Y-Not

Jim Weirich / Chief Scientist / { New Context }

Movtivating Talk ...

- Non-Technical
-
-

Movtivating Talk ...

- ~~Non-Technical~~ Highly Technical
-
-

Movtivating Talk ...

- ~~Non-Technical~~ Highly Technical
- Relevent
-

Movtivating Talk ...

- ~~Non-Technical~~ Highly Technical
- ~~Relevent~~ Extremely Pointless
-

Movtivating Talk ...

- ~~Non-Technical~~ Highly Technical
- ~~Relevent~~ Extremely Pointless
- Good

Movtivating Talk ...

- ~~Non-Technical~~ Highly Technical
- ~~Relevent~~ Extremely Pointless
- ~~Good~~ Worst Ruby Code Ever

Keynote Talk ...

- ~~Non-Technical~~ Highly Technical
- ~~Relevent~~ Extremely Pointless
- ~~Good~~ Worst Ruby Code Ever

FUN!

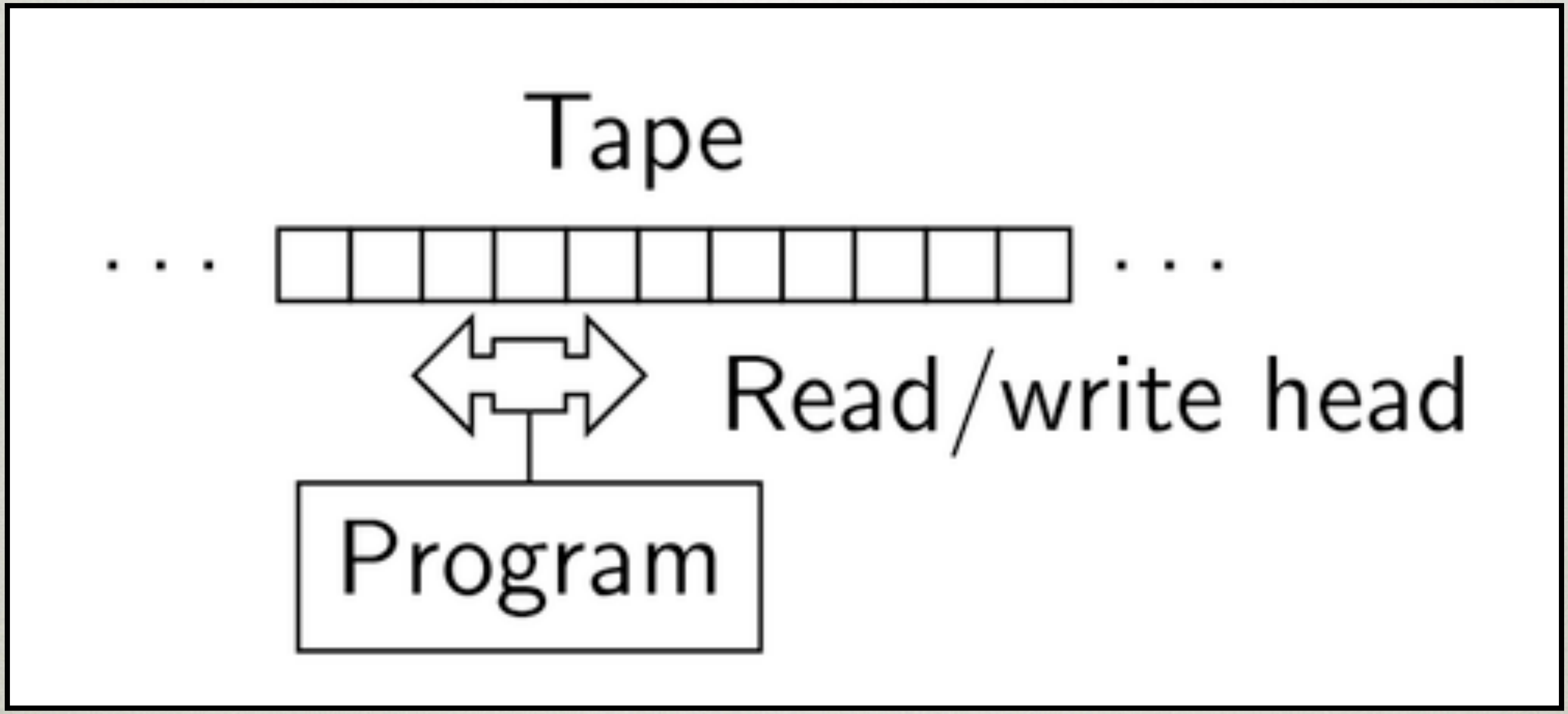
Audience Participation

Cos(n)

