KeyMapItems(bpy_struct)

base class — bpy_struct

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

Collection of keymap items

new(idname, type, value, *, any=False, shift=0, ctrl=0, alt=0, oskey=0, key_modifier='NONE', direction='ANY', repeat=False, head=False)

new

PARAMETERS:

- **idname** (*string, (never None)*) – Operator Identifier
- **type** (enum in Event Type Items) – Type
- **value** (enum in Event Value Items) – Value
- **any** (*boolean, (optional)*) – Any
- **shift** (*int in [-1, 1], (optional)*) – Shift
- **ctrl** (*int in [-1, 1], (optional)*) – Ctrl
- **alt** (*int in [-1, 1], (optional)*) – Alt
- **oskey** (*int in [-1, 1], (optional)*) – OS Key
- **key_modifier** (enum in Event Type Items, (optional)) – Key Modifier
- **direction** (enum in Event Direction Items, (optional)) – Direction
- **repeat** (*boolean, (optional)*) – Repeat, When set, accept key-repeat events
- **head** (*boolean, (optional)*) – At Head, Force item to be added at start (not end) of key map so that it doesn't get blocked by an existir key map item

RETURNS:

Item, Added key map item

RETURN TYPE:

KeyMapItem

new_modal(propvalue, type, value, *, any=False, shift=0, ctrl=0, alt=0, oskey=0, key_modifier='NONE', direction='ANY', repeat=False)

new_modal

PARAMETERS:

- **propvalue** (*string, (never None)*) – Property Value
- **type** (enum in Event Type Items) – Type
- **value** (enum in Event Value Items) – Value
- **any** (*boolean, (optional)*) – Any
- **shift** (*int in [-1, 1], (optional)*) – Shift
- **ctrl** (*int in [-1, 1], (optional)*) – Ctrl
- **alt** (*int in [-1, 1], (optional)*) – Alt
- **oskey** (*int in [-1, 1], (optional)*) – OS Key
- **key_modifier** (enum in Event Type Items, (optional)) – Key Modifier
- **direction** (enum in Event Direction Items, (optional)) – Direction
- **repeat** (*boolean, (optional)*) – Repeat, When set, accept key-repeat events

RETURNS:

Item, Added key map item

RETURN TYPE:

KeyMapItem

new_from_item(item, *, head=False)

new_from_item

PARAMETERS:

- **item** (KeyMapItem, (never None)) – Item, Item to use as a reference
- **head** (*boolean, (optional)*) – At Head

RETURNS:

Item, Added key map item

RETURN TYPE:

KeyMapItem

remove(item)

remove

PARAMETERS:

item (KeyMapItem, (never None)) – Item

from_id(id)

from_id

PARAMETERS:

id (*int in [-inf, inf]*) – id, ID of the item

RETURNS:

Item

RETURN TYPE:

KeyMapItem

find_from_operator(idname, *, properties=None, include={'ACTIONZONE', 'KEYBOARD', 'MOUSE', 'NDOF'}, exclude={})

find_from_operator

PARAMETERS:

- **idname** (*string, (never None)*) – Operator Identifier
- **include** (enum set in Event Type Mask Items, (optional)) – Include
- **exclude** (enum set in Event Type Mask Items, (optional)) – Exclude

RETURN TYPE:

KeyMapItem

match_event(event)

match_event

RETURN TYPE:

KeyMapItem

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.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

References

- KeyMap.keymap_items