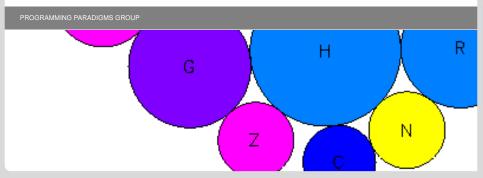




#### A Haskell Roadshow

Joachim Breitner
The Karlsruhe Functional Programmers Meetup Group
December 18, 2012



#### Haskell is know for



being a

functional pure lazy evaluated strongly typed interpreted compiled

programming language . . .

#### Haskell is know for



being a

functional pure lazy evaluated strongly typed interpreted compiled

programming language ...

#### Lets demonstrate that



# Visualize (one aspect) of this data:

/dev/fd/63	Wed	Dec	1	L9	09:47:49	2012	2
mea not allow transcript		ACA 10 0					men and above supplies
mes are store partiety		men no e					
men an about parties;		A14 10 0					data for about patricty
men are show postery		A14 10 0					data but about patricty
mes are store patterny		men ner e					data not about patractly
men an about patricky		200 00 0					men on ston paracey
men an about patricty		A14 14 4					data but about patracty
men are show postery		A14 10 0					data but about patricty
mes not show patterny		Mark his o					data but about patracty
men an about patricky		200 00 0					men on ston paracey
mes are store patterny		men no e					data not about patractly
man an about patricty		ACA 100 0					data but about patracty
men an about patricty		A14 14 4					data but about patracty
men an about patricty		A14 14 4					data but about patracty
men art store parties		M14 10 0					data for about patricty
mes are store partiety		men no e					
man an about patricty		ACA 100 0					data but about patracty
men an about patricty		A14 14 4					data but about patracty
men an about parties;		A14 10 0					data for about patricty
		-					
men are show postery		A14 10 0					data but about patricty
mes not street particip		men no e					mera non amoun patracry
Mark Set Wash (MINISTY		200.00					men on ston paracry
mes not strain particip		men no e					mera non amoun patracry
Mark and whose publishing		A14 147 4					data but about patracty
men an about parties;		A14 10 0					data for about patricty
men are show postery		A14 10 0					data for about patricty
men art shan partity		Mark to a					data but about patracty
men an about patricky		200 00 0					men on ston paracey
men art shan partity		Mark to a					data but about patracty
man an about patricty		ACA 100 0					data but about patracty
men an about parties;		A14 10 0					data for about patricty
men are show postery		A14 10 0					data but about patricty
							data but about patracty
men are show postery		A14 14 4					data but about patricty
mes are store patterny		men ner e	non j	passery			data not about patractly
men an about patricky		200 00 0					men on ston paracey
may be sharp parties.		A14 10 0	moun )				data but about patracty
Mark Set Wash (1981-19)		A14 14 4					data for about patricty
men an about patricty		A14 10 0					data for about patricty
men an about parties		MATE DE A	non I				data but about patricts

# Live demonstration



Haskell is indeed

functional?
pure
lazy evaluated
strongly typed
interpreted
compiled

programming language ...



Haskell is indeed

functional 
 pure ?
lazy evaluated
strongly typed
interpreted
compiled

programming language ...



Haskell is indeed

functional √
pure √
lazy evaluated ?
strongly typed
interpreted
compiled

programming language ...



Haskell is indeed

```
functional √
pure √
lazy evaluated √
strongly typed ?
interpreted
compiled
```

programming language ...



Haskell is indeed

```
functional √
pure √
lazy evaluated √
strongly typed √
interpreted ?
compiled
```

programming language ...



Haskell is indeed

```
functional √
pure √
lazy evaluated √
strongly typed √
interpreted √
compiled ?
```

programming language ...



Haskell is indeed

```
functional √
pure √
lazy evaluated √
strongly typed √
interpreted √
compiled √
```

programming language ...

# What we skipped today



# All the small things...

- More about data types
- (Many) more benefits from the type system
- Polymorphism
- Type classes
- Monads
- Foreign Function Interface

# What we skipped today



# All the small things...

- More about data types
- (Many) more benefits from the type system
- Polymorphism
- Type classes
- Monads
- Foreign Function Interface

# ... you will find here

- Tutorial "Learn you a Haskell"
- O'Reilly book "Real World Haskell"
- Tutorial "Write Yourself a Scheme in 48 Hours"

#### Conclusion



# Writing Haskell code

- takes less time,
- produces less bugs and
- is more fun.

Therefore, CU all on

#haskell on IRC (freenode)

and on the

haskell-cafe@haskell.org mailing list!

© 2012 Joachim Breitner.

Distributed under the terms of the Creative Commons Attribution license.