Effectively Computable

Alan Turing
1912 - 1954

Universal Turing Machine



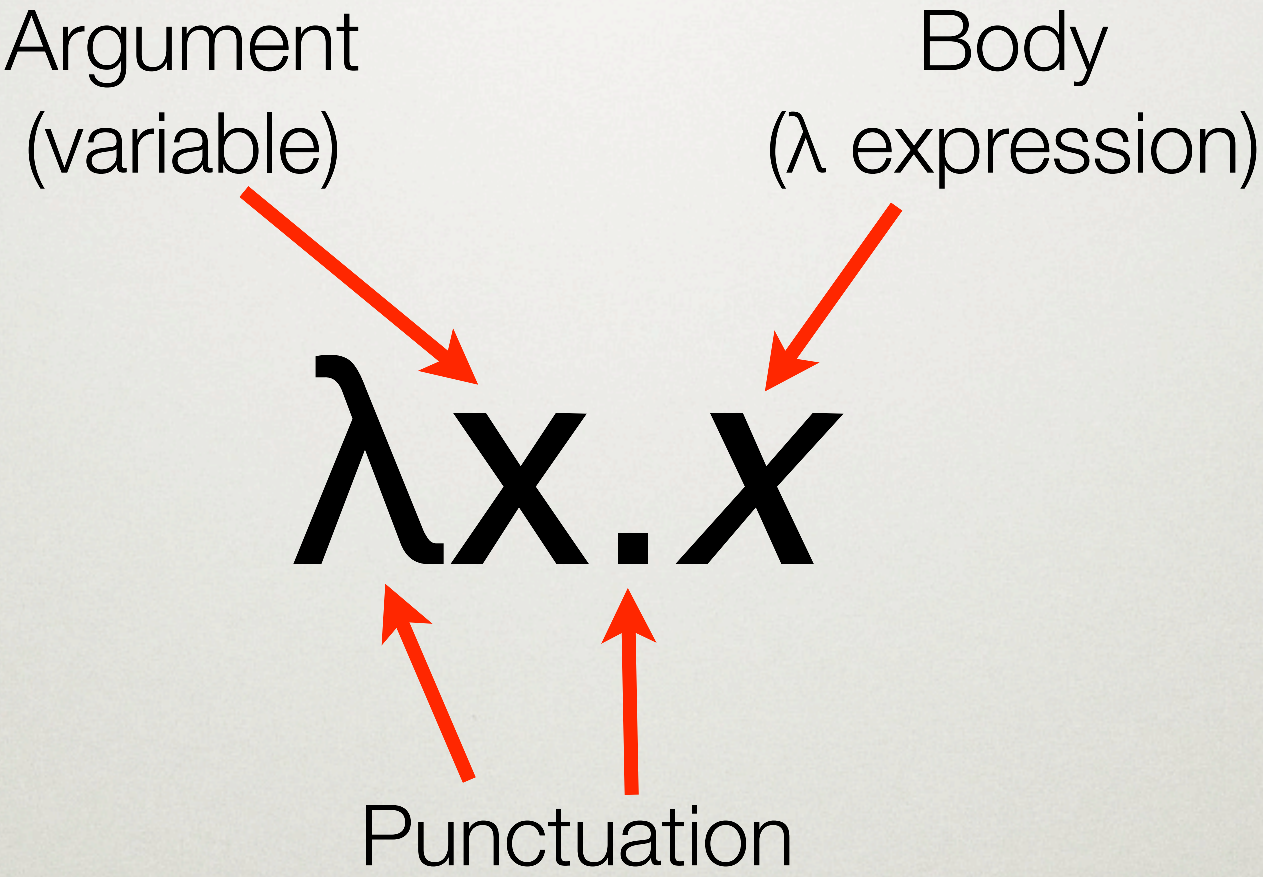
Turing Thesis

Anything that is "effectively computable" can be computed by a Universal Turing Machine.

1936-7

Alonzo Church
1903-1995

λ Calculus



Argument (variable)

Body (λ expression)

λx.x

Punctuation

λ Calculus

Zero	$\lambda f.\lambda x.x$
One	$\lambda f.\lambda x.\mathtt{f}\ x$
Two	$\lambda f.\lambda x.\mathtt{f}\ (\mathtt{f}\ x)$
True	$\lambda x.\lambda y.x$
False	$\lambda x.\lambda y.y$

λ Calculus

- Functions are the only data type
- λ binding is the only way to associate values to variables
- Calculation happens via Beta (or Alpha) reduction

Beta Reduction

TRUE ONE ZERO

$$((\lambda x.\lambda y.x)(\lambda f.\lambda x.\mathtt{f}\ x))(\lambda f.\lambda x.x)$$

ONE ZERO

$$(\lambda y.(\lambda f.\lambda x.\mathtt{f}\ x))(\lambda f.\lambda x.x)$$

ONE

$$(\lambda f.\lambda x.\mathtt{f}\ x)$$

Church Thesis

Anything that is
"effectively computable"
can be computed by via
λ Calculus.

1933, 1935

Influences Today

λ Calculus Legacy

$$\lambda f.\lambda x.\mathtt{f}\ x$$

Lisp

```
(lambda (f) (lambda (x) (f x))
```

Clojure

```
(fn [f] (fn [x] (f x) )
```

Ruby

```
lambda { |f| lambda { |x| f[x] } }
