BlazeHtml

A DSL for HTML generation in Haskell

Hello!

My name is jasper

Studying BSc CS @ UGent

I like to make things

@jaspervdj

jaspervdj.be

Introduction

A web app usually has 3 important layers:

- web application server
- data storage layer
- html generation layer

Overview

- Immutability
- Design patterns
- eDSL's
- Benchmarks

Imperative HTML

```
Tag b = new Tag("body");
Tag main = new Tag("div");
b.add(main);
Tag h2 = new Tag("h2");
h2.add("Introduction");
main.add(h2);
```


IMMUTABLE KITTEH
DOES NOT MOVE

Trees in Haskell

```
data Tree
    = Node Tree Tree
    | Empty
```

Immutability

Immutable data:

```
f :: Tree
  -> Tree
  -> Tree
```

- f will not modify the input
- Create a new value

Efficiency?

f joins trees:

```
f x y = Node x y
```

Even shorter definition:

```
f = Node
```


style
title
div
head
body
html

Design Patterns

Design patterns are mostly based on concepts from Category Theory.

Category Theory, n.: general abstract nonsense

Monoids

```
class Monoid a where
  mempty :: a
  mappend :: a -> a -> a
```

Can be anything from computations to lists.

Monoids

This allows use of standard functions:

```
ul $ mconcat $ map
  (li . string)
  ["one", "two", "three"]
```

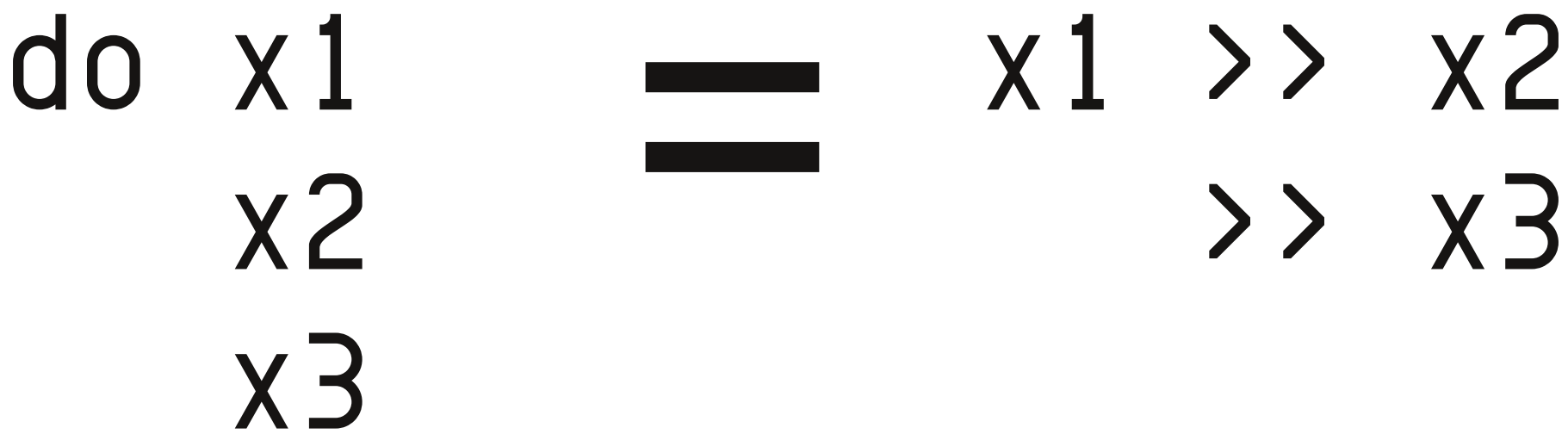
An eDSL for HTML

Use do notation in Haskell

```
x1 = do
   x2 <- f1
   x2 foo
   x3 bar
```

An eDSL for HTML

Do notation: syntactic sugar for monad operations.



