Object(ID)

Basic Object Operations Example

This script demonstrates basic operations on object like creating new object, placing it into a view layer, selecting it and making it active.

```
import bpy

view_layer = bpy.context.view_layer

# Create new light datablock.
light_data = bpy.data.lights.new(name="New Light", type='POINT')

# Create new object with our light datablock.
light_object = bpy.data.objects.new(name="New Light", object_data=light_data)

# Link light object to the active collection of current view layer,
# so that it'll appear in the current scene.
view_layer.active_layer_collection.collection.objects.link(light_object)

# Place light to a specified location.
light_object.location = (5.0, 5.0, 5.0)

# And finally select it and make it active.
light_object.select_set(True)
view_layer.objects.active = light_object
```

base classes — `bpy_struct`, `ID`

class bpy.types.**Object(ID)**

Object data-block defining an object in a scene

active_material

Active material being displayed

TYPE:

`Material`

active_material_index

Index of active material slot

TYPE:

int in [0, inf], default 0

active_selection_set

Index of the currently active selection set

TYPE:

int in [-inf, inf], default 0

active_shape_key

Current shape key

TYPE:

ShapeKey , (readonly)

active_shape_key_index

Current shape key index

TYPE:

int in [-32768, 32767], default 0

add_rest_position_attribute

Add a “rest_position” attribute that is a copy of the position attribute before shape keys and modifiers are evaluated

TYPE:

boolean, default False

animation_data

Animation data for this data-block

TYPE:

AnimData , (readonly)

animation_visualization

Animation data for this data-block

TYPE:

AnimViz , (readonly, never None)

bound_box

Object’s bounding box in object-space coordinates, all values are -1.0 when not available

TYPE:

float multi-dimensional array of 8 * 3 items in [-inf, inf], default ((0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0), (0.0, 0.0, 0.0)), (readonly)

collision

Settings for using the object as a collider in physics simulation

TYPE:

CollisionSettings , (readonly)

color

Object color and alpha, used when the Object Color mode is enabled

TYPE:

float array of 4 items in [0, inf], default (1.0, 1.0, 1.0, 1.0)

constraints

Constraints affecting the transformation of the object

TYPE:

ObjectConstraints bpy_prop_collection of Constraint , (readonly)

cycles

Cycles object settings

TYPE:

CyclesObjectSettings , (readonly)

data

Object data

TYPE:

ID

delta_location

Extra translation added to the location of the object

TYPE:

mathutils.Vector of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

delta_rotation_euler

Extra rotation added to the rotation of the object (when using Euler rotations)

TYPE:

mathutils.Euler rotation of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

delta_rotation_quaternion

Extra rotation added to the rotation of the object (when using Quaternion rotations)

TYPE:

mathutils.Quaternion rotation of 4 items in [-inf, inf], default (1.0, 0.0, 0.0, 0.0)

delta_scale

Extra scaling added to the scale of the object

TYPE:

mathutils.Vector of 3 items in [-inf, inf], default (1.0, 1.0, 1.0)

dimensions

Absolute bounding box dimensions of the object. Warning: Assigning to it or its members multiple consecutive times will not work correctly, a this needs up-to-date evaluated data

TYPE:

mathutils.Vector of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

display

Object display settings for 3D viewport

TYPE:

ObjectDisplay, (readonly, never None)

display_bounds_type

Object boundary display type

- BOX Box – Display bounds as box.
- SPHERE Sphere – Display bounds as sphere.
- CYLINDER Cylinder – Display bounds as cylinder.
- CONE Cone – Display bounds as cone.
- CAPSULE Capsule – Display bounds as capsule.

TYPE:

enum in ['BOX', 'SPHERE', 'CYLINDER', 'CONE', 'CAPSULE'], default 'BOX'

display_type

How to display object in viewport

- BOUNDS Bounds – Display the bounds of the object.

- `WIRE` Wire – Display the object as a wireframe.
- `SOLID` Solid – Display the object as a solid (if solid drawing is enabled in the viewport).
- `TEXTURED` Textured – Display the object with textures (if textures are enabled in the viewport).

TYPE:

enum in ['BOUNDS', 'WIRE', 'SOLID', 'TEXTURED'], default 'TEXTURED'

empty_display_size

Size of display for empties in the viewport

TYPE:

float in [0.0001, 1000], default 1.0

empty_display_type

Viewport display style for empties

TYPE:

enum in Object Empty Drawtype Items, default 'PLAIN_AXES'

empty_image_depth

Determine which other objects will occlude the image

TYPE:

enum in ['DEFAULT', 'FRONT', 'BACK'], default 'DEFAULT'

empty_image_offset

Origin offset distance

TYPE:

float array of 2 items in [-inf, inf], default (-0.5, -0.5)

empty_image_side

Show front/back side

TYPE:

enum in ['DOUBLE_SIDED', 'FRONT', 'BACK'], default 'DOUBLE_SIDED'

field

Settings for using the object as a field in physics simulation

TYPE:

`FieldSettings`, (readonly)

hide_probe_plane

Globally disable in planar light probes

TYPE:

boolean, default False

hide_probe_sphere

Globally disable in spherical light probes

TYPE:

boolean, default False

hide_probe_volume

Globally disable in volume probes

TYPE:

boolean, default False

hide_render

Globally disable in renders

TYPE:

boolean, default False

hide_select

Disable selection in viewport

TYPE:

boolean, default False

hide_viewport

Globally disable in viewports

TYPE:

boolean, default False

image_user

Parameters defining which layer, pass and frame of the image is displayed

TYPE:

`ImageUser` , (readonly, never None)

instance_collection

Instance an existing collection

TYPE:

`Collection`

instance_faces_scale

Scale the face instance objects

TYPE:

float in [0.001, 10000], default 1.0

instance_type

If not None, object instancing method to use

- `NONE` None.
- `VERTS` Vertices – Instantiate child objects on all vertices.
- `FACES` Faces – Instantiate child objects on all faces.
- `COLLECTION` Collection – Enable collection instancing.

TYPE:

enum in ['NONE', 'VERTS', 'FACES', 'COLLECTION'], default 'NONE'

is_from_instancer

Object comes from a instancer

TYPE:

boolean, default False, (readonly)

is_from_set

Object comes from a background set

TYPE:

boolean, default False, (readonly)

is_holdout

Render objects as a holdout or matte, creating a hole in the image with zero alpha, to fill out in compositing with real footage or another rende

TYPE:

boolean, default False

is_instancer

TYPE:

boolean, default False, (readonly)

is_shadow_catcher

Only render shadows and reflections on this object, for compositing renders into real footage. Objects with this setting are considered to alrea exist in the footage, objects without it are synthetic objects being composited into it.

TYPE:

boolean, default False

light_linking

Light linking settings

TYPE:

`ObjectLightLinking` , (readonly, never None)

lightgroup

Lightgroup that the object belongs to

TYPE:

string, default “”, (never None)

lineart

Line Art settings for the object

TYPE:

`ObjectLineArt` , (readonly)

location

Location of the object

TYPE:

`mathutils.Vector` of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

lock_location

Lock editing of location when transforming

TYPE:

boolean array of 3 items, default (False, False, False)

lock_rotation

Lock editing of rotation when transforming

TYPE:

boolean array of 3 items, default (False, False, False)

lock_rotation_w

Lock editing of 'angle' component of four-component rotations when transforming

TYPE:

boolean, default False

lock_rotations_4d

Lock editing of four component rotations by components (instead of as Eulers)

TYPE:

boolean, default True

lock_scale

Lock editing of scale when transforming

TYPE:

boolean array of 3 items, default (False, False, False)

material_slots

Material slots in the object

TYPE:

`bpy_prop_collection` of `MaterialSlot`, (readonly)

matrix_basis

Matrix access to location, rotation and scale (including deltas), before constraints and parenting are applied

TYPE:

`mathutils.Matrix` of 4 * 4 items in [-inf, inf], default ((0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0
0.0))

matrix_local

Parent relative transformation matrix. Warning: Only takes into account object parenting, so e.g. in case of bone parenting you get a matrix relative to the Armature object, not to the actual parent bone

TYPE:

`mathutils.Matrix` of 4 * 4 items in [-inf, inf], default ((0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0
0.0))

matrix_parent_inverse

Inverse of object's parent matrix at time of parenting

TYPE:

`mathutils.Matrix` of 4 * 4 items in [-inf, inf], default ((1.0, 0.0, 0.0, 0.0), (0.0, 1.0, 0.0, 0.0), (0.0, 0.0, 1.0, 0.0), (0.0, 0.0, 0
1.0))

matrix_world

Worldspace transformation matrix

TYPE:

`mathutils.Matrix` of 4 * 4 items in [-inf, inf], default ((0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0
0.0))

mode

Object interaction mode

TYPE:

enum in Object Mode Items, default 'OBJECT', (readonly)

modifiers

Modifiers affecting the geometric data of the object

TYPE:

ObjectModifiers bpy_prop_collection of Modifier, (readonly)

motion_path

Motion Path for this element

TYPE:

MotionPath, (readonly)

parent

Parent object

TYPE:

Object

parent_bone

Name of parent bone in case of a bone parenting relation

TYPE:

string, default "", (never None)

parent_type

Type of parent relation

- OBJECT Object – The object is parented to an object.
- ARMATURE Armature.
- LATTICE Lattice – The object is parented to a lattice.
- VERTEX Vertex – The object is parented to a vertex.
- VERTEX_3 3 Vertices.
- BONE Bone – The object is parented to a bone.

TYPE:

enum in ['OBJECT', 'ARMATURE', 'LATTICE', 'VERTEX', 'VERTEX_3', 'BONE'], default 'OBJECT'

parent_vertices

Indices of vertices in case of a vertex parenting relation

TYPE:

int array of 3 items in [0, inf], default (0, 0, 0)

particle_systems

Particle systems emitted from the object

TYPE:

ParticleSystems bpy_prop_collection of ParticleSystem, (readonly)

pass_index

Index number for the "Object Index" render pass

TYPE:

int in [0, 32767], default 0

pose

Current pose for armatures

TYPE:

Pose, (readonly)

rigid_body

Settings for rigid body simulation

TYPE:

RigidBodyObject, (readonly)

rigid_body_constraint

Constraint constraining rigid bodies

TYPE:

RigidBodyConstraint, (readonly)

rotation_axis_angle

Angle of Rotation for Axis-Angle rotation representation

TYPE:

float array of 4 items in [-inf, inf], default (0.0, 0.0, 1.0, 0.0)

rotation_euler

Rotation in Eulers

TYPE:

mathutils.Euler rotation of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

rotation_mode

TYPE:

enum in Object Rotation Mode Items, default 'XYZ'

rotation_quaternion

Rotation in Quaternions

TYPE:

mathutils.Quaternion rotation of 4 items in [-inf, inf], default (1.0, 0.0, 0.0, 0.0)

scale

Scaling of the object

TYPE:

mathutils.Vector of 3 items in [-inf, inf], default (1.0, 1.0, 1.0)

selection_sets

List of groups of bones for easy selection

TYPE:

bpy_prop_collection of SelectionSet, (readonly)

shader_effects

Effects affecting display of object

TYPE:

ObjectShaderFx bpy_prop_collection of ShaderFx, (readonly)

show_all_edges

Display all edges for mesh objects

TYPE:
boolean, default False

show_axis

Display the object's origin and axes

TYPE:
boolean, default False

show_bounds

Display the object's bounds

TYPE:
boolean, default False

show_empty_image_only_axis_aligned

Only display the image when it is aligned with the view axis

TYPE:
boolean, default False

show_empty_image_orthographic

Display image in orthographic mode

TYPE:
boolean, default True

show_empty_image_perspective

Display image in perspective mode

TYPE:
boolean, default True

show_in_front

Make the object display in front of others

TYPE:
boolean, default False

show_instancer_for_render

Make instancer visible when rendering

TYPE:
boolean, default True

show_instancer_for_viewport

Make instancer visible in the viewport

TYPE:
boolean, default True

show_name

Display the object's name

TYPE:
boolean, default False

show_only_shape_key

Only show the active shape key at full value

TYPE:

boolean, default False

show_texture_space

Display the object's texture space

TYPE:

boolean, default False

show_transparent

Display material transparency in the object

TYPE:

boolean, default False

show_wire

Display the object's wireframe over solid shading

TYPE:

boolean, default False

soft_body

Settings for soft body simulation

TYPE:

`SoftBodySettings`, (readonly)

track_axis

Axis that points in the 'forward' direction (applies to Instance Vertices when Align to Vertex Normal is enabled)

TYPE:

enum in Object Axis Items, default 'POS_X'

type

Type of object

TYPE:

enum in Object Type Items, default 'EMPTY', (readonly)

up_axis

Axis that points in the upward direction (applies to Instance Vertices when Align to Vertex Normal is enabled)

TYPE:

enum in ['X', 'Y', 'Z'], default 'X'

use_camera_lock_parent

View Lock 3D viewport camera transformation affects the object's parent instead

TYPE:

boolean, default False

use_dynamic_topology_sculpting

TYPE:

boolean, default False, (readonly)

use_empty_image_alpha

Use alpha blending instead of alpha test (can produce sorting artifacts)

