```
/* functional syntax */

/* lambdas */

my_lambda fn(u32) u32 = (value) {
  ret value + 5
}

value_14 := my_func(9)

my_lambda := (value u32) string {
  ret string.from_u32(in_value)
}


// lambdas as parameters
fn my_func(in_param u32, in_lambda fn(u32) string) string {
  ret in_lambda(in_param)
}

// lambdas as arguments
str_val := my_func(6, (in_value u32) {
  ret string.from_u32(in_value)
})

// stored lambdas as arguments
str_val := my_func(6, my_lambda)

// funcitons as arguments
fn my_func_arg(in_param u32) string {
  ret string.from_u32(in_param)
}

str_val := my_func(6, my_func_arg)


PrintLn("string value= ${str_val}")
```