(*, factor=1.0)**

Exaggerate or minimize the value of the selected keys

PARAMETERS:

factor (*float in [-inf, inf], (optional)*) – Factor, Control how far to push or pull the keys

bpy.ops.graph.**reveal(*, select=True)**

Make previously hidden curves visible again in Graph Editor view

PARAMETERS:

select (*boolean, (optional)*) – Select

bpy.ops.graph.**samples_to_keys()**

Convert selected channels from samples to keyframes

bpy.ops.graph.**scale_average(*, factor=1.0)**

Scale selected key values by their combined average

PARAMETERS:

factor (*float in [-inf, inf], (optional)*) – Scale Factor, The scale factor applied to the curve segments

bpy.ops.graph.**scale_from_neighbor(*, factor=0.0, anchor='LEFT')**

Increase or decrease the value of selected keys in relationship to the neighboring one

PARAMETERS:

- **factor** (*float in [-inf, inf], (optional)*) – Factor, The factor to scale keys with
- **anchor** (*enum in ['LEFT', 'RIGHT'], (optional)*) – Reference Key, Which end of the segment to use as a reference to scale from

bpy.ops.graph.**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.graph.**select_box(*, axis_range=False, include_handles=True, tweak=False, use_curve_selection=True, xmin=0, xmax=0, ymin=0 ymax=0, wait_for_input=True, mode='SET')**

Select all keyframes within the specified region

PARAMETERS:

- **axis_range** (*boolean, (optional)*) – Axis Range
- **include_handles** (*boolean, (optional)*) – Include Handles, Are handles tested individually against the selection criteria
- **tweak** (*boolean, (optional)*) – Tweak, Operator has been activated using a click-drag event
- **use_curve_selection** (*boolean, (optional)*) – Select Curves, Allow selecting all the keyframes of a curve by selecting the calculated F-curve
- **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.

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

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
- **mode** (*enum in ['SET', 'ADD', 'SUB'], (optional)*) –
  Mode
  - SET Set – Set a new selection.
  - ADD Extend – Extend existing selection.
  - SUB Subtract – Subtract existing selection.
- **use_curve_selection** (*boolean, (optional)*) – Select Curves, Allow selecting all the keyframes of a curve by selecting the curve itself

bpy.ops.graph.**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.graph.**select_key_handles(*, left_handle_action='SELECT', right_handle_action='SELECT', key_action='KEEP')**

For selected keyframes, select/deselect any combination of the key itself and its handles

PARAMETERS:

- **left_handle_action** (*enum in ['SELECT', 'DESELECT', 'KEEP'], (optional)*) –
  Left Handle, Effect on the left handle
  - SELECT Select.
  - DESELECT Deselect.
  - KEEP Keep – Leave as is.
- **right_handle_action** (*enum in ['SELECT', 'DESELECT', 'KEEP'], (optional)*) –
  Right Handle, Effect on the right handle
  - SELECT Select.
  - DESELECT Deselect.
  - KEEP Keep – Leave as is.
- **key_action** (*enum in ['SELECT', 'DESELECT', 'KEEP'], (optional)*) –
  Key, Effect on the key itself
  - SELECT Select.
  - DESELECT Deselect.
  - KEEP Keep – Leave as is.

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

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.
- **use_curve_selection** (*boolean, (optional)*) – Select Curves, Allow selecting all the keyframes of a curve by selecting the curve itself

bpy.ops.graph.**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.graph.**select_less()**

Deselect keyframes on ends of selection islands

bpy.ops.graph.**select_linked()**

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

bpy.ops.graph.**select_more()**

Select keyframes beside already selected ones

bpy.ops.graph.**shear(*, factor=0.0, direction='FROM_LEFT')**

Affect the value of the keys linearly, keeping the same relationship between them using either the left or the right key as reference

PARAMETERS:

- **factor** (*float in [-inf, inf], (optional)*) – Shear Factor, The amount of shear to apply
- **direction** (*enum in ['FROM_LEFT', 'FROM_RIGHT'], (optional)*) –
  Direction, Which end of the segment to use as a reference to shear from
  - `FROM_LEFT` From Left – Shear the keys using the left key as reference.
  - `FROM_RIGHT` From Right – Shear the keys using the right key as reference.

bpy.ops.graph.**smooth()**

Apply weighted moving means to make selected F-Curves less bumpy

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

Snap selected keyframes to the chosen times/values

PARAMETERS:

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

Type

- `CFRA` Selection to Current Frame – Snap selected keyframes to the current frame.
- `VALUE` Selection to Cursor Value – Set values of selected keyframes to the cursor value (Y/Horizontal component).
- `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.
- `HORIZONTAL` Flatten Handles – Flatten handles for a smoother transition

HORIZONTAL Match Handles – Match handles for a shallower transition.

bpy.ops.graph.**snap_cursor_value()**

Place the cursor value on the average value of selected keyframes

bpy.ops.graph.**sound_to_samples(*, filepath='', check_existing=False, filter_blender=False, filter_backup=False, filter_image=False, filter_movie=True, filter_python=False, filter_font=False, filter_sound=True, filter_text=False, filter_archive=False, filter_btx=False, filter_collada=False, filter_alembic=False, filter_usd=False, filter_obj=False, filter_volume=False, filter_folder=True, filter_blenlib=False, filemode=9, show_multiview=False, use_multiview=False, display_type='DEFAULT', sort_method='', low=0.0, high=100000.0, attack=0.005, release=0.2, threshold=0.0, use_accumulate=False, use_additive=False, use_square=False, sthreshold=0.1)**

Bakes a sound wave to samples on selected channels

PARAMETERS:

- **filepath** (*string, (optional, never None)*) – File Path, Path to file
- **check_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter_blender** (*boolean, (optional)*) – Filter .blend files
- **filter_backup** (*boolean, (optional)*) – Filter .blend files
- **filter_image** (*boolean, (optional)*) – Filter image files
- **filter_movie** (*boolean, (optional)*) – Filter movie files
- **filter_python** (*boolean, (optional)*) – Filter Python files
- **filter_font** (*boolean, (optional)*) – Filter font files
- **filter_sound** (*boolean, (optional)*) – Filter sound files
- **filter_text** (*boolean, (optional)*) – Filter text files
- **filter_archive** (*boolean, (optional)*) – Filter archive files
- **filter_btx** (*boolean, (optional)*) – Filter btx files
- **filter_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter_alembic** (*boolean, (optional)*) – Filter Alembic files
- **filter_usd** (*boolean, (optional)*) – Filter USD files
- **filter_obj** (*boolean, (optional)*) – Filter OBJ files
- **filter_volume** (*boolean, (optional)*) – Filter OpenVDB volume files
- **filter_folder** (*boolean, (optional)*) – Filter folders
- **filter_blenlib** (*boolean, (optional)*) – Filter Blender IDs
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **show_multiview** (*boolean, (optional)*) – Enable Multi-View
- **use_multiview** (*boolean, (optional)*) – Use Multi-View
- **display_type** (*enum in ['DEFAULT', 'LIST_VERTICAL', 'LIST_HORIZONTAL', 'THUMBNAIL'], (optional)*) – Display Type
  - `DEFAULT` Default – Automatically determine display type for files.
  - `LIST_VERTICAL` Short List – Display files as short list.
  - `LIST_HORIZONTAL` Long List – Display files as a detailed list.
  - `THUMBNAIL` Thumbnails – Display files as thumbnails.
- **sort_method** (*enum in [], (optional)*) – File sorting mode
- **low** (*float in [0, 100000], (optional)*) – Lowest Frequency, Cutoff frequency of a high-pass filter that is applied to the audio data
- **high** (*float in [0, 100000], (optional)*) – Highest Frequency, Cutoff frequency of a low-pass filter that is applied to the audio data
- **attack** (*float in [0, 2], (optional)*) – Attack Time, Value for the envelope calculation that tells how fast the envelope can rise (the lower the value the steeper it can rise)
- **release** (*float in [0, 5], (optional)*) – Release Time, Value for the envelope calculation that tells how fast the envelope can fall (the lower the value the steeper it can fall)
- **threshold** (*float in [0, 1], (optional)*) – Threshold, Minimum amplitude value needed to influence the envelope
- **use_accumulate** (*boolean, (optional)*) – Accumulate, Only the positive differences of the envelope amplitudes are summarized to produce t

output

- **use_additive** (*boolean, (optional)*) – Additive, The amplitudes of the envelope are summarized (or, when Accumulate is enabled, both positive and negative differences are accumulated)
- **use_square** (*boolean, (optional)*) – Square, The output is a square curve (negative values always result in -1, and positive ones in 1)
- **sthreshold** (*float in [0, 1], (optional)*) – Square Threshold, Square only: all values with an absolute amplitude lower than that result in 0

bpy.ops.graph.**time_offset(*, frame_offset=0.0)**

Shifts the value of selected keys in time

PARAMETERS:

frame_offset (*float in [-inf, inf], (optional)*) – Frame Offset, How far in frames to offset the animation

bpy.ops.graph.**view_all(*, include_handles=True)**

Reset viewable area to show full keyframe range

PARAMETERS:

include_handles (*boolean, (optional)*) – Include Handles, Include handles of keyframes when calculating extents

bpy.ops.graph.**view_frame()**

Move the view to the current frame

bpy.ops.graph.**view_selected(*, include_handles=True)**

Reset viewable area to show selected keyframe range

PARAMETERS:

include_handles (*boolean, (optional)*) – Include Handles, Include handles of keyframes when calculating extents