

kolorScript

Where everything is simple!

From installation... to running your code

November 17, 2024 SVFIG Forth Day 2024

Intro



This is me ...
19 years ago ☺

- George Phillip Orais
- Filipino born in Germany ... moved to Philippines
- Highschool: GW BASIC
- BS Computer Engineer: 8088 ASM, Turbo C, PIC C, Visual BASIC, Java
- Software Engineer (Philippines): Visual C# 6.0, SQL, Visual C++ 6.0
- Embedded Software Engineer (China): 4bit ASM, Embedded C, Visual C# .Net, PicoLisp, Verilog
- Senior Embedded Software Engineer (Japan): Embedded C, Python, Lua, FORTH, Javascript



What does this logo mean?



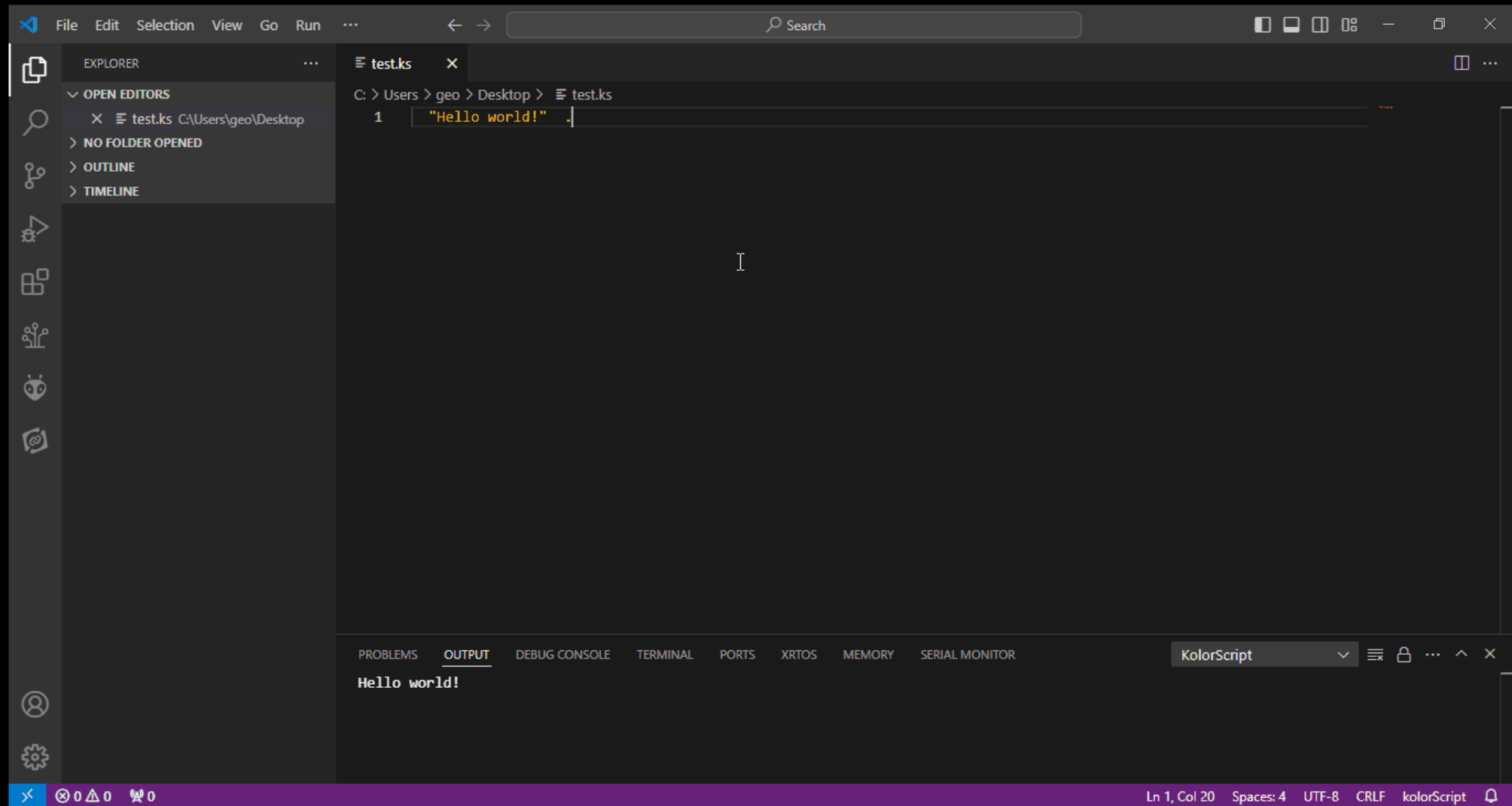
^{ko}
^{lo}
^r
Kolor = “kolor” written in Baybayin

Where:

- “kolor” means color in Filipino language
- Baybayin is an ancient Filipino script

What is **kolorScript**?

