```
func start()
   id_$24; -- identifier expression beginning with a letter
   @id123; -- identifier expression beginning with at-sign
   5 -- numeric literal
   ; -- expressions are separated by commas
   “string”; -- string literal
   add[2, 3]; -- function call with expressions in comma-separated list in square brackets
   5 / 1; -- divide binary arithmetic operator
   4 * 6; -- times binary arithmetic operator
   3 – 2; -- minus binary arithmetic operator
   5 if 3 < 2 else 4; -- example of conditional expression
   5.4; -- numerical literal with optional fractional part
   5**2; -- numerical literal with optional exponent part
   (5+2) * 3 == 21 -- parentheses group subexpressions (no equality check explained in language but
   -- we used it to show off a different feature)
end


func add(x, y)
   x + y -- plus binary arithmetic operator
end


func negate(x)
   -x -- unary prefix negation
end


func factorial(x)
   x! -- unary postfix factorial
end
```

```
func precedence()

   3 if -3!*2+15 == 3 else “no” -- shows the precedence of operators and checks that it works
   -- correctly, printing the actual answer if the precedence works as explained.

end
```

```
“this shows off a single quote: \' in this language and a double quote: \". This breaks the string into a
new line \n I am not sure what this delimiter does but here it is \\ Here is the hex-code for white?:
\u{FFFFFF}”
```