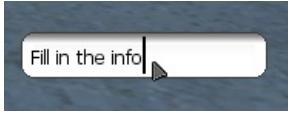


GuiCreateEdit



Example GUI edit field.

This function is for creating a new GUI edit box. This is a text box in which the user can input text. Edit boxes only allow a single line of text. If you want to allow multiple lines of text create a memo box using `guiCreateMemo`.

Syntax

```
element guiCreateEdit ( float x, float y, float width, float height, string text, [ bool relative = false, gui-element parent = nil ] )
```

OOP Syntax [Help! I don't understand this!](#)

Method: *GuiEdit(...)*

Required Arguments

- **x:** A float of the 2D x position of the GUI edit box on a player's screen. This is affected by the *relative* argument.
- **y:** A float of the 2D y position of the GUI edit box on a player's screen. This is affected by the *relative* argument.
- **width:** A float of the width of the GUI edit box. This is affected by the *relative* argument.
- **height:** A float of the height of the GUI edit box. This is affected by the *relative* argument.
- **text:** A string of the text that will be displayed by default in the edit box.

Optional Arguments

NOTE: When using optional arguments, you might need to supply all arguments before the one you wish to use. For more information on optional arguments, see optional arguments.

- **relative:** This is whether sizes and positioning are relative. If this is *true*, then all x,y,width,height floats must be between 0 and 1, representing measures relative to the parent.
- **parent:** This is the parent that the GUI edit box is attached to. If the *relative* argument is true, sizes and positioning will be made relative to this parent. If the *relative* argument is false, positioning will be the number of offset pixels from the parent's origin. If no parent is passed, the parent will become the screen - causing positioning and sizing according to screen positioning.

Returns

Returns a gui-edit element of the created edit box if it was successfully created, false otherwise.