Skip to content

Armature Operators

bpy.ops.armature.**align()**

Align selected bones to the active bone (or to their parent)

bpy.ops.armature.**assign_to_collection(*, collection_index=-1, new_collection_name='')**

Assign all selected bones to a collection, or unassign them, depending on whether the active bone is already assigned or not

PARAMETERS:

- **collection_index** (*int in [-1, inf], (optional)*) – Collection Index, Index of the collection to assign selected bones to. When the operator should create a new bone collection, use new_collection_name to define the collection name, and set this parameter to the parent index of the new bone collection
- **new_collection_name** (*string, (optional, never None)*) – Name, Name of a to-be-added bone collection. Only pass this if you want to create a new bone collection and assign the selected bones to it. To assign to an existing collection, do not include this parameter and use collection_index

bpy.ops.armature.**autoside_names(*, type='XAXIS')**

Automatically renames the selected bones according to which side of the target axis they fall on

PARAMETERS:

type (*enum in ['XAXIS', 'YAXIS', 'ZAXIS'], (optional)*) –

Axis, Axis to tag names with

- `XAXIS` X-Axis – Left/Right.
- `YAXIS` Y-Axis – Front/Back.
- `ZAXIS` Z-Axis – Top/Bottom.

bpy.ops.armature.**bone_primitive_add(*, name='')**

Add a new bone located at the 3D cursor

PARAMETERS:

name (*string, (optional, never None)*) – Name, Name of the newly created bone

bpy.ops.armature.**calculate_roll(*, type='POS_X', axis_flip=False, axis_only=False)**

Automatically fix alignment of select bones' axes

PARAMETERS:

- **type** (*enum in ['POS_X', 'POS_Z', 'GLOBAL_POS_X', 'GLOBAL_POS_Y', 'GLOBAL_POS_Z', 'NEG_X', 'NEG_Z', 'GLOBAL_NEG_ 'GLOBAL_NEG_Y', 'GLOBAL_NEG_Z', 'ACTIVE', 'VIEW', 'CURSOR'], (optional)*) – Type
- **axis_flip** (*boolean, (optional)*) – Flip Axis, Negate the alignment axis
- **axis_only** (*boolean, (optional)*) – Shortest Rotation, Ignore the axis direction, use the shortest rotation to align

bpy.ops.armature.**click_extrude()**

Create a new bone going from the last selected joint to the mouse position

bpy.ops.armature.**collection_add()**

Add a new bone collection

bpy.ops.armature.**collection_assign(*, name='')**

Add selected bones to the chosen bone collection

PARAMETERS:

name (*string, (optional, never None)*) – Bone Collection, Name of the bone collection to assign this bone to; empty to assign to the active bone collection bpy.ops.armature.**collection_create_and_assign(*, name='')**

Create a new bone collection and assign all selected bones

PARAMETERS:

name (*string, (optional, never None)*) – Bone Collection, Name of the bone collection to create

bpy.ops.armature.**collection_deselect()**

Deselect bones of active Bone Collection

bpy.ops.armature.**collection_move(*, direction='UP')**

Change position of active Bone Collection in list of Bone collections

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Direction to move the active Bone Collection towards

bpy.ops.armature.**collection_remove()**

Remove the active bone collection

bpy.ops.armature.**collection_remove_unused()**

Remove all bone collections that have neither bones nor children. This is done recursively, so bone collections that only have unused children are al removed

FILE:

startup/bl_operators/anim.py:603

bpy.ops.armature.**collection_select()**

Select bones in active Bone Collection

bpy.ops.armature.**collection_show_all()**

Show all bone collections

FILE:

startup/bl_operators/anim.py:558

bpy.ops.armature.**collection_unassign(*, name='')**

Remove selected bones from the active bone collection

PARAMETERS:

name (*string, (optional, never None)*) – Bone Collection, Name of the bone collection to unassign this bone from; empty to unassign from tl active bone collection

bpy.ops.armature.**collection_unassign_named(*, name='', bone_name='')**

Unassign the named bone from this bone collection

PARAMETERS:

- **name** (*string, (optional, never None)*) – Bone Collection, Name of the bone collection to unassign this bone from; empty to unassign from tl active bone collection
- **bone_name** (*string, (optional, never None)*) – Bone Name, Name of the bone to unassign from the collection; empty to use the active bone

bpy.ops.armature.**collection_unsolo_all()**

Clear the 'solo' setting on all bone collections

FILE:

startup/bl_operators/anim.py:581

bpy.ops.armature.**copy_bone_color_to_selected(*, bone_type='EDIT')**

Copy the bone color of the active bone to all selected bones

PARAMETERS:

bone_type (*enum in ['EDIT', 'POSE'], (optional)*) –

Type

- `EDIT` Bone – Copy Bone colors from the active bone to all selected bones.
- `POSE` Pose Bone – Copy Pose Bone colors from the active pose bone to all selected pose bones.

FILE:

startup/bl_operators/anim.py:477

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

Remove selected bones from the armature

PARAMETERS:

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

bpy.ops.armature.**dissolve()**

Dissolve selected bones from the armature

bpy.ops.armature.**duplicate(*, do_flip_names=False)**

Make copies of the selected bones within the same armature

PARAMETERS:

do_flip_names (*boolean, (optional)*) – Flip Names, Try to flip names of the bones, if possible, instead of adding a number extension

bpy.ops.armature.**duplicate_move(*, ARMATURE_OT_duplicate=None, TRANSFORM_OT_translate=None)**

Make copies of the selected bones within the same armature and move them

PARAMETERS:

- **ARMATURE_OT_duplicate** (`ARMATURE_OT_duplicate`, (optional)) – Duplicate Selected Bone(s), Make copies of the selected bones within the same armature
- **TRANSFORM_OT_translate** (`TRANSFORM_OT_translate`, (optional)) – Move, Move selected items

bpy.ops.armature.**extrude(*, forked=False)**

Create new bones from the selected joints

PARAMETERS:

forked (*boolean, (optional)*) – Forked

bpy.ops.armature.**extrude_forked(*, ARMATURE_OT_extrude=None, TRANSFORM_OT_translate=None)**

Create new bones from the selected joints and move them

PARAMETERS:

- **ARMATURE_OT_extrude** (`ARMATURE_OT_extrude`, (optional)) – Extrude, Create new bones from the selected joints
- **TRANSFORM_OT_translate** (`TRANSFORM_OT_translate`, (optional)) – Move, Move selected items

bpy.ops.armature.**extrude_move(*, ARMATURE_OT_extrude=None, TRANSFORM_OT_translate=None)**

Create new bones from the selected joints and move them

PARAMETERS:

- **ARMATURE_OT_extrude** (`ARMATURE_OT_extrude`, (optional)) – Extrude, Create new bones from the selected joints
- **TRANSFORM_OT_translate** (`TRANSFORM_OT_translate`, (optional)) – Move, Move selected items

bpy.ops.armature.**fill()**

Add bone between selected joint(s) and/or 3D cursor bpy.ops.armature.**flip_names(*, do_strip_numbers=False)**

Flips (and corrects) the axis suffixes of the names of selected bones

PARAMETERS:

do_strip_numbers (*boolean, (optional)*) – Strip Numbers, Try to remove right-most dot-number from flipped names.Warning: May result i incoherent naming in some cases

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

Tag selected bones to not be visible in Edit Mode

PARAMETERS:

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

bpy.ops.armature.**move_to_collection(*, collection_index=-1, new_collection_name='')**

Move bones to a collection

PARAMETERS:

- **collection_index** (*int in [-1, inf], (optional)*) – Collection Index, Index of the collection to move selected bones to. When the operator shou create a new bone collection, do not include this parameter and pass new_collection_name
- **new_collection_name** (*string, (optional, never None)*) – Name, Name of a to-be-added bone collection. Only pass this if you want to create a new bone collection and move the selected bones to it. To move to an existing collection, do not include this parameter and use collection_index

bpy.ops.armature.**parent_clear(*, type='CLEAR')**

Remove the parent-child relationship between selected bones and their parents

PARAMETERS:

type (*enum in ['CLEAR', 'DISCONNECT'], (optional)*) – Clear Type, What way to clear parenting

bpy.ops.armature.**parent_set(*, type='CONNECTED')**

Set the active bone as the parent of the selected bones

PARAMETERS:

type (*enum in ['CONNECTED', 'OFFSET'], (optional)*) – Parent Type, Type of parenting

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

Reveal all bones hidden in Edit Mode

PARAMETERS:

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

bpy.ops.armature.**roll_clear(*, roll=0.0)**

Clear roll for selected bones

PARAMETERS:

roll (*float in [-6.28319, 6.28319], (optional)*) – Roll

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

Toggle selection status of all bones

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.

- DESELECT Deselect – Deselect all elements.
- INVERT Invert – Invert selection of all elements.

bpy.ops.armature.**select_hierarchy(*, direction='PARENT', extend=False)**

Select immediate parent/children of selected bones

PARAMETERS:

- **direction** (*enum in ['PARENT', 'CHILD'], (optional)*) – Direction
- **extend** (*boolean, (optional)*) – Extend, Extend the selection

bpy.ops.armature.**select_less()**

Deselect those bones at the boundary of each selection region

bpy.ops.armature.**select_linked(*, all_forks=False)**

Select all bones linked by parent/child connections to the current selection

PARAMETERS:

all_forks (*boolean, (optional)*) – All Forks, Follow forks in the parents chain

bpy.ops.armature.**select_linked_pick(*, deselect=False, all_forks=False)**

(De)select bones linked by parent/child connections under the mouse cursor

PARAMETERS:

- **deselect** (*boolean, (optional)*) – Deselect
- **all_forks** (*boolean, (optional)*) – All Forks, Follow forks in the parents chain

bpy.ops.armature.**select_mirror(*, only_active=False, extend=False)**

Mirror the bone selection

PARAMETERS:

- **only_active** (*boolean, (optional)*) – Active Only, Only operate on the active bone
- **extend** (*boolean, (optional)*) – Extend, Extend the selection

bpy.ops.armature.**select_more()**

Select those bones connected to the initial selection

bpy.ops.armature.**select_similar(*, type='LENGTH', threshold=0.1)**

Select similar bones by property types

PARAMETERS:

- **type** (*enum in ['CHILDREN', 'CHILDREN_IMMEDIATE', 'SIBLINGS', 'LENGTH', 'DIRECTION', 'PREFIX', 'SUFFIX', 'BONE_COLLECTION', 'COLOR', 'SHAPE'], (optional)*) – Type
- **threshold** (*float in [0, 1], (optional)*) – Threshold

bpy.ops.armature.**separate()**

Isolate selected bones into a separate armature

bpy.ops.armature.**shortest_path_pick()**

Select shortest path between two bones

bpy.ops.armature.**split()**

Split off selected bones from connected unselected bones

bpy.ops.armature.**subdivide(*, number_cuts=1)**

Break selected bones into chains of smaller bones

PARAMETERS:

PARAMETERS:

number_cuts (*int in [1, 1000], (optional)*) – Number of Cuts

bpy.ops.armature.**switch_direction()**

Change the direction that a chain of bones points in (head and tail swap)

bpy.ops.armature.**symmetrize(*, direction='NEGATIVE_X')**

Enforce symmetry, make copies of the selection or use existing

PARAMETERS:

direction (*enum in ['NEGATIVE_X', 'POSITIVE_X'], (optional)*) – Direction, Which sides to copy from and to (when both are selected)