

[Skip to content](#)

# SpaceTextEditor(Space)

base classes — [bpy\\_struct](#), [Space](#)

**class** bpy.types.SpaceTextEditor(Space)

Text editor space data

**find\_text**

Text to search for with the find tool

**TYPE:**

string, default “”, (never None)

**font\_size**

Font size to use for displaying the text

**TYPE:**

int in [1, 256], default 0

**margin\_column**

Column number to show right margin at

**TYPE:**

int in [0, 1024], default 0

**replace\_text**

Text to replace selected text with using the replace tool

**TYPE:**

string, default “”, (never None)

**show\_line\_highlight**

Highlight the current line

**TYPE:**

boolean, default False

**show\_line\_numbers**

Show line numbers next to the text

**TYPE:**

boolean, default False

**show\_margin**

Show right margin

**TYPE:**

boolean, default False

**show\_region\_footer**

**TYPE:**

boolean, default False

**show\_region\_ui**

**TYPE:**

boolean, default False

boolean, default False

### **show\_syntax\_highlight**

Syntax highlight for scripting

#### **TYPE:**

boolean, default False

### **show\_word\_wrap**

Wrap words if there is not enough horizontal space

#### **TYPE:**

boolean, default False

### **tab\_width**

Number of spaces to display tabs with

#### **TYPE:**

int in [2, 8], default 0

### **text**

Text displayed and edited in this space

#### **TYPE:**

Text

### **top**

Top line visible

#### **TYPE:**

int in [0, inf], default 0

### **use\_find\_all**

Search in all text data-blocks, instead of only the active one

#### **TYPE:**

boolean, default False

### **use\_find\_wrap**

Search again from the start of the file when reaching the end

#### **TYPE:**

boolean, default False

### **use\_live\_edit**

Run Python while editing

#### **TYPE:**

boolean, default False

### **use\_match\_case**

Search string is sensitive to uppercase and lowercase letters

#### **TYPE:**

boolean, default False

### **use\_overwrite**

Overwrite characters when typing rather than inserting them

**TYPE:**

boolean, default False

**visible\_lines**

Amount of lines that can be visible in current editor

**TYPE:**

int in  $[-inf, inf]$ , default 0, (readonly)

**is\_syntax\_highlight\_supported()**

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

**RETURN TYPE:**

boolean

**region\_location\_from\_cursor(line, column)**

Retrieve the region position from the given line and character position

**PARAMETERS:**

- **line** (*int in  $[-inf, inf]$* ) – Line, Line index
- **column** (*int in  $[-inf, inf]$* ) – Column, Column index

**RETURNS:**

Region coordinates

**RETURN TYPE:**

int array of 2 items in  $[-1, inf]$

**classmethod bl\_ma\_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\_ma\_get\_subclass\_py(id, default=None)****PARAMETERS:**

**id** (*str*) – The RNA type identifier.

**RETURNS:**

The class or default when not found.

**RETURN TYPE:**

type

**classmethod draw\_handler\_add(callback, args, region\_type, draw\_type)**

Add a new draw handler to this space type. It will be called every time the specified region in the space type will be drawn. Note: All arguments are positional only for now.

**PARAMETERS:**

- **callback** (*Callable[[Any, ...], Any]*) – A function that will be called when the region is drawn. It gets the specified arguments as input, it's return value is ignored.
- **args** (*tuple[Any, ...]*) – Arguments that will be passed to the callback.
- **region\_type** (*str*) – The region type the callback draws in; usually `WINDOW`. (`bpy.types.Region.type`)
- **draw\_type** (*str*) – Usually `POST_PIXEL` for 2D drawing and `POST_VIEW` for 3D drawing. In some cases `PRE_VIEW` can be

used. `BACKDROP` can be used for backdrops in the node editor.

#### RETURNS:

Handler that can be removed later on.

#### RETURN TYPE:

object

#### `classmethod draw_handler_remove(handler, region_type)`

Remove a draw handler that was added previously.

#### PARAMETERS:

- **handler** (*object*) – The draw handler that should be removed.
- **region\_type** (*str*) – Region type the callback was added to.

## Inherited Properties

- `bpy_struct.id_data`
- `Space.show_locked_time`
- `Space.type`
- `Space.show_region_header`

## 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`
- `Space.bl_rna_get_subclass`
- `Space.bl_rna_get_subclass_py`
- `Space.draw_handler_add`
- `Space.draw_handler_remove`