









- For developers
- For management
- For company

Developers

Compiler support



Advanced type system




```
data USD
data EUR
```

```
newtype Currency a = Currency Double deriving Show
```

```
add :: Currency a → Currency a → Currency a
add (Currency a) (Currency b) = Currency $ a + b
```

```
account :: Currency USD
```

```
account = Currency 5.0
```

```
transaction1 :: Currency USD
```

```
transaction1 = Currency 5.0
```

```
transaction2 :: Currency EUR
```

```
transaction2 = Currency 5.0
```

```
main = do
```

```
    print $ add account transaction1 - Currency 10.0
```

```
    print $ add account transaction2 - won't compile
```


```
$ ./currency_test.hs
```

```
currency_test.hs:23:25: error:
```

- Couldn't match type 'EUR' with 'USD'
Expected type: Currency USD
Actual type: Currency EUR
- In the second argument of 'add',
namely 'transaction2'
In the second argument of '(\$)',
namely 'add account transaction2'
In a stmt of a 'do' block:
print \$ add account transaction2

Computations are more explicit

```
1 gtod = scd->tick_gtod + __gtod_offset;  
2 clock = gtod + delta;  
3 min_clock = wrap_max(gtod, old_clock);  
4 max_clock = wrap_max(old_clock, gtod + TICK_NSEC);
```



Explicit computation patterns

Referential transparency (a.k.a. refactoring without fear)

Immutability (a.k.a. you can trust this function)

Explicit handling of side effects

Tooling support, great ecosystem

Remarkable performance

Do you need a parser?

Management

High quality code (less bugs, easy to maintain and to learn)

Awesome for agile development (because of referential transparency, easy to support and refactor large code base and create dsl)

Heavy boost for skills development inside your team

Rumored to be much cost efficient (Aaron Contorer)

You can attract a lot of talented people

Company

A lot of big companies are investing in this


Contributions is much more visible - not one of
thousands nameless companies with java
technology stack, you have a ~~dragon~~ Haskell in
production

Great anchor for development of technical culture
(CS and stuff)

Questions?

 github.com/erthalion

 @erthalion

 9erthalion6 at gmail dot com