A Brief History of the Freedom of Expressions

Matt Might

http://matt.might.net/
Sapir-Whorf Hypothesis

Language limits thought.

Edward Sapir

Benjamin Whorf
Programming languages

• Where did they come from?
• What kinds of languages do we have now?
• What will they do for us tomorrow?
What is a programming language?

A **programming language** is a language that can describe a transformation from inputs to outputs...
What is a programming language?

A **programming language** is a language that can describe a transformation from inputs to outputs... 

...without ambiguity.
Origins of programming languages
Origins of mathematical notation
Euclid’s algorithm (300 BC)

- Greatest common divisor
- $\text{GCD}(24, 18) = 6$
- $\text{GCD}(4, 6) = 2$

(Probably) Euclid
Δύο αριθμοί δοθέντων μή πρῶτον πρὸς ἀλλήλους τὸ μέγιστον αὐτῶν κοινὸν μέτρον εὑρεῖν.

ΑΕ

Ζ

Η

Δ

Β

Γ

"Εστωσαν οἱ δοθέντες δύο αριθμοὶ μὴ πρῶτοι πρὸς ἀλλήλους οἱ ΑΒ, ΓΔ. δεὶ δὴ τῶν ΑΒ, ΓΔ τὸ μέγιστον κοινὸν μέτρον εὑρεῖν.

Εἰ μὲν οὖν ὁ ΓΔ τὸν ΑΒ μετρεῖ, μετρεῖ δὲ καὶ ἑαυτόν, ὁ ΓΔ ἀρα τῶν ΓΔ, ΑΒ κοινὸν μέτρον ἔστιν. καὶ φανερῶν, ὅτι καὶ μέγιστον συνες γὰρ μεῖζον τοῦ ΓΔ τὸν ΓΔ μετρήσει.

Εἰ δὲ οὐ μετρεῖ ὁ ΓΔ τὸν ΑΒ, τῶν ΑΒ, ΓΔ ἀνθυραρ-ρουμένου δεῖ τοῦ ἐλάσσονος ἀπὸ τοῦ μεῖζονος λειψάνησα τις ἀριθμός, δεῖ μετρήσει τὸν πρὸς ἑαυτοῦ. μονάς μὲν γὰρ οὐ λειψάνησα: εἰ δὲ μὴ, ἔστωσαν οἱ ΑΒ, ΓΔ πρῶτοι πρὸς ἀλλήλους· ὑπερ οὐχ ὑπόκειται. λειψάνησα τις ἀρα ἀριθμός, δεῖ μετρήσει τὸν πρὸς ἑαυτοῦ. καὶ μὲν ΓΔ τὸν BE μετρῶν λειπτῶν ἑαυτοῦ ἐλάσσονα τὸν ΕΑ, ὁ δὲ ΕΑ τὸν ΔΖ μετρῶν λειπτῶν ἑαυτοῦ ἐλάσσονα τὸν ΖΓ, ὁ δὲ ΓΖ τὸν ΑΕ μετρῶ. ἐπεὶ οὖν ὁ ΓΖ τὸν ΑΕ μετρεῖ, ὁ δὲ ΔΕ τὸν ΔΖ μετρεῖ, καὶ ὁ ΓΖ ἀρα τὸν ΔΖ μετρήσει. μετρεῖ δὲ καὶ ἑαυτόν· καὶ ἑαυτὸν καὶ ἐλασσόνα τὸν ΓΔ μετρήσει. ὁ δὲ ΓΔ τὸν BE μετρεῖ· καὶ ὁ ΓΖ ἀρα τὸν BE μετρεῖ· μετρεῖ δὲ καὶ τὸν ΕΑ· καὶ ἑαυτὸν καὶ ἐλασσόνα τὸν ΒΑ μετρήσει· μετρεῖ δὲ καὶ τὸν ΓΔ· ὁ ΓΖ ἀρα τῶν ΑΒ, ΓΔ μετρεῖ. ὁ ΓΖ ἀρα τῶν ΑΒ, ΓΔ κοινὸν μέτρον ἔστιν. λέγω δή, ὅτι καὶ μέγιστον, εἰ γὰρ μὴ ἔστιν ὁ ΓΖ τῶν ΑΒ, ΓΔ μέγιστον κοινