

[Skip to content](#)

Context Access (`bpy.context`)

The context members available depend on the area of Blender which is currently being accessed.

Note that all context values are read-only, but may be modified through the data API or by running operators.

Global Context

These properties are available in any contexts.

`bpy.context.area`

TYPE:

`bpy.types.Area` , (readonly)

`bpy.context.asset`

TYPE:

`bpy.types.AssetRepresentation` , (readonly)

`bpy.context.blend_data`

TYPE:

`bpy.types.BlendData` , (readonly)

`bpy.context.collection`

TYPE:

`bpy.types.Collection` , (readonly)

`bpy.context.engine`

TYPE:

string, default “”, (readonly, never None)

`bpy.context.gizmo_group`

TYPE:

`bpy.types.GizmoGroup` , (readonly)

`bpy.context.layer_collection`

TYPE:

`bpy.types.LayerCollection` , (readonly)

`bpy.context.mode`

TYPE:

enum in [Context Mode Items](#), default ‘EDIT_MESH’, (readonly)

`bpy.context.preferences`

TYPE:

`bpy.types.Preferences` , (readonly)

`bpy.context.region`

TYPE:

`bpy.types.Region` , (readonly)

`bpy.context.region_data`

TYPE:

`bpy.types.RegionView3D` (readonly)

`bpy.types.RegionView3D`, (readonly)

bpy.context.region_popup

The temporary region for pop-ups (including menus and pop-overs)

TYPE:

`bpy.types.Region`, (readonly)

bpy.context.scene

TYPE:

`bpy.types.Scene`, (readonly)

bpy.context.screen

TYPE:

`bpy.types.Screen`, (readonly)

bpy.context.space_data

The current space, may be None in background-mode, when the cursor is outside the window or when using menu-search

TYPE:

`bpy.types.Space`, (readonly)

bpy.context.tool_settings

TYPE:

`bpy.types.ToolSettings`, (readonly)

bpy.context.view_layer

TYPE:

`bpy.types.ViewLayer`, (readonly)

bpy.context.window

TYPE:

`bpy.types.Window`, (readonly)

bpy.context.window_manager

TYPE:

`bpy.types.WindowManager`, (readonly)

bpy.context.workspace

TYPE:

`bpy.types.WorkSpace`, (readonly)

Buttons Context

bpy.context.texture_slot

TYPE:

`bpy.types.TextureSlot`

bpy.context.scene

TYPE:

`bpy.types.Scene`

bpy.context.world

TYPE:

`bpy.types.World`

`bpy.types.World`

`bpy.context.object`

TYPE:

`bpy.types.Object`

`bpy.context.mesh`

TYPE:

`bpy.types.Mesh`

`bpy.context.armature`

TYPE:

`bpy.types.Armature`

`bpy.context.lattice`

TYPE:

`bpy.types.Lattice`

`bpy.context.curve`

TYPE:

`bpy.types.Curve`

`bpy.context.meta_ball`

TYPE:

`bpy.types.MetaBall`

`bpy.context.light`

TYPE:

`bpy.types.Light`

`bpy.context.speaker`

TYPE:

`bpy.types.Speaker`

`bpy.context.lightprobe`

TYPE:

`bpy.types.LightProbe`

`bpy.context.camera`

TYPE:

`bpy.types.Camera`

`bpy.context.material`

TYPE:

`bpy.types.Material`

`bpy.context.material_slot`

TYPE:

`bpy.types.MaterialSlot`

`bpy.context.texture`

TYPE:

`bpy.types.Texture`

bpy.context.texture_user

TYPE:

`bpy.types.ID`

bpy.context.texture_user_property

TYPE:

`bpy.types.Property`

bpy.context.texture_node

TYPE:

`bpy.types.Node`

bpy.context.bone

TYPE:

`bpy.types.Bone`

bpy.context.edit_bone

TYPE:

`bpy.types.EditBone`

bpy.context.pose_bone

TYPE:

`bpy.types.PoseBone`

bpy.context.particle_system

TYPE:

`bpy.types.ParticleSystem`

bpy.context.particle_system_editable

TYPE:

`bpy.types.ParticleSystem`

bpy.context.particle_settings

TYPE:

`bpy.types.ParticleSettings`

bpy.context.cloth

TYPE:

`bpy.types.ClothModifier`

bpy.context.soft_body

TYPE:

`bpy.types.SoftBodyModifier`

bpy.context.fluid

TYPE:

`bpy.types.FluidSimulationModifier`

bpy.context.collision

TYPE:

`bpy.types.CollisionModifier`

bpy.context.brush

TYPE:

`bpy.types.Brush`

`bpy.context.dynamic_paint`

TYPE:

`bpy.types.DynamicPaintModifier`

`bpy.context.line_style`

TYPE:

`bpy.types.FreestyleLineStyle`

`bpy.context.collection`

TYPE:

`bpy.types.LayerCollection`

`bpy.context.gpencil`

TYPE:

`bpy.types.GreasePencil`

`bpy.context.grease_pencil`

TYPE:

`bpy.types.GreasePencilv3`

`bpy.context.curves`

TYPE:

Hair Curves

`bpy.context.volume`

TYPE:

`bpy.types.Volume`

Clip Context

`bpy.context.edit_movieclip`

TYPE:

`bpy.types.MovieClip`

`bpy.context.edit_mask`

TYPE:

`bpy.types.Mask`

File Context

`bpy.context.active_file`

TYPE:

`bpy.types.FileSelectEntry`

`bpy.context.selected_files`

TYPE:

sequence of `bpy.types.FileSelectEntry`

`bpy.context.asset_library_reference`

TYPE:

`bpy.types.AssetLibraryReference`

`bpy.context.selected_assets`

TYPE:

sequence of `bpy.types.AssetRepresentation`

`bpy.context.id`

TYPE:

`bpy.types.ID`

`bpy.context.selected_ids`

TYPE:

sequence of `bpy.types.ID`

Image Context

`bpy.context.edit_image`

TYPE:

`bpy.types.Image`

`bpy.context.edit_mask`

TYPE:

`bpy.types.Mask`

Node Context

`bpy.context.selected_nodes`

TYPE:

sequence of `bpy.types.Node`

`bpy.context.active_node`

TYPE:

`bpy.types.Node`

`bpy.context.light`

TYPE:

`bpy.types.Light`

`bpy.context.material`

TYPE:

`bpy.types.Material`

`bpy.context.world`

TYPE:

`bpy.types.World`

Screen Context

`bpy.context.scene`

TYPE:

`bpy.types.Scene`

bpy.context.view_layer

TYPE:

`bpy.types.ViewLayer`

bpy.context.visible_objects

TYPE:

sequence of `bpy.types.Object`

bpy.context.selectable_objects

TYPE:

sequence of `bpy.types.Object`

bpy.context.selected_objects

TYPE:

sequence of `bpy.types.Object`

bpy.context.editable_objects

TYPE:

sequence of `bpy.types.Object`

bpy.context.selected_editable_objects

TYPE:

sequence of `bpy.types.Object`

bpy.context.objects_in_mode

TYPE:

sequence of `bpy.types.Object`

bpy.context.objects_in_mode_unique_data

TYPE:

sequence of `bpy.types.Object`

bpy.context.visible_bones

TYPE:

sequence of `bpy.types.EditBone`

bpy.context.editable_bones

TYPE:

sequence of `bpy.types.EditBone`

bpy.context.selected_bones

TYPE:

sequence of `bpy.types.EditBone`

bpy.context.selected_editable_bones

TYPE:

sequence of `bpy.types.EditBone`

bpy.context.visible_pose_bones

TYPE:

sequence of `bpy.types.PoseBone`

bpy.context.selected pose bones

TYPE:

sequence of `bpy.types.PoseBone`

`bpy.context.selected_pose_bones_from_active_object`

TYPE:

sequence of `bpy.types.PoseBone`

`bpy.context.active_bone`

TYPE:

`bpy.types.EditBone` or `bpy.types.Bone`

`bpy.context.active_pose_bone`

TYPE:

`bpy.types.PoseBone`

`bpy.context.active_object`

TYPE:

`bpy.types.Object`

`bpy.context.object`

TYPE:

`bpy.types.Object`

`bpy.context.edit_object`

TYPE:

`bpy.types.Object`

`bpy.context.sculpt_object`

TYPE:

`bpy.types.Object`

`bpy.context.vertex_paint_object`

TYPE:

`bpy.types.Object`

`bpy.context.weight_paint_object`

TYPE:

`bpy.types.Object`

`bpy.context.image_paint_object`

TYPE:

`bpy.types.Object`

`bpy.context.particle_edit_object`

TYPE:

`bpy.types.Object`

`bpy.context.pose_object`

TYPE:

`bpy.types.Object`

`bpy.context.active_sequence_strip`

TYPE:

`bpy.types.Strip`

`bpy.context.sequences`

TYPE:

sequence of `bpy.types.Strip`

`bpy.context.selected_sequences`

TYPE:

sequence of `bpy.types.Strip`

`bpy.context.selected_editable_sequences`

TYPE:

sequence of `bpy.types.Strip`

`bpy.context.active_nla_track`

TYPE:

`bpy.types.NlaTrack`

`bpy.context.active_nla_strip`

TYPE:

`bpy.types.NlaStrip`

`bpy.context.selected_nla_strips`

TYPE:

sequence of `bpy.types.NlaStrip`

`bpy.context.selected_movieclip_tracks`

TYPE:

sequence of `bpy.types.MovieTrackingTrack`

`bpy.context.annotation_data`

TYPE:

`bpy.types.GreasePencil`

`bpy.context.annotation_data_owner`

TYPE:

`bpy.types.ID`

`bpy.context.active_annotation_layer`

TYPE:

`bpy.types.GPencilLayer`

`bpy.context.grease_pencil`

TYPE:

`bpy.types.GreasePencilv3`

`bpy.context.active_operator`

TYPE:

`bpy.types.Operator`

`bpy.context.active_action`

TYPE:

`bpy.types.Action`

-- --

bpy.context.selected_visible_actions

TYPE:

sequence of `bpy.types.Action`

bpy.context.selected_editable_actions

TYPE:

sequence of `bpy.types.Action`

bpy.context.visible_fcurves

TYPE:

sequence of `bpy.types.FCurve`

bpy.context.editable_fcurves

TYPE:

sequence of `bpy.types.FCurve`

bpy.context.selected_visible_fcurves

TYPE:

sequence of `bpy.types.FCurve`

bpy.context.selected_editable_fcurves

TYPE:

sequence of `bpy.types.FCurve`

bpy.context.active_editable_fcurve

TYPE:

`bpy.types.FCurve`

bpy.context.selected_editable_keyframes

TYPE:

sequence of `bpy.types.Keyframe`

bpy.context.ui_list

TYPE:

`bpy.types.UIList`

bpy.context.property

TYPE:

`bpy.types.AnyType` or str or int

Get the property associated with a hovered button. Returns a tuple of the data-block, data path to the property, and array index.

Note

When the property doesn't have an associated `bpy.types.ID` non-ID data may be returned. This may occur when accessing windowing data, for example, operator properties.

```
# Example inserting keyframe for the hovered property.
active_property = bpy.context.property
if active_property:
    datablock, data_path, index = active_property
    datablock.keyframe_insert(data_path=data_path, index=index, frame=1)
```

bpy.context.asset_library_reference

TYPE:

[bpy.types.AssetLibraryReference](#)

bpy.context.active_strip

TYPE:

[bpy.types.Strip](#)

bpy.context.strips

TYPE:

sequence of [bpy.types.Strip](#)

bpy.context.selected_strips

TYPE:

sequence of [bpy.types.Strip](#)

bpy.context.selected_editable_strips

TYPE:

sequence of [bpy.types.Strip](#)

Sequencer Context

bpy.context.edit_mask

TYPE:

[bpy.types.Mask](#)

Text Context

bpy.context.edit_text

TYPE:

[bpy.types.Text](#)

View3D Context

bpy.context.active_object

TYPE:

[bpy.types.Object](#)

bpy.context.selected_ids

TYPE:

sequence of [bpy.types.ID](#)