TYPE:

boolean, default False

use_grease_pencil_lights

Lights affect Grease Pencil object

TYPE:

boolean, default True

use_instance_faces_scale

Scale instance based on face size

TYPE:

boolean, default False

use_instance_vertices_rotation

Rotate instance according to vertex normal

TYPE:

boolean, default False

use_mesh_mirror_x

Enable mesh symmetry in the X axis

TYPE:

boolean, default False

use_mesh_mirror_y

Enable mesh symmetry in the Y axis

TYPE:

boolean, default False

use_mesh_mirror_z

Enable mesh symmetry in the Z axis

TYPE:

boolean, default False

use_shape_key_edit_mode

Display shape keys in edit mode (for meshes only)

TYPE:

boolean, default False

use_simulation_cache

Cache frames during simulation nodes playback

TYPE:

boolean, default True

vertex_groups

Vertex groups of the object

TYPE:

`VertexGroups bpy_prop_collection` of `VertexGroup`, (readonly)

visible_camera

Object visibility to camera rays

TYPE:

boolean, default True

visible_diffuse

Object visibility to diffuse rays

TYPE:

boolean, default True

visible_glossy

Object visibility to glossy rays

TYPE:

boolean, default True

visible_shadow

Object visibility to shadow rays

TYPE:

boolean, default True

visible_transmission

Object visibility to transmission rays

TYPE:

boolean, default True

visible_volume_scatter

Object visibility to volume scattering rays

TYPE:

boolean, default True

children

All the children of this object.

TYPE:

tuple of `Object`

Note

Takes `O(len(bpy.data.objects))` time.

(readonly)

children_recursive

A list of all children from this object.

TYPE:

tuple of `Object`

Note

Takes `O(len(bpy.data.objects))` time.

(readonly)

users_collection

The collections this object is in.

TYPE:

tuple of `Collection`

Note

Takes `O(len(bpy.data.collections) + len(bpy.data.scenes))` time.

(readonly)

users_scene

The scenes this object is in.

TYPE:

tuple of `Scene`

Note

Takes `O(len(bpy.data.scenes) * len(bpy.data.objects))` time.

(readonly)

select_get(*, view_layer=None)

Test if the object is selected. The selection state is per view layer.

PARAMETERS:

view_layer (`ViewLayer`, (optional)) – Use this instead of the active view layer

RETURNS:

Object selected

RETURN TYPE:

boolean

select_set(state, *, view_layer=None)

Select or deselect the object. The selection state is per view layer.

PARAMETERS:

- **state** (*boolean*) – Selection state to define
- **view_layer** (`ViewLayer`, (optional)) – Use this instead of the active view layer

hide_get(*, view_layer=None)

Test if the object is hidden for viewport editing. This hiding state is per view layer.

PARAMETERS:

view_layer (`ViewLayer`, (optional)) – Use this instead of the active view layer

RETURNS:

Object hidden

RETURN TYPE:

boolean

hide_set(state, *, view_layer=None)

Hide the object for viewport editing. This hiding state is per view layer.

PARAMETERS:

- **state** (*boolean*) – Hide state to define
- **view_layer** (`ViewLayer` , (optional)) – Use this instead of the active view layer

visible_get(*, view_layer=None, viewport=None)

Test if the object is visible in the 3D viewport, taking into account all visibility settings

PARAMETERS:

- **view_layer** (`ViewLayer` , (optional)) – Use this instead of the active view layer
- **viewport** (`SpaceView3D` , (optional)) – Use this instead of the active 3D viewport

RETURNS:

Object visible

RETURN TYPE:

boolean

holdout_get(*, view_layer=None)

Test if object is masked in the view layer

PARAMETERS:

view_layer (`ViewLayer` , (optional)) – Use this instead of the active view layer

RETURNS:

Object holdout

RETURN TYPE:

boolean

indirect_only_get(*, view_layer=None)

Test if object is set to contribute only indirectly (through shadows and reflections) in the view layer

PARAMETERS:

view_layer (`ViewLayer` , (optional)) – Use this instead of the active view layer

RETURNS:

Object indirect only

RETURN TYPE:

boolean

local_view_get(viewport)

Get the local view state for this object

PARAMETERS:

viewport (`SpaceView3D`) – Viewport in local view

RETURNS:

Object local view state

RETURN TYPE:

boolean

local_view_set(viewport, state)

Set the local view state for this object

PARAMETERS:

- **viewport** (`SpaceView3D`) – Viewport in local view

- **state** (*boolean*) – Local view state to define

visible_in_viewport_get(viewport)

Check for local view and local collections for this viewport and object

PARAMETERS:

viewport (`SpaceView3D`) – Viewport in local collections

RETURNS:

Object viewport visibility

RETURN TYPE:

boolean

convert_space(*, pose_bone=None, matrix=((0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0), (0.0, 0.0, 0.0, 0.0)), from_space='WORLD', to_space='WORLD')

Convert (transform) the given matrix from one space to another

PARAMETERS:

- **pose_bone** (`PoseBone` , (optional)) – Bone to use to define spaces (may be None, in which case only the two 'WORLD' and 'LOCAL' spaces are usable)
- **matrix** (`mathutils.Matrix` of 4 * 4 items in [-inf, inf], (optional)) – The matrix to transform
- **from_space** (*enum in ['WORLD', 'POSE', 'LOCAL_WITH_PARENT', 'LOCAL'], (optional)*) –
  The space in which 'matrix' is currently
  - `WORLD` World Space – The most global space in Blender.
  - `POSE` Pose Space – The pose space of a bone (its armature's object space).
  - `LOCAL_WITH_PARENT` Local With Parent – The rest pose local space of a bone (this matrix includes parent transforms).
  - `LOCAL` Local Space – The local space of an object/bone.
- **to_space** (*enum in ['WORLD', 'POSE', 'LOCAL_WITH_PARENT', 'LOCAL'], (optional)*) –
  The space to which you want to transform 'matrix'
  - `WORLD` World Space – The most global space in Blender.
  - `POSE` Pose Space – The pose space of a bone (its armature's object space).
  - `LOCAL_WITH_PARENT` Local With Parent – The rest pose local space of a bone (this matrix includes parent transforms).
  - `LOCAL` Local Space – The local space of an object/bone.

RETURNS:

The transformed matrix

RETURN TYPE:

`mathutils.Matrix` of 4 * 4 items in [-inf, inf]

calc_matrix_camera(depsgraph, *, x=1, y=1, scale_x=1.0, scale_y=1.0)

Generate the camera projection matrix of this object (mostly useful for Camera and Light types)

PARAMETERS:

- **depsgraph** (`Depsgraph`) – Depsgraph to get evaluated data from
- **x** (*int in [0, inf], (optional)*) – Width of the render area
- **y** (*int in [0, inf], (optional)*) – Height of the render area
- **scale_x** (*float in [1e-06, inf], (optional)*) – Width scaling factor
- **scale_y** (*float in [1e-06, inf], (optional)*) – Height scaling factor

RETURNS:

The camera projection matrix

RETURN TYPE:

`mathutils.Matrix` of 4 * 4 items in [-inf, inf]

mathutils.Matrix of 4 * 4 items in [-inf, inf]

camera_fit_coords(depsgraph, coordinates)

Compute the coordinate (and scale for ortho cameras) given object should be to 'see' all given coordinates

PARAMETERS:

- **depsgraph** (Depsgraph) – Depsgraph to get evaluated data from
- **coordinates** (*float array of 1 items in [-inf, inf], (never None)*) – Coordinates to fit in

RETURNS:

co_return, The location to aim to be able to see all given points, mathutils.Vector of 3 items in [-inf, inf]

scale_return, The ortho scale to aim to be able to see all given points (if relevant), float in [-inf, inf]

RETURN TYPE:

(mathutils.Vector of 3 items in [-inf, inf], float in [-inf, inf])

crazyspace_eval(depsgraph, scene)

Compute orientation mapping between vertices of an original object and object with shape keys and deforming modifiers applied.The evaluati is to be freed with the crazyspace_eval_free function

PARAMETERS:

- **depsgraph** (Depsgraph) – Dependency Graph, Evaluated dependency graph
- **scene** (Scene) – Scene, Scene of the object

crazyspace_displacement_to_deformed(*, vertex_index=0, displacement=(0.0, 0.0, 0.0))

Convert displacement vector from non-deformed object space to deformed object space

PARAMETERS:

- **vertex_index** (*int in [-inf, inf], (optional)*) – vertex_index
- **displacement** (mathutils.Vector of 3 items in [-inf, inf], (optional)) – displacement

RETURNS:

displacement_deformed

RETURN TYPE:

mathutils.Vector of 3 items in [-inf, inf]

crazyspace_displacement_to_original(*, vertex_index=0, displacement=(0.0, 0.0, 0.0))

Free evaluated state of crazyspace

PARAMETERS:

- **vertex_index** (*int in [-inf, inf], (optional)*) – vertex_index
- **displacement** (mathutils.Vector of 3 items in [-inf, inf], (optional)) – displacement

RETURNS:

displacement_original

RETURN TYPE:

mathutils.Vector of 3 items in [-inf, inf]

crazyspace_eval_clear()

crazyspace_eval_clear

to_mesh(*, preserve_all_data_layers=False, depsgraph=None)

Create a Mesh data-block from the current state of the object. The object owns the data-block. To force free it use to_mesh_clear(). The res is temporary and cannot be used by objects from the main database.

