Skip to content

EditBone(bpy_struct)

base class — `bpy_struct`

class bpy.types.**EditBone(bpy_struct)**

Edit mode bone in an armature data-block

bbone_curveinx

X-axis handle offset for start of the B-Bone's curve, adjusts curvature

TYPE:

float in [-inf, inf], default 0.0

bbone_curveinz

Z-axis handle offset for start of the B-Bone's curve, adjusts curvature

TYPE:

float in [-inf, inf], default 0.0

bbone_curveoutx

X-axis handle offset for end of the B-Bone's curve, adjusts curvature

TYPE:

float in [-inf, inf], default 0.0

bbone_curveoutz

Z-axis handle offset for end of the B-Bone's curve, adjusts curvature

TYPE:

float in [-inf, inf], default 0.0

bbone_custom_handle_end

Bone that serves as the end handle for the B-Bone curve

TYPE:

`EditBone`

bbone_custom_handle_start

Bone that serves as the start handle for the B-Bone curve

TYPE:

`EditBone`

bbone_easein

Length of first Bézier Handle (for B-Bones only)

TYPE:

float in [-inf, inf], default 1.0

bbone_easeout

Length of second Bézier Handle (for B-Bones only)

TYPE:

float in [-inf, inf], default 1.0

bbone_handle_type_end

Selects how the end handle of the B-Bone is computed

- `AUTO` Automatic – Use connected parent and children to compute the handle.
- `ABSOLUTE` Absolute – Use the position of the specified bone to compute the handle.
- `RELATIVE` Relative – Use the offset of the specified bone from rest pose to compute the handle.
- `TANGENT` Tangent – Use the orientation of the specified bone to compute the handle, ignoring the location.

TYPE:
enum in [‘AUTO’, ‘ABSOLUTE’, ‘RELATIVE’, ‘TANGENT’], default ‘AUTO’

bbone_handle_type_start

Selects how the start handle of the B-Bone is computed

- `AUTO` Automatic – Use connected parent and children to compute the handle.
- `ABSOLUTE` Absolute – Use the position of the specified bone to compute the handle.
- `RELATIVE` Relative – Use the offset of the specified bone from rest pose to compute the handle.
- `TANGENT` Tangent – Use the orientation of the specified bone to compute the handle, ignoring the location.

TYPE:
enum in [‘AUTO’, ‘ABSOLUTE’, ‘RELATIVE’, ‘TANGENT’], default ‘AUTO’

bbone_handle_use_ease_end

Multiply the B-Bone Ease Out channel by the local Y scale value of the end handle. This is done after the Scale Easing option and isn’t affecte by it.

TYPE:
boolean, default False

bbone_handle_use_ease_start

Multiply the B-Bone Ease In channel by the local Y scale value of the start handle. This is done after the Scale Easing option and isn’t affectec by it.

TYPE:
boolean, default False

bbone_handle_use_scale_end

Multiply B-Bone Scale Out channels by the local scale values of the end handle. This is done after the Scale Easing option and isn’t affected b it.

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

bbone_handle_use_scale_start

Multiply B-Bone Scale In channels by the local scale values of the start handle. This is done after the Scale Easing option and isn’t affected by

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

bbone_mapping_mode

Selects how the vertices are mapped to B-Bone segments based on their position

- `STRAIGHT` Straight – Fast mapping that is good for most situations, but ignores the rest pose curvature of the B-Bone.
- `CURVED` Curved – Slower mapping that gives better deformation for B-Bones that are sharply curved in rest pose.

TYPE:
enum in [‘STRAIGHT’, ‘CURVED’], default ‘STRAIGHT’

bbone_rollin

Roll offset for the start of the B-Bone, adjusts twist

TYPE:

float in [-inf, inf], default 0.0

bbone_rollout

Roll offset for the end of the B-Bone, adjusts twist

TYPE:

float in [-inf, inf], default 0.0

bbone_scalein

Scale factors for the start of the B-Bone, adjusts thickness (for tapering effects)

TYPE:

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

bbone_scaleout

Scale factors for the end of the B-Bone, adjusts thickness (for tapering effects)

TYPE:

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

bbone_segments

Number of subdivisions of bone (for B-Bones only)

TYPE:

int in [1, 32], default 0

bbone_x

B-Bone X size

TYPE:

float in [-inf, inf], default 0.0

bbone_z

B-Bone Z size

TYPE:

float in [-inf, inf], default 0.0

collections

Bone Collections that contain this bone

TYPE:

`bpy_prop_collection` of `BoneCollection`, (readonly)

color

TYPE:

`BoneColor`, (readonly)

envelope_distance

Bone deformation distance (for Envelope deform only)

TYPE:

float in [0, 1000], default 0.0

envelope_weight

Bone deformation weight (for Envelope deform only)

TYPE:
float in [0, 1000], default 0.0

head

Location of head end of the bone

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

head_radius

Radius of head of bone (for Envelope deform only)

TYPE:
float in [-inf, inf], default 0.0

hide

Bone is not visible when in Edit Mode

TYPE:
boolean, default False

hide_select

Bone is able to be selected

TYPE:
boolean, default False

inherit_scale

Specifies how the bone inherits scaling from the parent bone

- `FULL` Full – Inherit all effects of parent scaling.
- `FIX_SHEAR` Fix Shear – Inherit scaling, but remove shearing of the child in the rest orientation.
- `ALIGNED` Aligned – Rotate non-uniform parent scaling to align with the child, applying parent X scale to child X axis, and so forth.
- `AVERAGE` Average – Inherit uniform scaling representing the overall change in the volume of the parent.
- `NONE` None – Completely ignore parent scaling.
- `NONE_LEGACY` None (Legacy) – Ignore parent scaling without compensating for parent shear. Replicates the effect of disabling the original Inherit Scale checkbox..