A. It is a **modern** variant of **colorForth**



➤ vscode extension

➤ ASCII text files

B. It is a **hybrid** Forth

| colorForth | ANS Forth | Others |
|--|--|--|
| * + 2/ 2* and xor ; push pop dup drop over swap nip if -if then for next @ ! load . | / - or min max abs = > < i open-file read-line close-file .s | >> << not ms .sih cls num? str? utime long-timestamp medium-timestamp short- timestamp drop-all count-all split-str index-of sub-str ... |

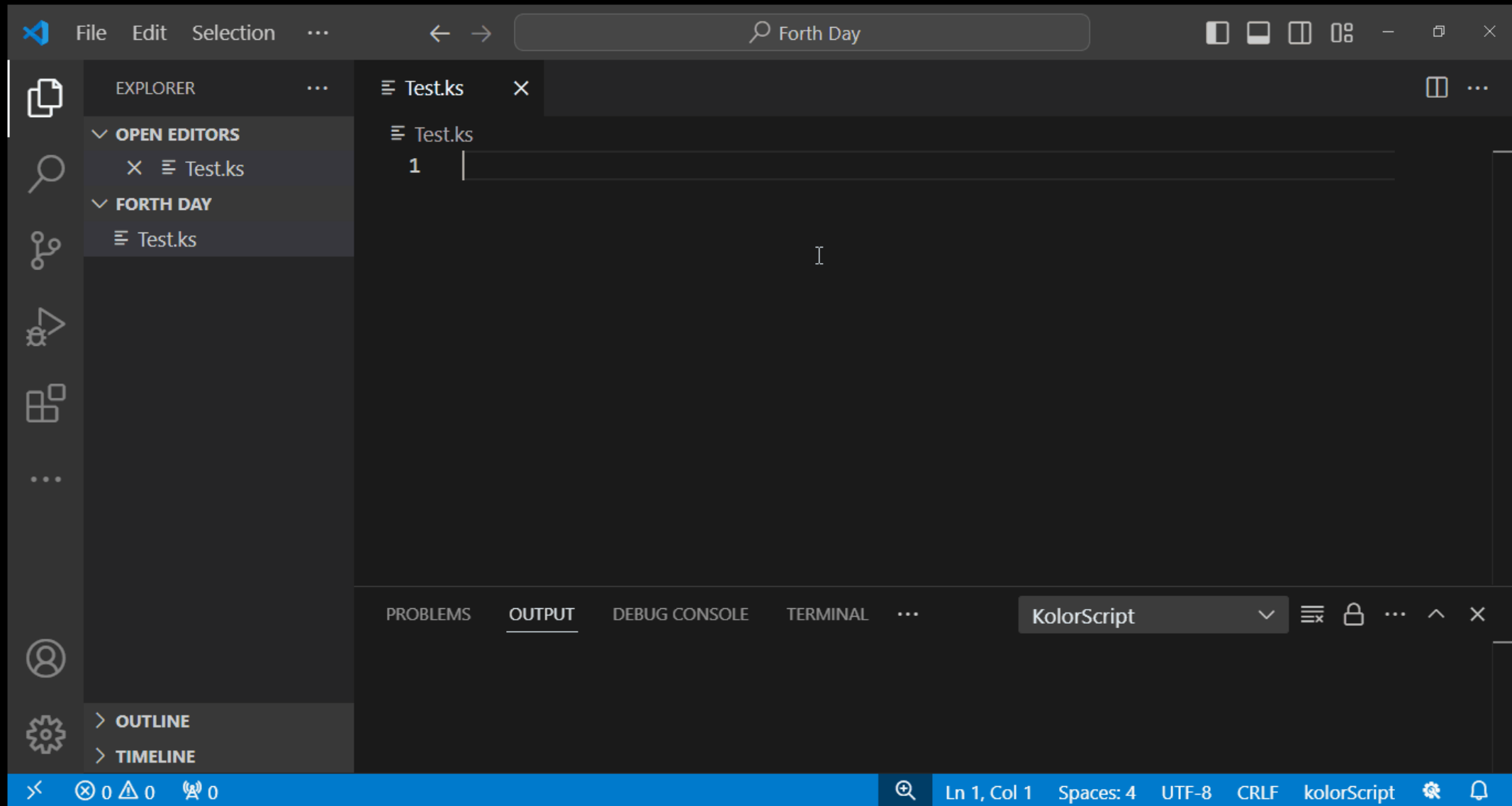
- Numbers are written as signed integer, hexadecimal, or float
- Strings are enclosed with single or double quote

C. It is a Forth intended for **scripting**

- Rapid prototyping
- Data parsing
- Code/Data generator
- Test automation (probably)
- Good alternative for Python, JavaScript, Lua or Ruby

