

```
1 //includes
2 #include "main.h"
3 #include "../src/globals.hpp"
4
5 //buttonM method definition
6 lv_res_t buttonMethod(lv_obj_t* btn){
7     //button handle code
8     return LV_RES_OK;
9 }
10
11 //method definition, method used to easily create button using pointers for continuity
12 void createButton(uint32_t number, lv_coord_t width, lv_coord_t height, lv_coord_t shiftX,
13 lv_coord_t shiftY, lv_color_t color, lv_align_t alignment, std::string text){
14     //create style, copy plain style into style, and set color
15     styles[number] = lv_style_t();
16     lv_style_copy(&styles[number], &lv_style_plain);
17     styles[number].body.main_color = color;
18     styles[number].body.grad_color = color;
19
20     //create button with alignment, size, button method/number, styles, and label with text
21     objects[number] = lv_btn_create(lv_scr_act(), NULL);
22     lv_obj_set_free_num(objects[number], number);
23     lv_btn_set_action(objects[number], LV_BTN_ACTION_CLICK, buttonMethod);
24     lv_obj_set_size(objects[number], width, height);
25     lv_obj_align(objects[number], NULL, alignment, shiftX, shiftY);
26     lv_btn_set_style(objects[number], LV_BTN_STATE_REL, &styles[number]);
27     lv_btn_set_style(objects[number], LV_BTN_STATE_PR, &lv_style_plain);
28     lv_label_set_text(lv_label_create(objects[number], NULL), text.c_str());
29 }
30
31 //example, creates two buttons
32 lv_obj_t** objects = new lv_obj_t*[2];
33 lv_style_t** styles = new lv_style_t*[2];
34 createButton(0, 160, 80, 0, 0, LV_COLOR_RED, LV_ALIGN_IN_TOP_LEFT, "Button 1");
35 createButton(1, 160, 80, 0, 0, LV_COLOR_BLUE, LV_ALIGN_IN_LEFT_MID, "Button 2");
```