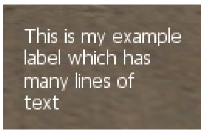
Client-side function

GuiCreateLabel



This is my example label which has many lines of text

Example GUI label.

This function is for creating a new GUI label. A label is simply a piece of text that cannot be edited by the user. If you would like to have a bigger text you'd have to change its font because font size is not supported.

Syntax

```
element guiCreateLabel ( 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**: *GuiLabel(...)*

Required Arguments

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

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 sizes relative to the parent.
- **parent:** This is the parent that the gui label 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 an element of the created label if it was successfully created, false otherwise.