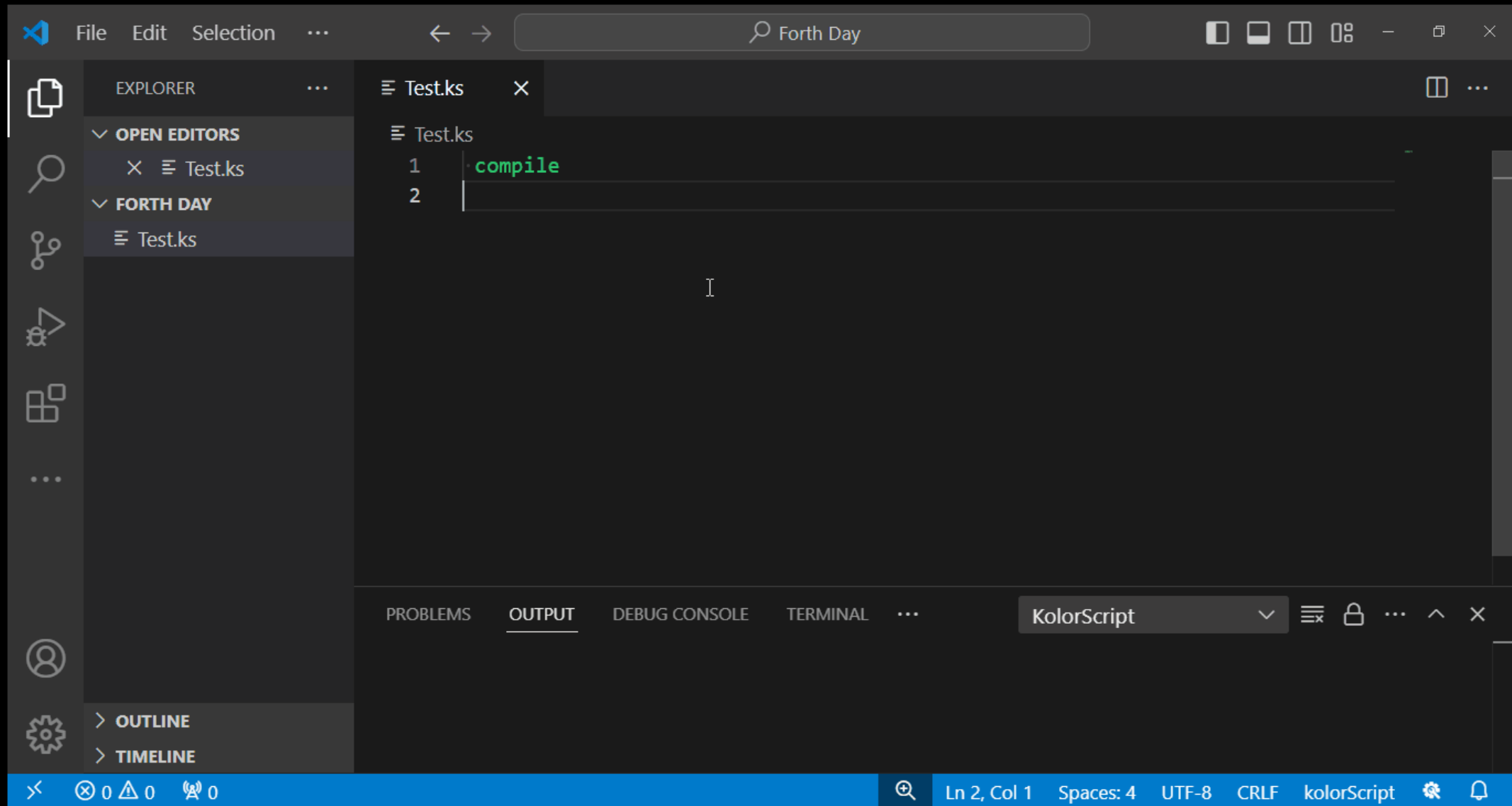
What are the colors in
kolorScript?

GREEN means “COMPILE this word”



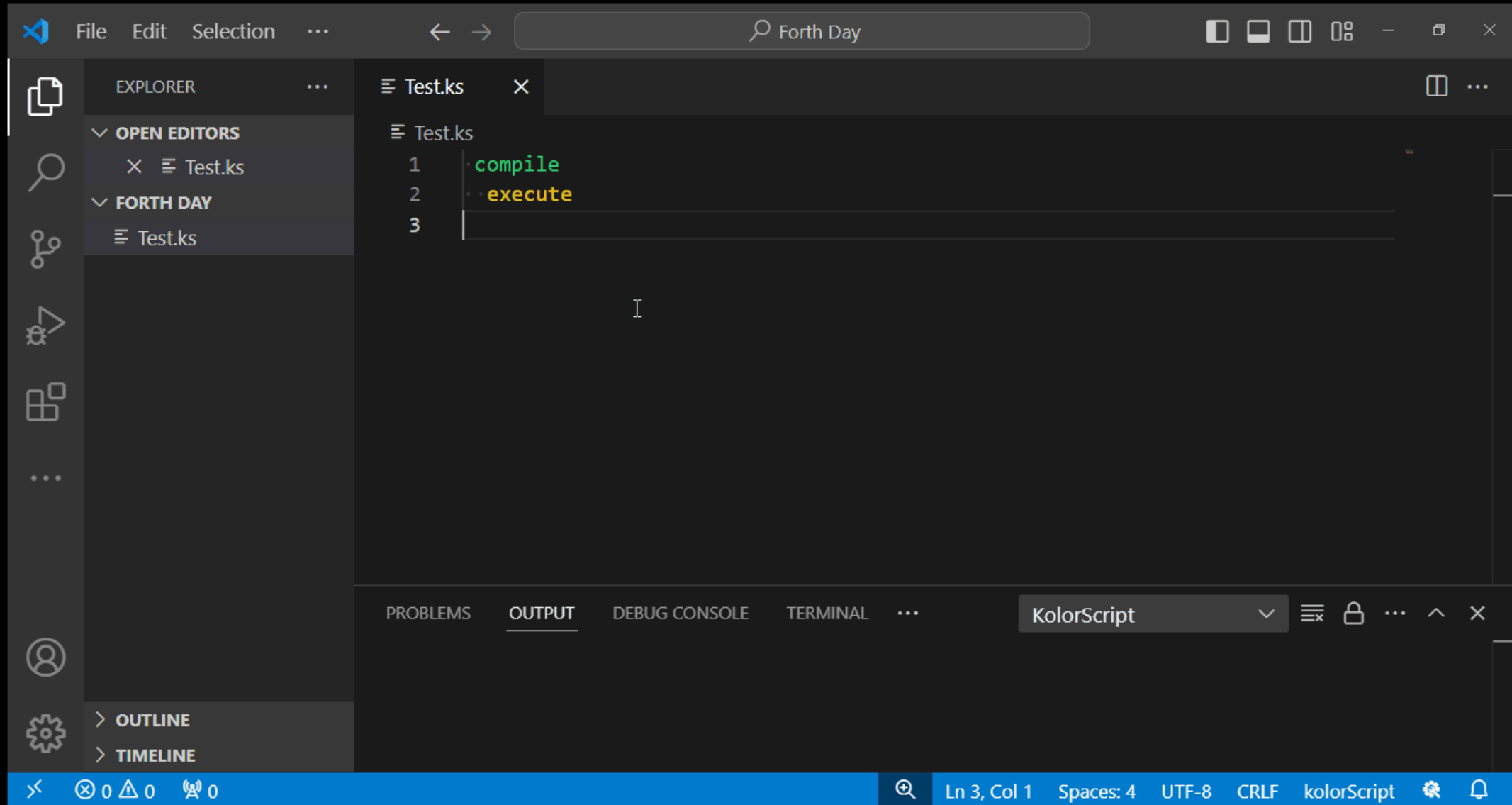
Tag rule: prefix 1 space

YELLOW means “INTERPRET this word”



