Skip to content

bpy_extras submodule (bpy_extras.object_utils)

bpy_extras.object_utils.**add_object_align_init(context, operator)**

Return a matrix using the operator settings and view context.

PARAMETERS:

- **context** (`bpy.types.Context`) – The context to use.
- **operator** (`bpy.types.Operator`) – The operator, checked for location and rotation properties.

RETURNS:

the matrix from the context and settings.

RETURN TYPE:

`mathutils.Matrix`

bpy_extras.object_utils.**object_data_add(context, obdata, operator=None, name=None)**

Add an object using the view context and preference to initialize the location, rotation and layer.

PARAMETERS:

- **context** (`bpy.types.Context`) – The context to use.
- **obdata** (`bpy.types.ID` | None) – Valid object data to used for the new object or None.
- **operator** (`bpy.types.Operator`) – The operator, checked for location and rotation properties.
- **name** (*str*) – Optional name

RETURNS:

the newly created object in the scene.

RETURN TYPE:

`bpy.types.Object`

bpy_extras.object_utils.**object_add_grid_scale(context)**

Return scale which should be applied on object data to align it to grid scale

bpy_extras.object_utils.**object_add_grid_scale_apply_operator(operator, context)**

Scale an operators distance values by the grid size.

bpy_extras.object_utils.**world_to_camera_view(scene, obj, coord)**

Returns the camera space coords for a 3d point. (also known as: normalized device coordinates - NDC).

Where (0, 0) is the bottom left and (1, 1) is the top right of the camera frame. values outside 0-1 are also supported. A negative ‘z’ value means th point is behind the camera.

Takes shift-x/y, lens angle and sensor size into account as well as perspective/ortho projections.

PARAMETERS:

- **scene** (`bpy.types.Scene`) – Scene to use for frame size.
- **obj** (`bpy.types.Object`) – Camera object.
- **coord** (`mathutils.Vector`) – World space location.

RETURNS:

a vector where X and Y map to the view plane and Z is the depth on the view axis.

RETURN TYPE:

`mathutils.Vector`

bpy_extras.object_utils.**object_report_if_active_shape_key_is_locked(obj, operator)**

Checks if the active shape key of the specified object is locked, and reports an error if so.

If the object has no shape keys, there is nothing to lock, and the function returns False.

PARAMETERS:

- **obj** (`bpy.types.Object`) – Object to check.
- **operator** (`bpy.types.Operator`) – Currently running operator to report the error through. Use None to suppress emitting the message.

RETURNS:

True if the shape key was locked.

class bpy_extras.object_utils.**AddObjectHelper**

align_update_callback(_context)