



fraktell

Fractal Visualization in Haskell

By Riley Weber

`github.com/rileyweber13/fraktell`

Haskell

- Functional
- Closely tied to theory
- Named after Haskell Curry
- lambda calculus theory:

Construct	Lambda Calculus Notation	Algebraic Notation
Parameter	x	x
Function	$(\lambda x.M)$	$f()$
Application	$(M\ N)$	$f(x)$

Haskell - example

```
triple :: Int -> Int  
triple x = 3 * x
```

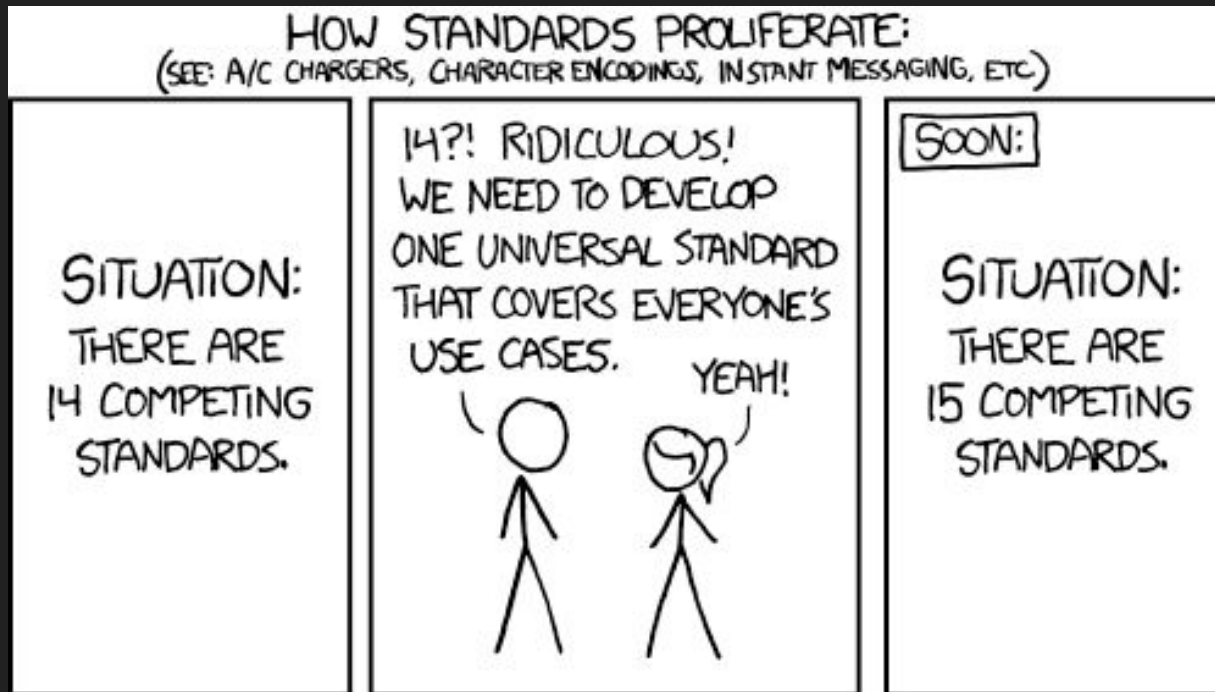
Usage: "triple 6" will produce "18"

Haskell - differences with other functional languages

- Close ties to math
- Dedication to functional purity
 - Function with side effects are marked:
`getArgs :: IO [String]`
- Good software tools
 - Stack
 - Hackage

Haskell - history

- Primary goal: create one universal functional language [1]

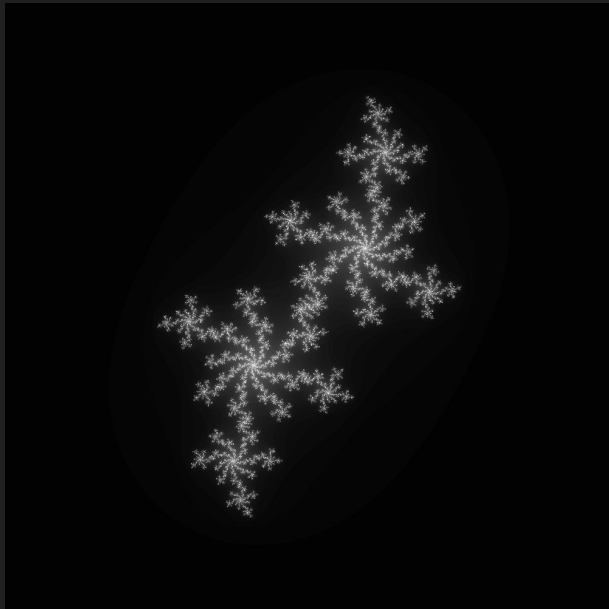


Haskell - history

- Except... it worked.
- Effectively replaced everything but Lisp [2]
- Other goals [1]
 - Be a useful teaching tool
 - Be suitable for real-world projects
 - Should be comprehensively described

What was useful? Why should you use it?

- Once your code compiles, it works more frequently than with imperative languages



What was useful? Why should you use it?

- Fewer runtime errors
- Tests were incredibly easy to write
- Easily supply functions as arguments
 - `"makeImageR :: ... g"`
 - `"julia :: f"`
- Lazy evaluation
 - `"func_num = read (args!!6) :: Int"`
- Partial evaluation of functions
 - `f1 c z = z**2 + c:`
 - Apply c only:
 - `f2 = f1 3`
 - `f2 7 == f1 3 7`

Highly Suitable for:

- Mathematical, scientific computing
- When correctness is very important
- When you can easily use pure functions
 - (this is easily parallelizable)

Not Well Suited for:

- Stated-based systems
 - MPI
 - TCP
- Anything networking, really
 - (I/O is a pain!)
- People who don't like reading documentation

References

1. S. P. Jones, Haskell 98 language and libraries: the revised report. Cambridge: Cambridge University Press, 2006.
2. “TIOBE Index for November 2019,” TIOBE. [Online]. Available: <https://www.tiobe.com/tiobe-index/>. [Accessed: 18-Nov-2019].