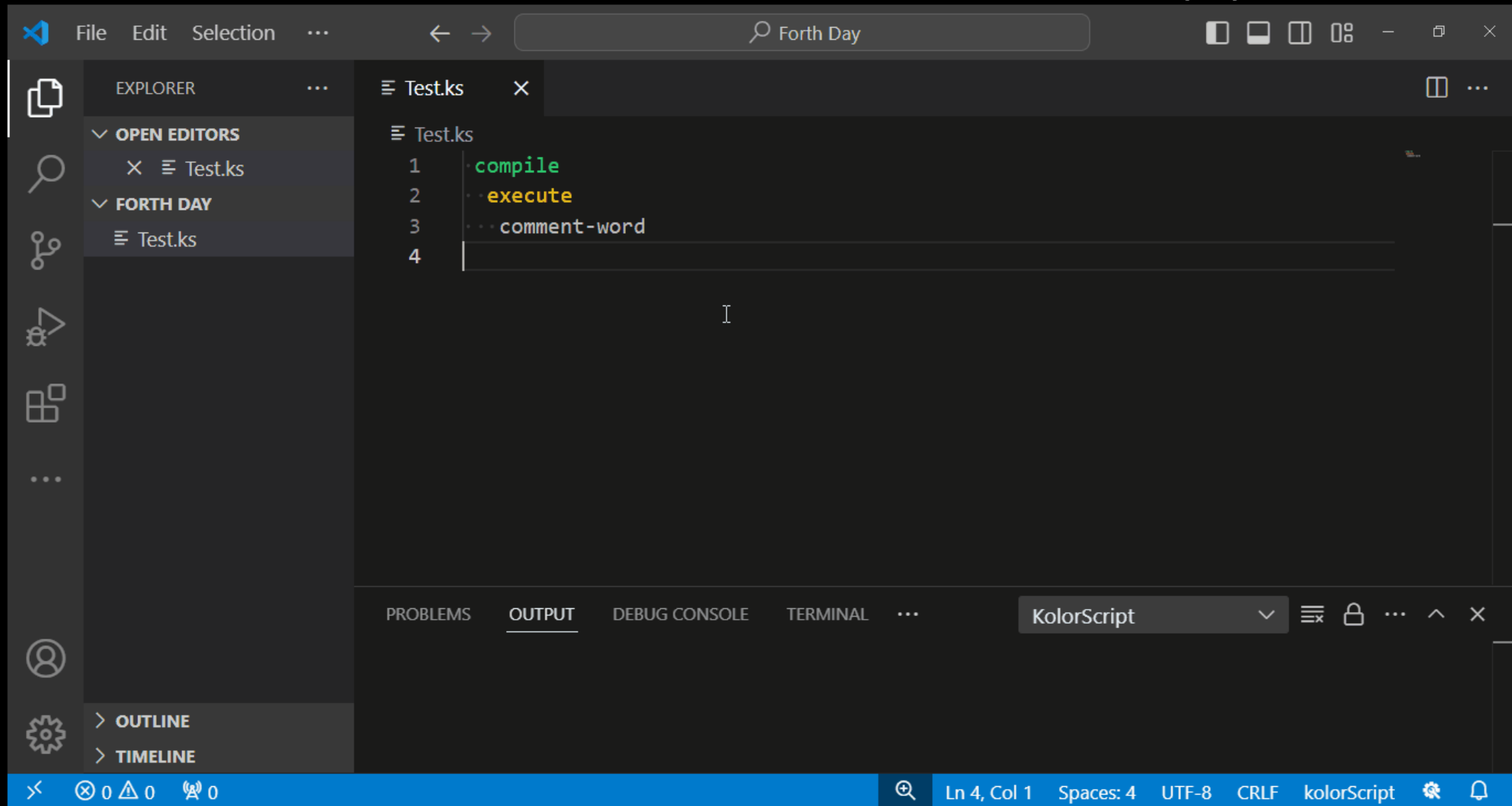
Tag rule: prefix 2 spaces

GRAY means “COMMENT out this word(s)”



Tag rule: prefix 3 spaces

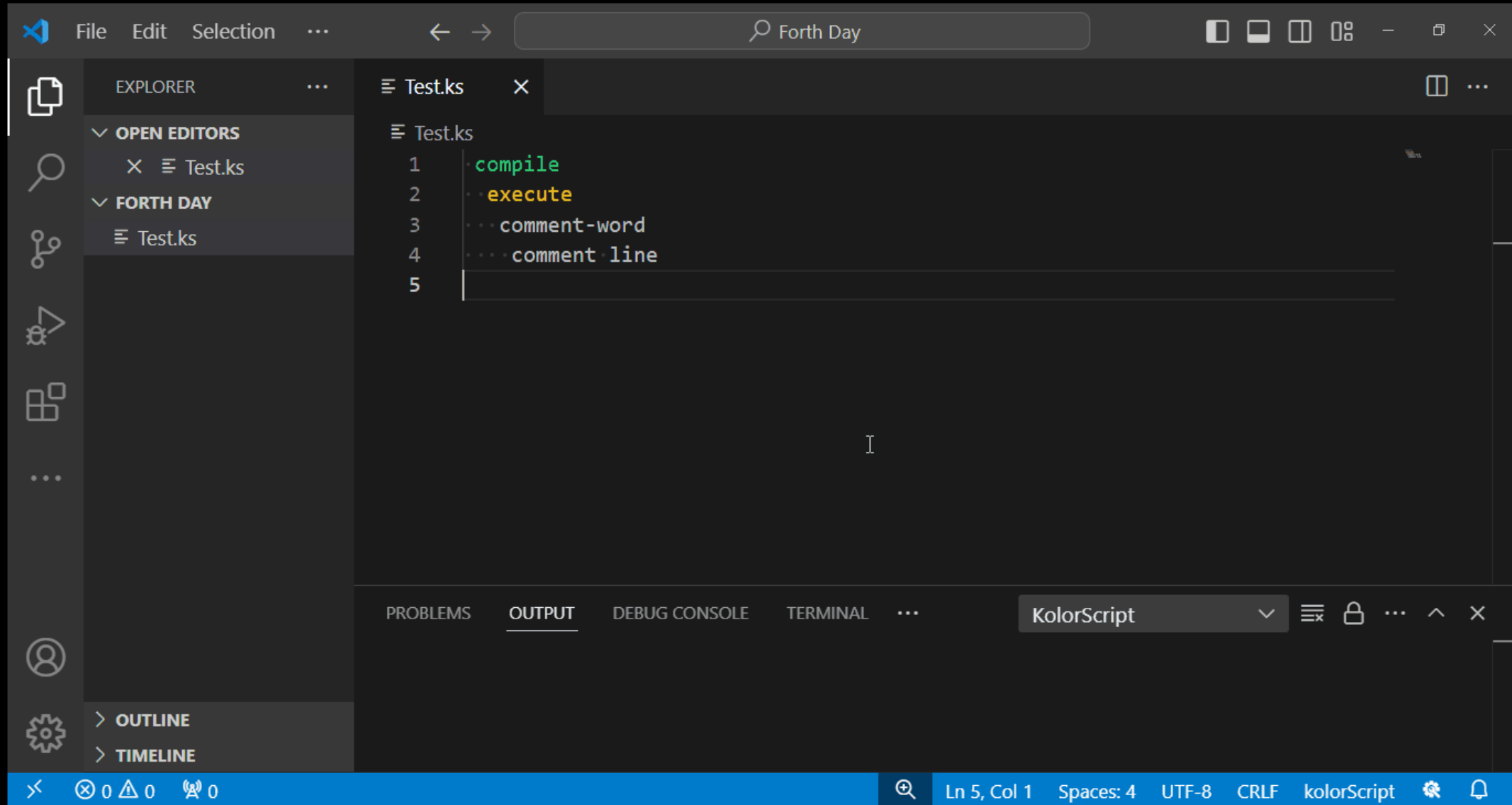
GRAY means “COMMENT out this word(s)”



```
1  compile
2  execute
3  comment-word
4
```