TYPE:
enum in ['FULL', 'FIX_SHEAR', 'ALIGNED', 'AVERAGE', 'NONE', 'NONE_LEGACY'], default 'FULL'

length

Length of the bone. Changing moves the tail end.

TYPE:
float in [0, inf], default 0.0

lock

Bone is not able to be transformed when in Edit Mode

TYPE:
boolean, default False

matrix

Matrix combining location and rotation of the bone (head position, direction and roll), in armature space (does not include/support bone's

Matrix combining location and rotation of the bone (head position, direction and roll), in armature space (does not include/support bone's length/size)

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))

name

TYPE:

string, default "", (never None)

parent

Parent edit bone (in same Armature)

TYPE:

`EditBone`

roll

Bone rotation around head-tail axis

TYPE:

float in [-inf, inf], default 0.0

select

TYPE:

boolean, default False

select_head

TYPE:

boolean, default False

select_tail

TYPE:

boolean, default False

show_wire

Bone is always displayed in wireframe regardless of viewport shading mode (useful for non-obstructive custom bone shapes)

TYPE:

boolean, default False

tail

Location of tail end of the bone

TYPE:

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

tail_radius

Radius of tail of bone (for Envelope deform only)

TYPE:

float in [-inf, inf], default 0.0

use_connect

When bone has a parent, bone's head is stuck to the parent's tail

TYPE:

boolean, default False

use_cyclic_offset

When bone doesn't have a parent, it receives cyclic offset effects (Deprecated)

TYPE:

boolean, default False

use_deform

Enable Bone to deform geometry

TYPE:

boolean, default False

use_endroll_as_inroll

Add Roll Out of the Start Handle bone to the Roll In value

TYPE:

boolean, default False

use_envelope_multiply

When deforming bone, multiply effects of Vertex Group weights with Envelope influence

TYPE:

boolean, default False

use_inherit_rotation

Bone inherits rotation or scale from parent bone

TYPE:

boolean, default False

use_local_location

Bone location is set in local space

TYPE:

boolean, default False

use_relative_parent

Object children will use relative transform, like deform

TYPE:

boolean, default False

use_scale_easing

Multiply the final easing values by the Scale In/Out Y factors

TYPE:

boolean, default False

basename

The name of this bone before any '.' character

(readonly)

center

The midpoint between the head and the tail.

(readonly)

children

A list of all the bones children.

> Note
>
> Takes `O(len(bones))` time.

(readonly)

children_recursive

A list of all children from this bone.

> Note
>
> Takes `O(len(bones)**2)` time.

(readonly)

children_recursive_basename

Returns a chain of children with the same base name as this bone. Only direct chains are supported, forks caused by multiple children with matching base names will terminate the function and not be returned.

> Note
>
> Takes `O(len(bones)**2)` time.

(readonly)

parent_recursive

A list of parents, starting with the immediate parent

(readonly)

vector

The direction this bone is pointing. Utility function for (tail - head)

(readonly)

x_axis

Vector pointing down the x-axis of the bone.

(readonly)

y_axis

Vector pointing down the y-axis of the bone.

(readonly)

z_axis

Vector pointing down the z-axis of the bone.

(readonly)

align_roll(vector)

Align the bone to a local-space roll so the Z axis points in the direction of the vector given

PARAMETERS:

vector (`mathutils.Vector` of 3 items in [-inf, inf]) – Vector

align_orientation(other)

Align this bone to another by moving its tail and settings its roll the length of the other bone is not used.

parent_index(parent_test)

The same as 'bone in other_bone.parent_recursive' but saved generating a list.

transform(matrix, *, scale=True, roll=True)

Transform the bones head, tail, roll and envelope (when the matrix has a scale component).

PARAMETERS:

- **matrix** (`mathutils.Matrix`) – 3x3 or 4x4 transformation matrix.
- **scale** (*bool*) – Scale the bone envelope by the matrix.
- **roll** (*bool*) – Correct the roll to point in the same relative direction to the head and tail.

translate(vec)

Utility function to add *vec* to the head and tail of this bone

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`

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.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.is_property_hidden
- bpy_struct.is_property_overridable_library
- bpy_struct.is_property_readonly
- bpy_struct.is_property_set
- bpy_struct.property_overridable_library_set
- bpy_struct.property_unset
- bpy_struct.type_recast
- bpy_struct.values

References

- bpy.context.active_bone
- bpy.context.edit_bone
- bpy.context.editable_bones
- bpy.context.selected_bones
- bpy.context.selected_editable_bones
- bpy.context.visible_bones
- Armature.edit_bones
- ArmatureEditBones.active
- ArmatureEditBones.new
- ArmatureEditBones.remove
- EditBone.bbone_custom_handle_end
- EditBone.bbone_custom_handle_start
- EditBone.parent

Copyright © Blender Authors
Made with Furo

Report issue on this page