🦋class

GTK::V3::Gtk::GtkToggleButt

Table of Contents

```
unit class GTK::V3::Gtk::GtkToggleButton;
also is GTK::V3::Gtk::GtkButton;
```

GtkToggleButton — Create buttons which retain their state

Synopsis

```
my GTK::V3::Gtk::GtkToggleButton $start-tggl .= new(:label('Start Process'));

# later in another class ...
method start-stop-process-handle( :widget($start-tggl) ) {
  if $start-tggl.get-active {
    $start-tggl.set-label('Stop Process');
    # start process ...
  }

  else {
    $start-tggl.set-label('Start Process');
    # stop process ...
  }
}
```

Methods

gtk_toggle_button_new

```
method gtk_toggle_button_new ( --> N-GObject )
```

Creates a new native toggle button object

[gtk_toggle_button_] new_with_label

```
method gtk_toggle_button_new_with_label ( Str $label --> N-GObject )
```

Creates a new native toggle button object with a label

[gtk_toggle_button_] new_with_mnemonic

```
method gtk_toggle_button_new_with_mnemonic ( Str $label --> N-GObject )
```

Creates a new GtkToggleButton containing a label. The label will be created using gtk_label_new_with_mnemonic(), so underscores in label indicate the mnemonic for the button.

[gtk_toggle_button_] get_active

```
method gtk_toggle_button_get_active ( --> Int )
```

Get the button state.

[gtk_toggle_button_] set_active

```
method gtk_toggle_button_set_active ( Int $active --> N-GObject )
```

Set the button state.

new

```
multi submethod BUILD ( Str :$label )
```

Create a GtkToggleButton with a label.

```
multi submethod BUILD ( Bool :$empty )
```

Create an empty GtkToggleButton.

```
multi submethod BUILD ( :$widget! )
```

Create a button using a native object from elsewhere. See also Gtk::V3::Glib::GObject.

```
multi submethod BUILD ( Str :$build-id! )
```

Create a button using a native object from a builder. See also Gtk::V3::Glib::GObject.