PARAMETERS:

- **preserve_all_data_layers** (*boolean, (optional)*) – Preserve all data layers in the mesh, like UV maps and vertex groups. By default Blender only computes the subset of data layers needed for viewport display and rendering, for better performance.
- **depsgraph** (`Depsgraph` , (optional)) – Dependency Graph, Evaluated dependency graph which is required when preserve_all_data_layers is true

RETURNS:

Mesh created from object

RETURN TYPE:

`Mesh`

to_mesh_clear()

Clears mesh data-block created by to_mesh()

to_curve(depsgraph, *, apply_modifiers=False)

Create a Curve data-block from the current state of the object. This only works for curve and text objects. The object owns the data-block. ´ force free it, use to_curve_clear(). The result is temporary and cannot be used by objects from the main database.

PARAMETERS:

- **depsgraph** (`Depsgraph`) – Dependency Graph, Evaluated dependency graph
- **apply_modifiers** (*boolean, (optional)*) – Apply the deform modifiers on the control points of the curve. This is only supported for curv objects.

RETURNS:

Curve created from object

RETURN TYPE:

`Curve`

to_curve_clear()

Clears curve data-block created by to_curve()

find_armature()

Find armature influencing this object as a parent or via a modifier

RETURNS:

Armature object influencing this object or nullptr

RETURN TYPE:

`Object`

shape_key_add(*, name='Key', from_mix=True)

Add shape key to this object

PARAMETERS:

- **name** (*string, (optional, never None)*) – Unique name for the new keyblock
- **from_mix** (*boolean, (optional)*) – Create new shape from existing mix of shapes

RETURNS:

New shape keyblock

RETURN TYPE:

`ShapeKey`

shape_key_remove(key)

Remove a Shape Key from this object

PARAMETERS:

key (`ShapeKey` , (never None)) – Keyblock to be removed

shape_key_clear()

Remove all Shape Keys from this object

ray_cast(origin, direction, *, distance=1.70141e+38, depsgraph=None)

Cast a ray onto evaluated geometry, in object space (using context's or provided depsgraph to get evaluated mesh if needed)

PARAMETERS:

- **origin** (`mathutils.Vector` of 3 items in [-inf, inf]) – Origin of the ray, in object space
- **direction** (`mathutils.Vector` of 3 items in [-inf, inf]) – Direction of the ray, in object space
- **distance** (*float in [0, inf], (optional)*) – Maximum distance
- **depsgraph** (`Depsgraph` , (optional)) – Depsgraph to use to get evaluated data, when called from original object (only needed if current Context's depsgraph is not suitable)

RETURNS:

result, Whether the ray successfully hit the geometry, boolean

location, The hit location of this ray cast, `mathutils.Vector` of 3 items in [-inf, inf]

normal, The face normal at the ray cast hit location, `mathutils.Vector` of 3 items in [-inf, inf]

index, The face index, -1 when original data isn't available, int in [-inf, inf]

RETURN TYPE:

(boolean, `mathutils.Vector` of 3 items in [-inf, inf], `mathutils.Vector` of 3 items in [-inf, inf], int in [-inf, inf])

closest_point_on_mesh(origin, *, distance=1.84467e+19, depsgraph=None)

Find the nearest point on evaluated geometry, in object space (using context's or provided depsgraph to get evaluated mesh if needed)

PARAMETERS:

- **origin** (`mathutils.Vector` of 3 items in [-inf, inf]) – Point to find closest geometry from (in object space)
- **distance** (*float in [0, inf], (optional)*) – Maximum distance
- **depsgraph** (`Depsgraph` , (optional)) – Depsgraph to use to get evaluated data, when called from original object (only needed if current Context's depsgraph is not suitable)

RETURNS:

result, Whether closest point on geometry was found, boolean

location, The location on the object closest to the point, `mathutils.Vector` of 3 items in [-inf, inf]

normal, The face normal at the closest point, `mathutils.Vector` of 3 items in [-inf, inf]

index, The face index, -1 when original data isn't available, int in [-inf, inf]

RETURN TYPE:

(boolean, `mathutils.Vector` of 3 items in [-inf, inf], `mathutils.Vector` of 3 items in [-inf, inf], int in [-inf, inf])

is_modified(scene, settings)

Determine if this object is modified from the base mesh data

PARAMETERS:

- **scene** (`Scene` , (never None)) – Scene in which to check the object
- **settings** (*enum in ['PREVIEW', 'RENDER']*) –

  Modifier settings to apply

  - `PREVIEW` Preview – Apply modifier preview settings.
  - `RENDER` Render – Apply modifier render settings.

