Scala

Commentary

A breath of fresh air to find a programming language that has its own website. This being the last language I use in this project, I will conclude that that is the most important aspect of a language. Finding all of the parts I need to write the cipher was found with a few clicks on their website. With every other language I had searched for getting started as my first search, but this was the only language to actually have one. Something I have been spoiled with being mainly a JavaScript developer. The getting started even included installing a VSCode plugin which was very helpful and helped me stay in the editor and not on their beautiful website. I found most of the language intuitive with the main changes just being the keywords. For example, the isUpper function was something that caught my eye in the autocomplete menu. I then checked the built-in documentation and based my program around it. I found that I could do a toUpperCase, and then if the character wasn't uppercase, then the character must not be a letter.

Google Searches

- Scala install
- Scala install ubuntu
- Scala compiler usage – found out you don't need to compile and you can just run as scripting language
- Scala getting started
- Scala for loops
- Scala functions
- Scala void function

Caesar Implementation

```
// run with scala caesar.scala
object Caesar extends App {

    println(encrypt("ATTACK AT ONCE", 4))
    println(decrypt("EXXEGO EX SRGI", 4))

    solve("abcdeFGHIJKLmnopqrstuvwxyz ,?;{[()]}", 26)

    def encrypt(input: String, shift: Int): String = {
        var result = ""
        for(i <- input.toUpperCase) {
            if(!i.isUpper) {
                result += i
            } else {
                val n = (i - 65 + shift) % 26 + 65
                val output = n.toChar
```

```
                result += output
            }
        }
        result
    }

    def decrypt(input: String, shift: Int): String = {
        encrypt(input, 26-shift)
    }

    def solve(input: String, maxShift: Int): Unit = {
        for(i <- 1 to maxShift) {
            val output = decrypt(input, i)
            println(i + "\t" + output)
        }
    }
}
```

Output

```
EXXEGO EX SRGI
ATTACK AT ONCE
1       ZABCDEFGHIJKLMNOPQRSTUVWXY ,?;{[()]}
2       YZABCDEFGHIJKLMNOPQRSTUVWX ,?;{[()]}
3       XYZABCDEFGHIJKLMNOPQRSTUVW ,?;{[()]}
4       WXYZABCDEFGHIJKLMNOPQRSTUV ,?;{[()]}
5       VWXYZABCDEFGHIJKLMNOPQRSTU ,?;{[()]}
6       UVWXYZABCDEFGHIJKLMNOPQRST ,?;{[()]}
7       TUVWXYZABCDEFGHIJKLMNOPQRS ,?;{[()]}
8       STUVWXYZABCDEFGHIJKLMNOPQR ,?;{[()]}
9       RSTUVWXYZABCDEFGHIJKLMNOPQ ,?;{[()]}
10      QRSTUVWXYZABCDEFGHIJKLMNOP ,?;{[()]}
11      PQRSTUVWXYZABCDEFGHIJKLMNO ,?;{[()]}
12      OPQRSTUVWXYZABCDEFGHIJKLMN ,?;{[()]}
13      NOPQRSTUVWXYZABCDEFGHIJKLM ,?;{[()]}
14      MNOPQRSTUVWXYZABCDEFGHIJKL ,?;{[()]}
15      LMNOPQRSTUVWXYZABCDEFGHIJK ,?;{[()]}
16      KLMNOPQRSTUVWXYZABCDEFGHIJ ,?;{[()]}
17      JKLMNOPQRSTUVWXYZABCDEFGHI ,?;{[()]}
18      IJKLMNOPQRSTUVWXYZABCDEFGH ,?;{[()]}
19      HIJKLMNOPQRSTUVWXYZABCDEFG ,?;{[()]}
20      GHIJKLMNOPQRSTUVWXYZABCDEF ,?;{[()]}
21      FGHIJKLMNOPQRSTUVWXYZABCDE ,?;{[()]}
22      EFGHIJKLMNOPQRSTUVWXYZABCD ,?;{[()]}
23      DEFGHIJKLMNOPQRSTUVWXYZABC ,?;{[()]}
24      CDEFGHIJKLMNOPQRSTUVWXYZAB ,?;{[()]}
25      BCDEFGHIJKLMNOPQRSTUVWXYZA ,?;{[()]}
26      ABCDEFGHIJKLMNOPQRSTUVWXYZ ,?;{[()]}
```

Log

Prediction: 2 hours

Date	Hours Spent	Accomplishments
2/16	0.5	Set up dev env, including VSCode plugin
2/16	0.5	Write encrypt, decrypt, and solve functions
2/21	0.5	Test thoroughly

Discrepancy of time

Development was quicker than expected here because of two main reasons, first, this is my 5th time writing the same program, and two, the tooling for Scala is next level. It's equivalent to what I get in my IDE from Python or JS and this rapidly increased my development. Being able to point out errors before I run the program was a huge plus.

Overall Review

Scala does a good job at being both readable and writable. It makes good changes overall to Java and follows a similar structure to today's most popular languages. I think def is my favorite keyword for functions, but val and var are too similar, minus points for readability, plus points for writability as it is just a single character. The for loops and function signatures are very readable and writable. They let the programmer know at a glance what the function does by including the type of all the variables at the top. The arrow for loop makes a lot of sense, as the value from the array on the right is being inserted to the variable, really like this feature. I like how scala handles "returning." This was one of the few things I like from the language R, and I don’t like much from that language. Overall, very readable and writable, thanks for making me use this language.

Ratings

Readability: 9/10

Writability: 8/10

Ranking: 1st

Final Rankings

1. Scala
2. Basic
3. Pascal
4. Fortran
5. COBOL