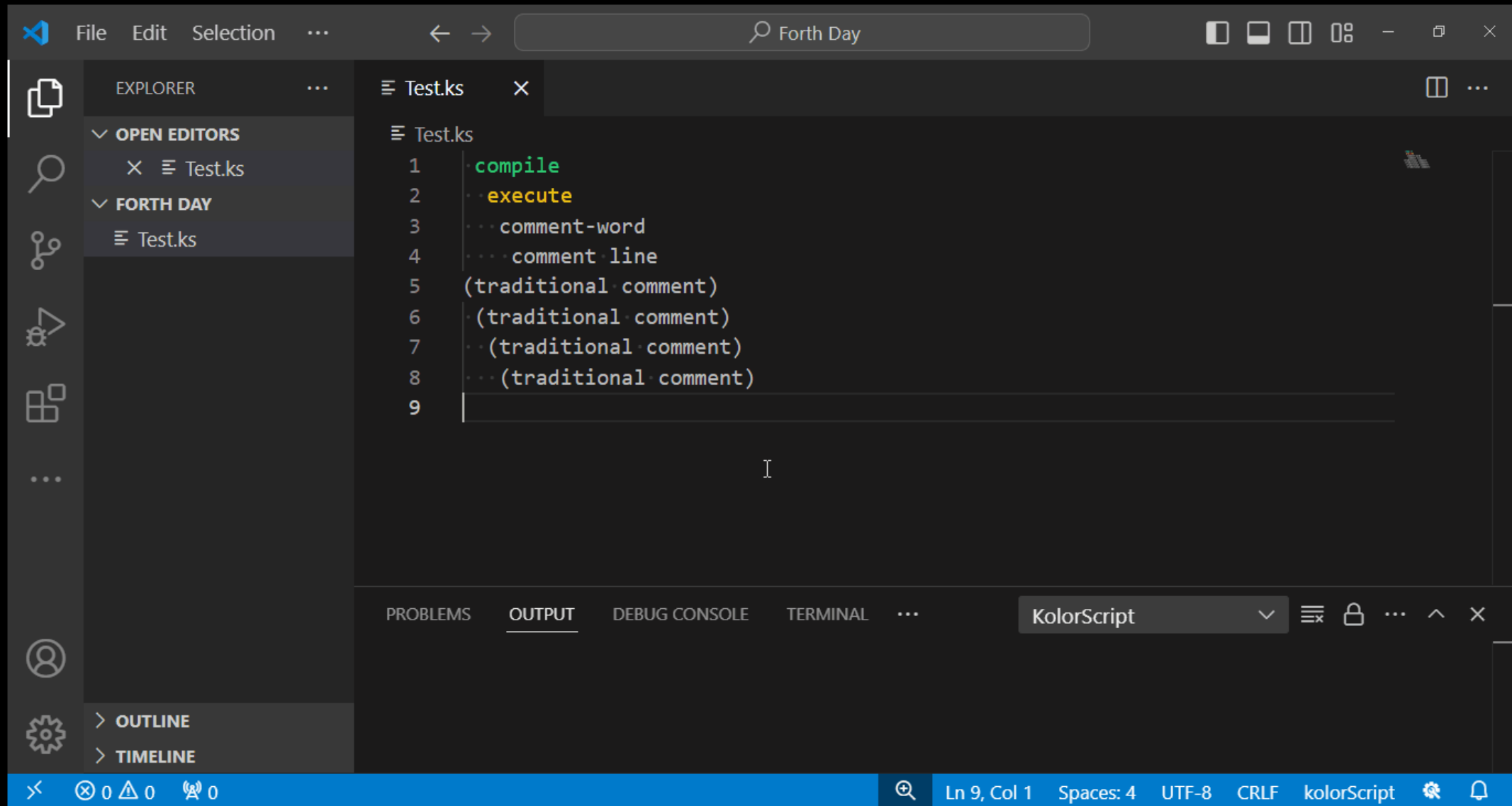
Tag rule: prefix 4 spaces

GRAY means “COMMENT out this word(s)”



Enclosed by parenthesis

BLUE means “define a CONSTANT”

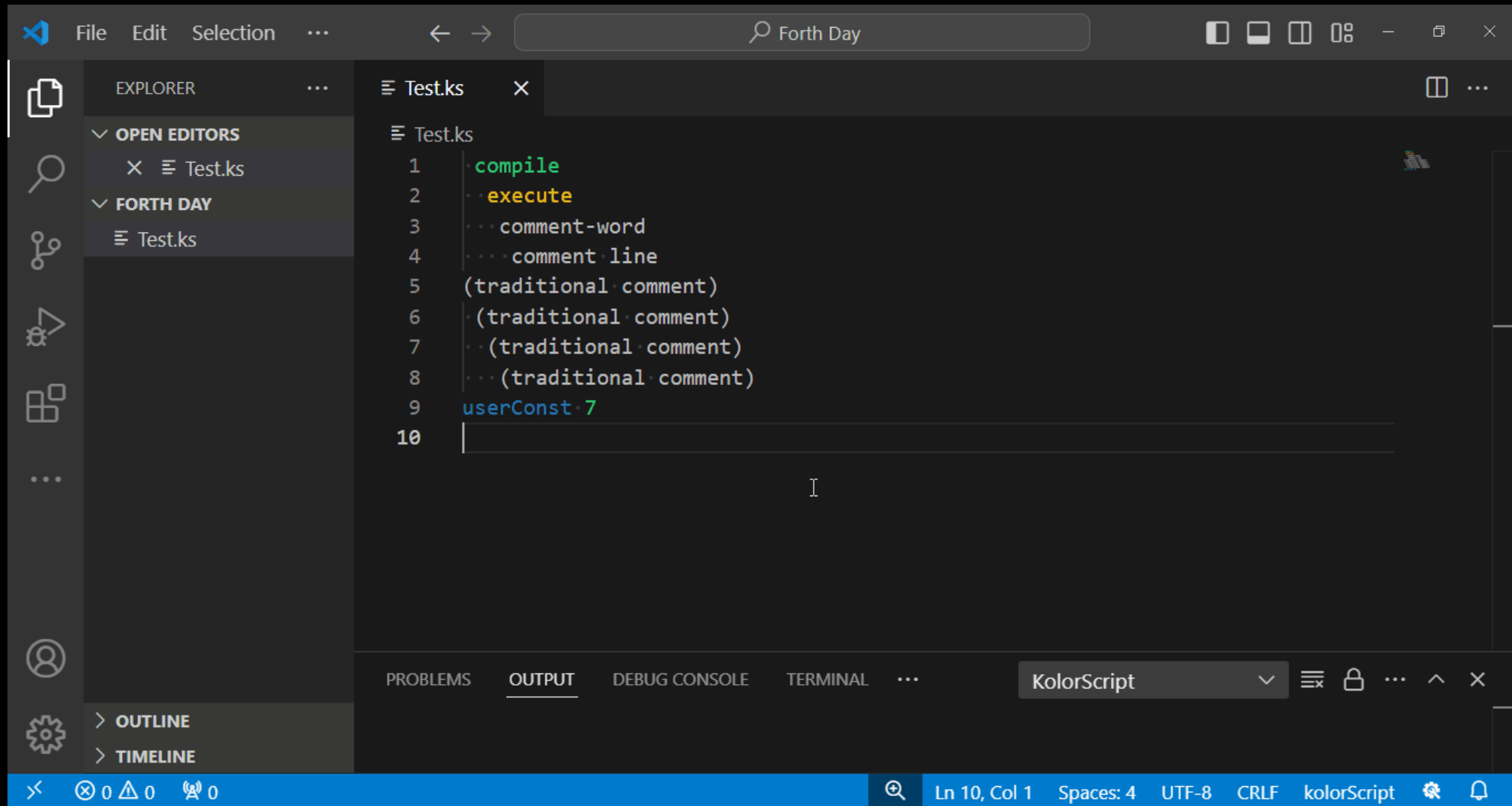


```
1 compile
2 execute
3   comment-word
4   comment line
5 (traditional comment)
6 (traditional comment)
7 (traditional comment)
8 (traditional comment)
9
```

Visual Studio Code interface showing a KolorScript file named `Test.ks`. The code defines a constant `comment-word` with a traditional comment. The status bar at the bottom indicates the cursor is at `Ln 9, Col 1`.