RETURNS:

Whether the object is modified

RETURN TYPE:

boolean

is_deform_modified(scene, settings)

Determine if this object is modified by a deformation from the base mesh data

PARAMETERS:

- **scene** (`Scene`, (never None)) – Scene in which to check the object
- **settings** (*enum in ['PREVIEW', 'RENDER']*) –

  Modifier settings to apply

  - `PREVIEW` Preview – Apply modifier preview settings.
  - `RENDER` Render – Apply modifier render settings.

RETURNS:

Whether the object is deform-modified

RETURN TYPE:

boolean

update_from_editmode()

Load the objects edit-mode data into the object data

RETURNS:

Success

RETURN TYPE:

boolean

cache_release()

Release memory used by caches associated with this object. Intended to be used by render engines only.

classmethod bl_rna_get_subclass(id, default=None)

PARAMETERS:

id (*str*) – The RNA type identifier.

RETURNS:

The RNA type or default when not found.

RETURN TYPE:

`bpy.types.Struct` subclass

classmethod bl_rna_get_subclass_py(id, default=None)

PARAMETERS:

id (*str*) – The RNA type identifier.

RETURNS:

The class or default when not found.

RETURN TYPE:

type

Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.name_full`
- `ID.id type`
- `ID.is_missing`
- `ID.is_runtime_data`
- `ID.is_editable`

- ID.session_uid
- ID.is_evaluated
- ID.original
- ID.users
- ID.use_fake_user
- ID.use_extra_user
- ID.is_embedded_data
- ID.tag
- ID.is_library_indirect
- ID.library
- ID.library_weak_reference
- ID.asset_data
- ID.override_library
- ID.preview

Inherited Functions

- bpy_struct.as_pointer
- bpy_struct.driver_add
- bpy_struct.driver_remove
- bpy_struct.get
- bpy_struct.id_properties_clear
- bpy_struct.id_properties_ensure
- bpy_struct.id_properties_ui
- bpy_struct.is_property_hidden
- bpy_struct.is_property_overridable_library
- bpy_struct.is_property_readonly
- bpy_struct.is_property_set
- bpy_struct.items
- bpy_struct.keyframe_delete
- bpy_struct.keyframe_insert
- bpy_struct.keys
- bpy_struct.path_from_id
- bpy_struct.path_resolve
- bpy_struct.pop
- bpy_struct.property_overridable_library_set
- bpy_struct.property_unset
- bpy_struct.type_recast
- bpy_struct.values
- ID.rename
- ID.evaluated_get
- ID.copy
- ID.asset_mark
- ID.asset_clear
- ID.asset_generate_preview
- ID.override_create
- ID.override_hierarchy_create
- ID.user_clear
- ID.user_remap
- ID.make_local
- ID.user_of_id
- ID.animation_data_create
- ID.animation_data_clear
- ID.update_tag
- ID.preview_ensure
- ID.bl_rna_get_subclass
- ID.bl_rna_get_subclass_py

References

- bpy.context.active_object
- bpy.context.edit_object
- bpy.context.editable_objects
- bpy.context.image_paint_object
- bpy.context.object
- bpy.context.objects_in_mode
- bpy.context.objects_in_mode_unique_data
- bpy.context.particle_edit_object
- bpy.context.pose_object
- bpy.context.sculpt_object
- bpy.context.selectable_objects
- bpy.context.selected_editable_objects
- bpy.context.selected_objects
- bpy.context.vertex_paint_object
- KinematicConstraint.pole_target
- KinematicConstraint.target
- LatticeModifier.object
- LayerObjects.active
- LayerObjects.selected
- LimitDistanceConstraint.target
- LineStyleAlphaModifier_DistanceFromObj
- LineStyleColorModifier_DistanceFromObj
- LineStyleThicknessModifier_DistanceFro
- LockedTrackConstraint.target
- MaskModifier.armature
- MeshDeformModifier.object
- MeshToVolumeModifier.object
- MirrorModifier.mirror_object