->(f) { ->(x) { f.(x) } }
```

CoffeeScript

```
(f) -> (x) -> f(x)
```

Javascript

```
function(f) { return function(x) { return f(x) } }
```

Fixpoints

Prediction ...

0.739085133215161

$$
\begin{aligned}
\cos(0) &= 1.00000000000000 \\
\cos(\cos(0)) &= 0.54030230586814 \\
\cos(\cos(\cos(0))) &= 0.85755321584639 \\
\cos^{4}(0) &= 0.65428979049777 \\
\cos^{5}(0) &= 0.79348035874256 \\
\cos^{10}(0) &= 0.73140404242251 \\
\cos^{80}(0) &= 0.73908513321515 \\
\cos^{89}(0) &= 0.73908513321516 \\
\cos^{90}(0) &= 0.73908513321516 \\
\cos^{91}(0) &= 0.73908513321516
\end{aligned}
$$

$$n_{\text{fixpoint}} = 0.739085133215161$$

$$n_{\text{fixpoint}} == \cos(n_{\text{fixpoint}})$$

A *Fixpoint* is

Any value, that when given to a function, returns the same value

$$x = f(x)$$

Example Fixpoints

0.739... = cos(0.739...)

0 = sin(0)

1 = sqrt(1)

0 = sqrt(0)

Demo

(conclusion)

Conclusion

The aspects of things that are most important to us are hidden because of their simplicity and familiarity

-- Prof. Ludwig Wittgenstein

If you Enjoyed This

http://experthuman.com/programming-with-nothing

-- Tom Stuart

(google “programming with nothing”)

Thank You

Jim Weirich
Chief Scientist
{ new context }

jim@newcontext.com
@jimweirich