

# Haskell

# Simple Recursion

2021-04-16

# Agenda

Using Haskell

GHCI

Patterns

Recursion

PA1 tips



# Haskell

- Functional programming language
- Equational reasoning
- NO side effects (e.g., write to file, DB, internet)
  - Unless you explicitly handle them
- Weird syntax

# Function Application

Elsa:  $\text{ITE } A \ B \ C = (((\text{ITE } A) \ B) \ C)$

Haskell:  $\text{ite } a \ b \ c = (((\text{ite } a) \ b) \ c)$

LEFT ASSOCIATIVE!

# Quiz

In Haskell, like Elsa, which is the correct parenthesization of:

`fold add 0 empty`

- A. `(fold (add 0 empty))`
- B. `(fold (add (0 empty)))`
- C. `(fold add) (0 empty)`
- D. `((fold add) 0) empty)`
- E. `????`

# Quiz

In Haskell, like Elsa, which is the correct parenthesization of:

```
fold add 0 empty
```

- A. `(fold (add 0 empty))`
- B. `(fold (add (0 empty))`
- C. `(fold add) (0 empty)`
- D. `((fold add) 0) empty) -- left associative`
- E. `????`

# GHCI

Glorious Glasgow Haskell Compiler Interpreter

- `$ make ghci`

Issue?: The following GHC options are incompatible with GHCi and have not been passed to it: `-threaded`

Configuring GHCi with the following packages: `hw1-haskell`

`/tmp/haskell-stack-ghci/74924166: createDirectory: permission denied`  
(Permission denied)

Fix: `$ TMPDIR=$HOME/.tmp make ghci`

# Haskell

learnyouahaskell.com



Hey yo! This is **Learn You a Haskell**, the funkiest way to learn Haskell, which is the best functional programming language around. You may have heard of it. This guide is meant for people who have programmed already, but have yet to try functional programming.

The whole thing is completely free to read online, but it's also available in print and I encourage you to buy as many copies as you can afford!

To contact me, shoot me an email to: [bonus at learmyouahaskell dot com](mailto:bonus@learnyouahaskell.com)! You can also find me idling on [#haskell](#) where I go by the name **BONUS**.

CLICK ↴  
**Buy it!**  
You know you want to!

**Read it online!**  
For free!

Got questions? READ THE **FAQ**

Holy shit!