

Constraint Operators

bpy.ops.constraint.add_target()

Add a target to the constraint

FILE:

[startup/bl_operators/constraint.py:26](#)

bpy.ops.constraint.apply(*, constraint="", owner='OBJECT', report=False)

Apply constraint and remove from the stack

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
 - **OBJECT** Object – Edit a constraint on the active object.
 - **BONE** Bone – Edit a constraint on the active bone.
- **report** (*boolean, (optional)*) – Report, Create a notification after the operation

bpy.ops.constraint.childof_clear_inverse(*, constraint="", owner='OBJECT')

Clear inverse correction for Child Of constraint

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
 - **OBJECT** Object – Edit a constraint on the active object.
 - **BONE** Bone – Edit a constraint on the active bone.

bpy.ops.constraint.childof_set_inverse(*, constraint="", owner='OBJECT')

Set inverse correction for Child Of constraint

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
 - **OBJECT** Object – Edit a constraint on the active object.
 - **BONE** Bone – Edit a constraint on the active bone.

bpy.ops.constraint.copy(*, constraint="", owner='OBJECT', report=False)

Duplicate constraint at the same position in the stack

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
 - **OBJECT** Object – Edit a constraint on the active object.
 - **BONE** Bone – Edit a constraint on the active bone.
- **report** (*boolean, (optional)*) – Report, Create a notification after the operation

bpy.ops.constraint.copy_to_selected(*, constraint="", owner='OBJECT')

Copy constraint to other selected objects/bones

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
 - OBJECT Object – Edit a constraint on the active object.
 - BONE Bone – Edit a constraint on the active bone.

bpy.ops.constraint.delete(*, constraint="", owner='OBJECT', report=False)

Remove constraint from constraint stack

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
 - OBJECT Object – Edit a constraint on the active object.
 - BONE Bone – Edit a constraint on the active bone.
- **report** (*boolean, (optional)*) – Report, Create a notification after the operation

bpy.ops.constraint.disable_keep_transform()

Set the influence of this constraint to zero while trying to maintain the object's transformation. Other active constraints can still influence the final transformation

FILE:

[startup/bl_operators/constraint.py:86](#)

bpy.ops.constraint.followpath_path_animate(*, constraint="", owner='OBJECT', frame_start=1, length=100)

Add default animation for path used by constraint if it isn't animated already

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
 - OBJECT Object – Edit a constraint on the active object.
 - BONE Bone – Edit a constraint on the active bone.
- **frame_start** (*int in [-1048574, 1048574], (optional)*) – Start Frame, First frame of path animation
- **length** (*int in [0, 1048574], (optional)*) – Length, Number of frames that path animation should take

bpy.ops.constraint.limitdistance_reset(*, constraint="", owner='OBJECT')

Reset limiting distance for Limit Distance Constraint

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
 - OBJECT Object – Edit a constraint on the active object.
 - BONE Bone – Edit a constraint on the active bone.

bpy.ops.constraint.move_down(*, constraint="", owner='OBJECT')

Move constraint down in constraint stack

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) –
Owner, The owner of this constraint
 - **OBJECT** Object – Edit a constraint on the active object.
 - **BONE** Bone – Edit a constraint on the active bone.

`bpy.ops.constraint.move_to_index(*, constraint="", owner='OBJECT', index=0)`

Change the constraint's position in the list so it evaluates after the set number of others

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) –
Owner, The owner of this constraint
 - **OBJECT** Object – Edit a constraint on the active object.
 - **BONE** Bone – Edit a constraint on the active bone.
- **index** (*int in [0, inf], (optional)*) – Index, The index to move the constraint to

`bpy.ops.constraint.move_up(*, constraint="", owner='OBJECT')`

Move constraint up in constraint stack

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) –
Owner, The owner of this constraint
 - **OBJECT** Object – Edit a constraint on the active object.
 - **BONE** Bone – Edit a constraint on the active bone.

`bpy.ops.constraint.normalize_target_weights()`

Normalize weights of all target bones

FILE:

[startup/bl_operators/constraint.py:61](#)

`bpy.ops.constraint.objectsolver_clear_inverse(*, constraint="", owner='OBJECT')`

Clear inverse correction for Object Solver constraint

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) –
Owner, The owner of this constraint
 - **OBJECT** Object – Edit a constraint on the active object.
 - **BONE** Bone – Edit a constraint on the active bone.

`bpy.ops.constraint.objectsolver_set_inverse(*, constraint="", owner='OBJECT')`

Set inverse correction for Object Solver constraint

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) –
Owner, The owner of this constraint

- **OBJECT** Object – Edit a constraint on the active object.
- **BONE** Bone – Edit a constraint on the active bone.

`bpy.ops.constraint.remove_target(*, index=0)`

Remove the target from the constraint

PARAMETERS:

index (*int in [-inf, inf], (optional)*) – index

FILE:

[startup/bl_operators/constraint.py:44](#)

`bpy.ops.constraint.stretchto_reset(*, constraint='', owner='OBJECT')`

Reset original length of bone for Stretch To Constraint

PARAMETERS:

- **constraint** (*string, (optional, never None)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) –
Owner, The owner of this constraint
 - **OBJECT** Object – Edit a constraint on the active object.
 - **BONE** Bone – Edit a constraint on the active bone.