- bpy.context.visible_objects
- bpy.context.weight_paint_object
- Action.flip_with_pose
- ActionConstraint.target
- ArmatureModifier.object
- ArrayModifier.curve
- ArrayModifier.end_cap
- ArrayModifier.offset_object
- ArrayModifier.start_cap
- BlendData.objects
- BlendDataMeshes.new_from_object
- BlendDataObjects.new
- BlendDataObjects.remove
- BoidRuleAvoid.object
- BoidRuleFollowLeader.object
- BoidRuleGoal.object
- BooleanModifier.object
- CameraDOFSettings.focus_object
- CastModifier.object
- ChildOfConstraint.target
- ClampToConstraint.target
- Collection.all_objects
- Collection.objects
- CollectionObjects.link
- CollectionObjects.unlink
- Constraint.space_object
- ConstraintTarget.target
- ConstraintTargetBone.target
- CopyLocationConstraint.target
- CopyRotationConstraint.target
- CopyScaleConstraint.target
- CopyTransformsConstraint.target
- Curve.bevel_object
- Curve.taper_object
- CurveModifier.object
- Curves.surface
- CyclesRenderSettings.dicing_camera
- DampedTrackConstraint.target
- DataTransferModifier.object
- Depsgraph.objects
- DepsgraphObjectInstance.instance_object
- DepsgraphObjectInstance.object
- DepsgraphObjectInstance.parent
- DisplaceModifier.texture_coords_object
- DynamicPaintSurface.output_exists
- FieldSettings.source_object
- FloorConstraint.target
- FluidDomainSettings.guide_parent
- NodeSocketObject.default_value
- NodeTreeInterfaceSocketObject.default_
- NormalEditModifier.target
- Object.find_armature
- Object.parent
- ObjectBase.object
- ObjectSolverConstraint.camera
- ParticleEdit.object
- ParticleEdit.shape_object
- ParticleHairKey.co_object
- ParticleHairKey.co_object_set
- ParticleInstanceModifier.object
- ParticleSettings.instance_object
- ParticleSettingsTextureSlot.object
- ParticleSystem.co_hair
- ParticleSystem.parent
- ParticleSystem.reactor_target_object
- ParticleTarget.object
- PivotConstraint.target
- PoseBone.custom_shape
- RenderEngine.bake
- RenderEngine.camera_model_matrix
- RenderEngine.camera_override
- RenderEngine.camera_shift_x
- RenderEngine.use_spherical_stereo
- RigidBodyConstraint.object1
- RigidBodyConstraint.object2
- RigidBodyWorld.convex_sweep_test
- BakeSettings.cage_object
- Scene.camera
- Scene.objects
- Scene.ray_cast
- Scene.uvedit_aspect
- SceneStrip.scene_camera
- ScrewModifier.object
- Sculpt.gravity_object
- ShaderFxShadow.object
- ShaderFxSwirl.object
- ShaderNodeTexCoord.object
- ShaderNodeTexPointDensity.object
- ShrinkwrapConstraint.target
- ShrinkwrapModifier.auxiliary_target
- ShrinkwrapModifier.target
- SimpleDeformModifier.origin
- SpaceView3D.camera
- SpaceView3D.lock_object
- SplineIKConstraint.target
- StretchToConstraint.target

- FollowPathConstraint.target
- FollowTrackConstraint.camera
- FollowTrackConstraint.depth_object
- GPencilSculptGuide.reference_object
- GeometryNodeInputObject.object
- GreasePencilArmatureModifier.object
- GreasePencilArrayModifier.offset_object
- GreasePencilBuildModifier.object
- GreasePencilHookModifier.object
- GreasePencilLatticeModifier.object
- GreasePencilLayer.parent
- GreasePencilLineartModifier.light_contour_object
- GreasePencilLineartModifier.source_camera
- GreasePencilLineartModifier.source_object
- GreasePencilMirrorModifier.object
- GreasePencilOutlineModifier.object
- GreasePencilShrinkwrapModifier.auxiliary_target
- GreasePencilShrinkwrapModifier.target
- GreasePencilTintModifier.object
- GreasePencilWeightProximityModifier.object
- HookModifier.object
- SurfaceDeformModifier.target
- TextCurve.follow_curve
- TimelineMarker.camera
- TrackToConstraint.target
- TransformConstraint.target
- UVProjector.object
- UVWarpModifier.object_from
- UVWarpModifier.object_to
- VertexWeightEditModifier.mask_tex_map_
- VertexWeightMixModifier.mask_tex_map_o
- VertexWeightProximityModifier.mask_tex
- VertexWeightProximityModifier.target
- ViewLayer.objects
- VolumeDdisplaceModifier.texture_map_obj
- VolumeToMeshModifier.object
- WarpModifier.object_from
- WarpModifier.object_to
- WarpModifier.texture_coords_object
- WaveModifier.start_position_object
- WaveModifier.texture_coords_object
- XrSessionSettings.base_pose_object