Example

```
html $ do
  head $ title "Foo"
  body $ do
    p "Paragraph 1"
    p "Paragraph 2"
```

Example

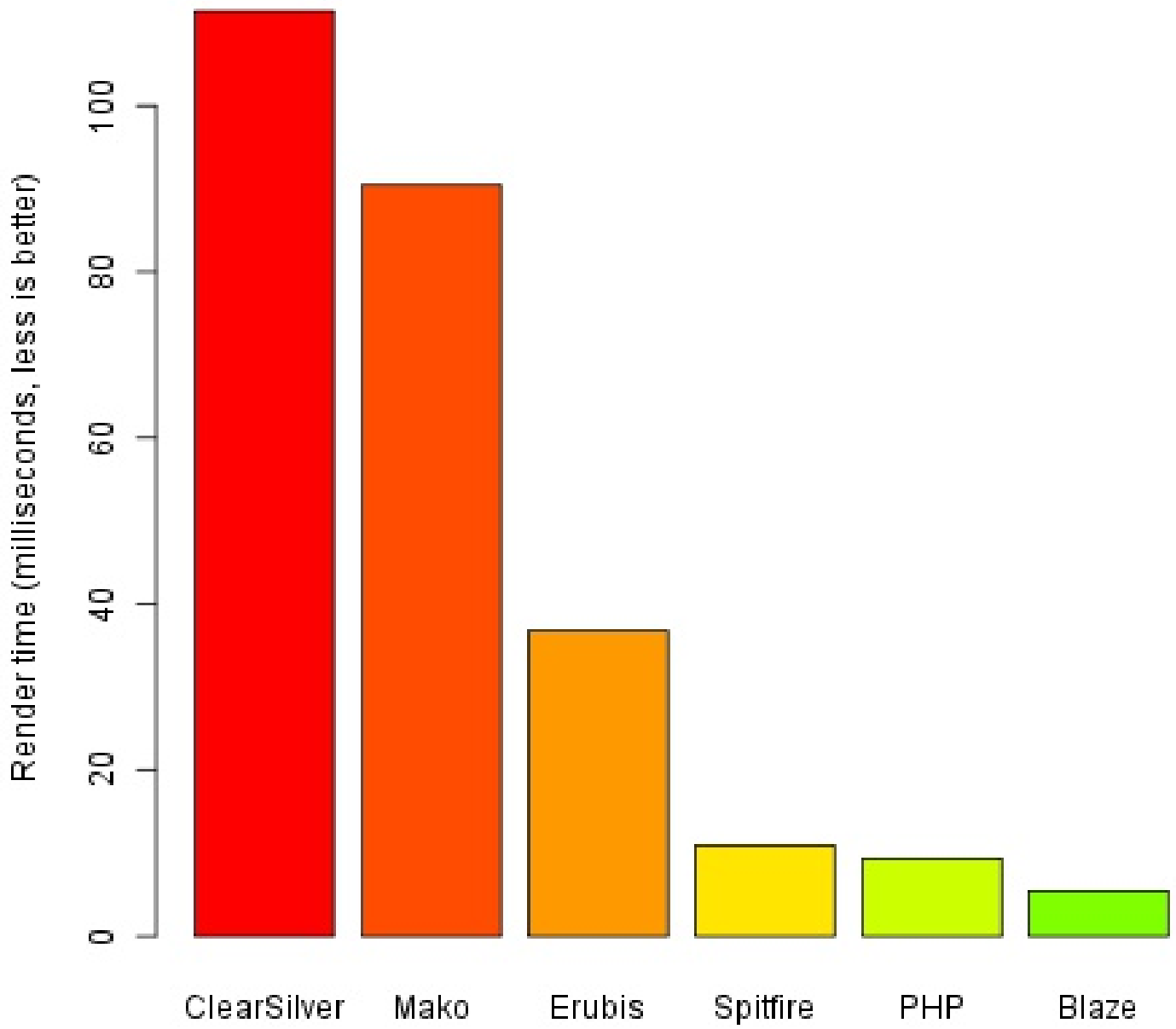
```
includeJS source = script
  ! type "text/javascript"
  ! src source
  $ mempty

includeJS "jquery.min.js"
```

Results

- Validity of HTML
- Compiled (fast)
- Easy syntax

Benchmarks


Render time (milliseconds, less is better)
0
20
40
60
80
100
ClearSilver
Mako
Erubis
Spitfire
PHP
Blaze


Nope, a monad
is just a monoid
in the category
of endofunctors.
Purely Functional
Data Structures
Chris Okasaki
Next question
please.