Text(ID)

base classes — bpy_struct , ID

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

Text data-block referencing an external or packed text file

current_character

Index of current character in current line, and also start index of character in selection if one exists

TYPE:

int in [0, inf], default 0

current_line

Current line, and start line of selection if one exists

TYPE:

TextLine , (readonly, never None)

current_line_index

Index of current TextLine in TextLine collection

TYPE:

int in [-inf, inf], default 0

filepath

Filename of the text file

TYPE:

string, default “”, (never None)

indentation

Use tabs or spaces for indentation

- TABS Tabs – Indent using tabs.
- SPACES Spaces – Indent using spaces.

TYPE:

enum in [‘TABS’, ‘SPACES’], default ‘TABS’

is_dirty

Text file has been edited since last save

TYPE:

boolean, default False, (readonly)

is_in_memory

Text file is in memory, without a corresponding file on disk

TYPE:

boolean, default False, (readonly)

is_modified

Text file on disk is different than the one in memory

TYPE:

boolean, default False, (readonly)

lines

Lines of text

TYPE:

bpy_prop_collection of TextLine, (readonly)

select_end_character

Index of character after end of selection in the selection end line

TYPE:

int in [0, inf], default 0

select_end_line

End line of selection

TYPE:

TextLine, (readonly, never None)

select_end_line_index

Index of last TextLine in selection

TYPE:

int in [-inf, inf], default 0

use_module

Run this text as a Python script on loading

TYPE:

boolean, default False

clear()

clear the text block

write(text)

write text at the cursor location and advance to the end of the text block

PARAMETERS:

text (*string, (never None)*) – New text for this data-block

from_string(text)

Replace text with this string.

as_string()

Return the text as a string

RETURN TYPE:

string, (never None)

is_syntax_highlight_supported()

Returns True if the editor supports syntax highlighting for the current text datablock

RETURN TYPE:

boolean

select_set(line_start, char_start, line_end, char_end)

Set selection range by line and character index

PARAMETERS:

- **line_start** (*int in [-inf, inf]*) – Start Line
- **char_start** (*int in [-inf, inf]*) – Start Character
- **line_end** (*int in [-inf, inf]*) – End Line
- **char_end** (*int in [-inf, inf]*) – End Character

cursor_set(line, *, character=0, select=False)

Set cursor by line and (optionally) character index

PARAMETERS:

- **line** (*int in [0, inf]*) – Line
- **character** (*int in [0, inf], (optional)*) – Character
- **select** (*boolean, (optional)*) – Select when moving the cursor

as_module()

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

region_as_string(range=None)

PARAMETERS:

range (*tuple[tuple[int, int], tuple[int, int]]*) – The region of text to be returned, defaulting to the selection when no range is passed. Each int pair represents a line and column: ((start_line, start_column), (end_line, end_column)) The values match Python's slicing logic (negative values count backwards from the end, the end value is not inclusive).

RETURNS:

The specified region as a string.

RETURN TYPE:

str.

region_from_string(body, range=None)

PARAMETERS:

- **body** (*str*) – The text to be inserted.
- **range** (*tuple[tuple[int, int], tuple[int, int]]*) – The region of text to be returned, defaulting to the selection when no range is passed. Each int pair represents a line and column: ((start_line, start_column), (end_line, end_column)) The values match Python's slicing logic (negative values count backwards from the end, the end value is not inclusive).

Inherited Properties

Inherited Properties

- bpy_struct.id_data
- ID.name
- ID.name_full
- ID.id_type
- ID.session_uid
- ID.is_evaluated
- ID.original
- ID.users
- ID.use_fake_user
- ID.use_extra_user
- ID.is_embedded_data
- ID.is_missing
- ID.is_runtime_data
- ID.is_editable
- 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.edit_text
- BlendData.texts
- BlendDataTexts.load
- BlendDataTexts.new
- BlendDataTexts.remove
- FreestyleModuleSettings.script
- NodeFrame.text
- PythonConstraint.text
- ShaderNodeScript.script
- ShaderNodeTexIES.ies
- SpaceTextEditor.text