

Erlang

Commentary

I began this language how I start all languages, look up their getting started page. Despite having a getting started page, it is very unhelpful on its own. It almost appears to be a broken link. It's clear the people that made this language were not big on web development, one of the reasons JS documentation is so plentiful online. Another quirky thing about the language is that you can't really run a file the same way you can in LISP and ML, you must first start a shell in the same directory as a .erl file, then compile, then run the functions that you choose to expose in the file. Very strange design, not sure I completely understand why that choice was made either. One thing that was great, it had the easiest install of all the languages as ubuntu's apt package manager had the erlang package officially supported.

Google Searches

- erlang getting started
- erlang ubuntu
- erlang run file
- erlang examples
- erlang get char code
- erlang from char code
- erlang if
- erlang map
- erlang map over x times
- erlang foreach

Caesar Implementation

```
-module(caesar).
-export([encrypt/2, decrypt/2, solve/2]).

rot(Ltr, Shift) when hd(Ltr) >= 65, hd(Ltr) <= 90 ->
    (hd(Ltr) - 65 + Shift) rem 26 + 65;
rot(Ltr, _) ->
    hd(Ltr).

encrypt(Word, Shift) ->
    lists:map(fun(Ltr) -> rot([string:to_upper(Ltr)], Shift) end, Word).

decrypt(Word, Shift) ->
    encrypt(Word, 26 - Shift).

solve(Word, Shift) ->
    lists:foreach(
        fun(X) -> io:format("Shift: ~p\tResult: ~p ~n", [X, decrypt(Word,
X)]) end,
```

```
lists:seq(1, Shift)
).
```

Output

```
3> caesar:encrypt("ATTACK AT ONCE", 4).
"EXXEGO EX SRGI"
4> caesar:decrypt("EXXEGO EX SRGI", 4).
"ATTACK AT ONCE"
5> caesar:solve("abcdeFGHIJKLmnopqrstuvwxyz ,?;{[( )]}", 26).
Shift: 1      Result: "ZABCDEFGHIJKLMNopQRSTUVWXYZ ,?;{[( )]}"
Shift: 2      Result: "YZABCDEFGHIJKLMNopQRSTUVWX ,?;{[( )]}"
Shift: 3      Result: "XYZABCDEFGHIJKLMNopQRSTUVW ,?;{[( )]}"
Shift: 4      Result: "WXYABCDEFGHIJKLMNopQRSTUV ,?;{[( )]}"
Shift: 5      Result: "VWXYZABCDEFGHIJKLMNopQRSTU ,?;{[( )]}"
Shift: 6      Result: "UVWXYZABCDEFGHIJKLMNopQRST ,?;{[( )]}"
Shift: 7      Result: "TUVWXYZABCDEFGHIJKLMNopQRS ,?;{[( )]}"
Shift: 8      Result: "STUVWXYZABCDEFGHIJKLMNopQR ,?;{[( )]}"
Shift: 9      Result: "RSTUVWXYZABCDEFGHIJKLMNopQ ,?;{[( )]}"
Shift: 10     Result: "QRSTUVWXYZABCDEFGHIJKLMNop ,?;{[( )]}"
Shift: 11     Result: "PQRSTUVWXYZABCDEFGHIJKLMNO ,?;{[( )]}"
Shift: 12     Result: "OPQRSTUVWXYZABCDEFGHIJKLMN ,?;{[( )]}"
Shift: 13     Result: "NOPQRSTUVWXYZABCDEFGHIJKLM ,?;{[( )]}"
Shift: 14     Result: "MNOPQRSTUVWXYZABCDEFGHIJKL ,?;{[( )]}"
Shift: 15     Result: "LMNOPQRSTUVWXYZABCDEFGHIJK ,?;{[( )]}"
Shift: 16     Result: "KLMNOPQRSTUVWXYZABCDEFGHIJ ,?;{[( )]}"
Shift: 17     Result: "JKLMNOPQRSTUVWXYZABCDEFGHI ,?;{[( )]}"
Shift: 18     Result: "IJKLMNOPQRSTUVWXYZABCDEFGH ,?;{[( )]}"
Shift: 19     Result: "HIJKLMNOPQRSTUVWXYZABCDEFG ,?;{[( )]}"
Shift: 20     Result: "GHIJKLMNOPQRSTUVWXYZABCDEF ,?;{[( )]}"
Shift: 21     Result: "FGHIJKLMNOPQRSTUVWXYZABCDE ,?;{[( )]}"
Shift: 22     Result: "EFGHIJKLMNOPQRSTUVWXYZABCD ,?;{[( )]}"
Shift: 23     Result: "DEFGHIJKLMNOPQRSTUVWXYZABC ,?;{[( )]}"
Shift: 24     Result: "CDEFGHIJKLMNOPQRSTUVWXYZAB ,?;{[( )]}"
Shift: 25     Result: "BCDEFGHIJKLMNOPQRSTUVWXYZA ,?;{[( )]}"
Shift: 26     Result: "ABCDEFGHIJKLMNOPQRSTUVWXYZ ,?;{[( )]}"
```

Log

Estimate: 2 hours

Date	Hours Spent	Accomplishments
4/6	.1	Set up erlang shell and run hello world
4/6	.5	Create rot function
4/6	.5	Create encrypt/decrypt
4/6	.25	Create solve function

Date	Hours Spent	Accomplishments
4/13	.25	Fix for special characters

Discrepancy of time

I basically take the same amount of time for each of these languages that I haven't used before. Therefore, I've gotten quite good at guessing the time it takes me.

Overall Review

In general, Erlang is a pretty good language. I would never use it though because I think that functional programming is mostly a novelty, not for real world applications. I would find it highly unlikely to find any company start a project today in erlang. There are a lot fo quirks to the language which add to its cost, specifically, not using and, not really making it obvious how characters and strings work, for some reason the head of a character is the character code??? and the way the shell works. The readability is pretty good, especially with ligatures. I did notice that they do the `>=` `=<` comparison operators wrong. Very confusing... should be equal on the right in both cases, they broke my ligatures... Takes away from the writability.

Ratings

Readability: 7/10

Writability: 6/10

Ranking: 3/5