Tag rule: No prefix space AND postfix 1 space

MAGENTA means “define a VARIABLE”

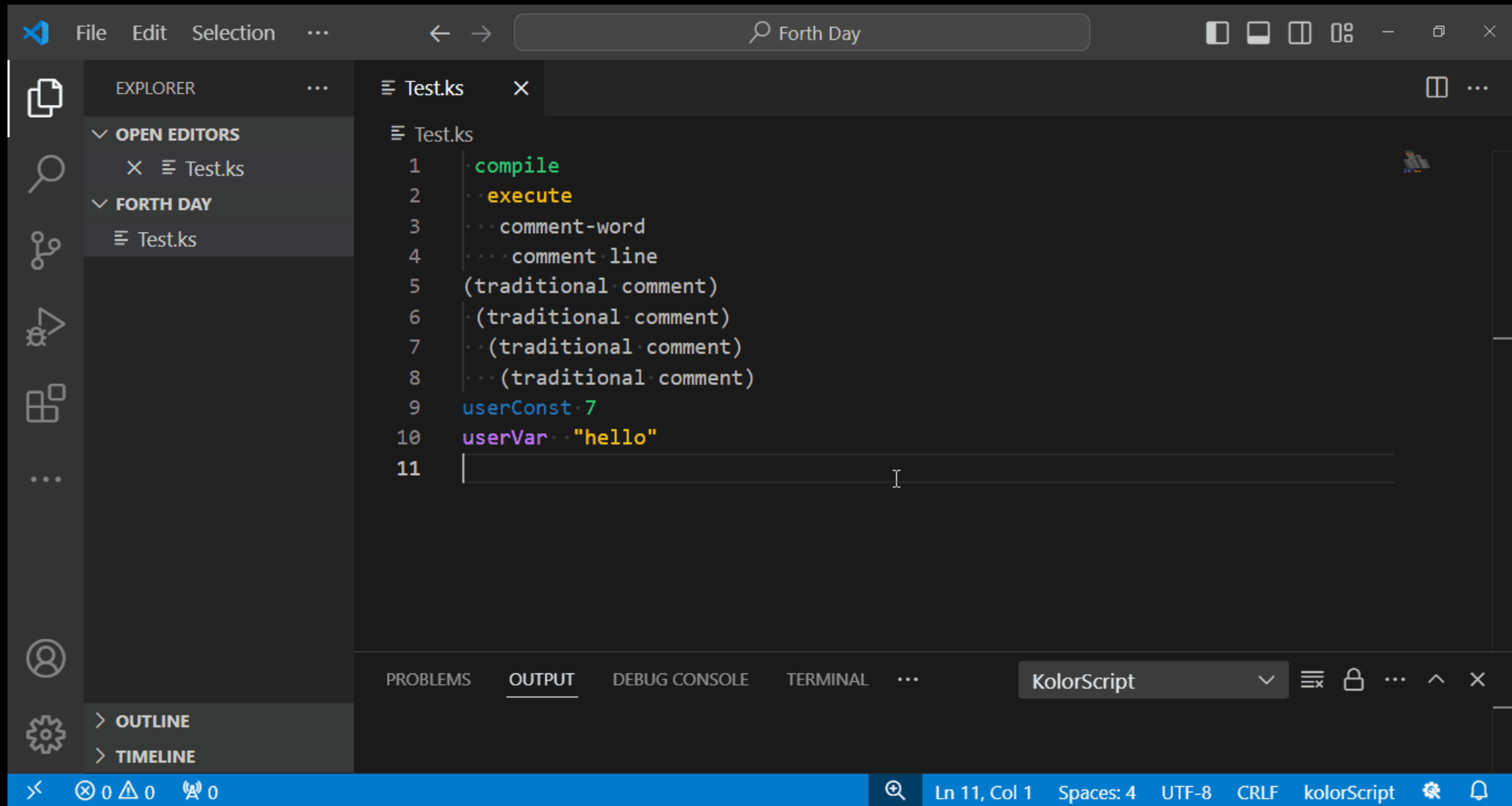


```
1  compile
2  execute
3  ... comment-word
4  ... comment line
5  (traditional comment)
6  (traditional comment)
7  (traditional comment)
8  (traditional comment)
9  userConst 7
10
```

Visual Studio Code interface showing the Explorer, Search, and Run and Debug views. The Explorer view shows the file structure with 'Test.ks' selected. The Search view shows the search results for 'Forth Day'. The Run and Debug view shows the output of the program.

Tag rule: No prefix space AND postfix 2 spaces

RED means “define a FUNCTION”



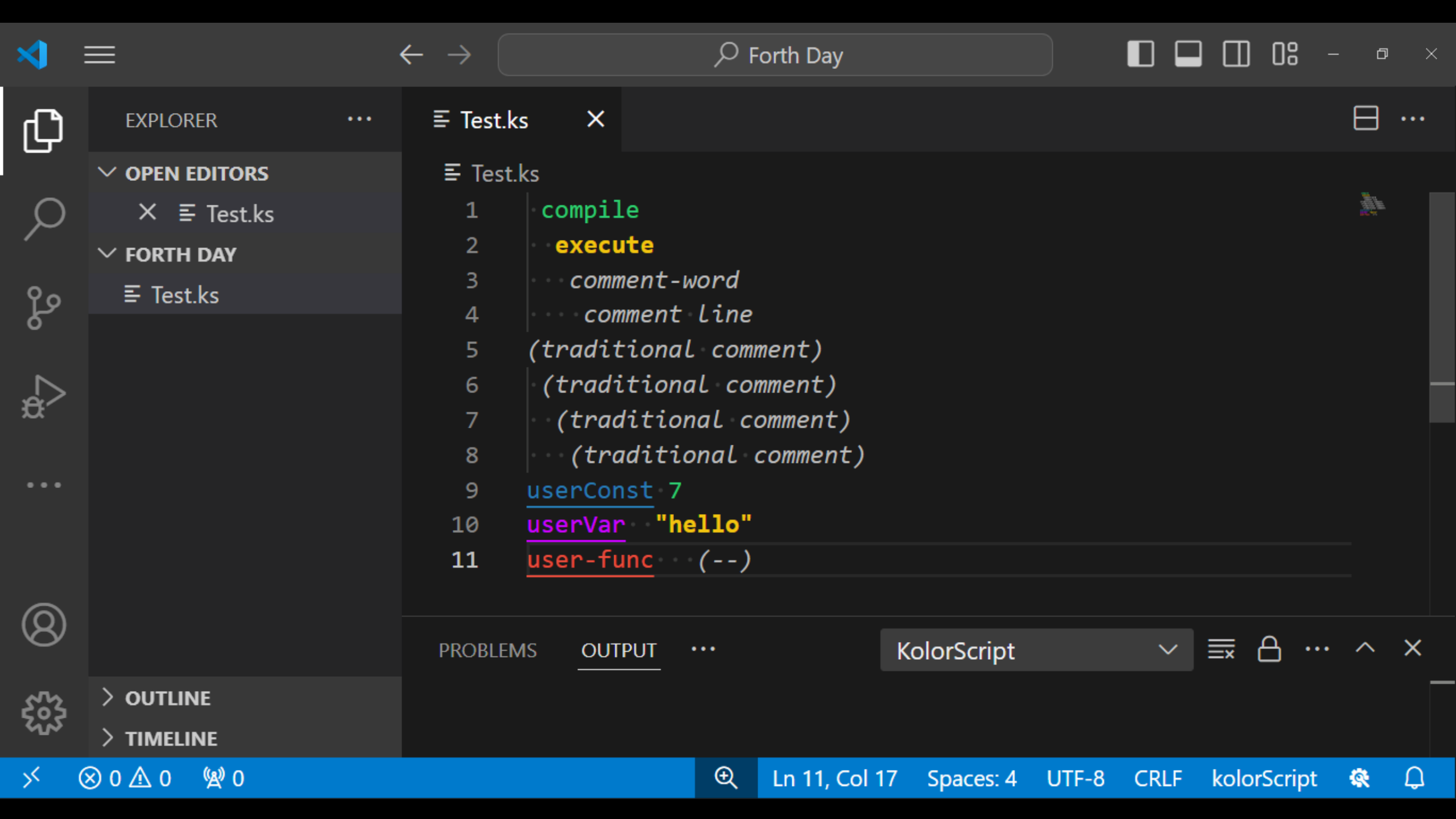
The screenshot shows the Visual Studio Code interface with a file named 'Test.ks' open. The Explorer sidebar on the left shows the file structure with 'OPEN EDITORS' and 'FORTH DAY' sections. The main editor area displays the following code:

```
1  compile
2  execute
3  ... comment-word
4  ... comment line
5  (traditional comment)
6  (traditional comment)
7  (traditional comment)
8  (traditional comment)
9  userConst 7
10 userVar "hello"
11
```

The status bar at the bottom indicates the current position is 'Ln 11, Col 1' with 'Spaces: 4', 'UTF-8', 'CRLF', and 'kolorScript' encoding.

Tag rule: No prefix space AND postfix 3 or 4 spaces

How does **kolorScript** address
color blindness?



Types and States

```
16
17 const Types = {
18   KS_TYPE_NUMBER: 0,
19   KS_TYPE_STRING: 1,
20   KS_TYPE_BUILTIN_FUNC: 2,
21   KS_TYPE_USERDEF_FUNC: 3,
22   KS_TYPE_USERDEF_VAR: 4,
23   KS_TPYE_USERDEF_CONST: 5
24 };
25
26 const States = {
27   KS_STATE_INTERPRET: 0,
28   KS_STATE_COMPILE_FUNC: 1,
29   KS_STATE_COMPILE_VAR: 2,
30   KS_STATE_COMPILE_CONST: 3
31 };
32
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS MEMORY XRTOS SERIAL MONITOR powershell

PS C:\Users\geo\koloscript-lang>

develop* 0 0 0 Ln 25, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript

PROS

- Very simple and very less code
- Codes is sharable using modern repo
- Runs on all major OS with no changes

CONS

- Consumes huge resource due to vscode
- Slow performance due to layer of abstraction
- Color visibility is still within local editor

Currently for **kolorScript**

- Version [1.1.13](#) is available in VS Marketplace
- Stock-Watcher project for John Peters
- Improvements still on going

What's next for **kolorScript**?

- Complete the documentation
- Provide example codes
- Add necessary/requested words
- Debugger
- Standalone VM
- Syntax highlighting on GitHub

Why create **kolorScript**?

- I love **colorForth**!
- I want to use **colorForth** anytime
- I want to access **colorForth** easily
- I want to run **colorForth** code on all major OS
- I want to introduce **colorForth** to non-Forthers
- I want to share **colorForth** codes

Demo time!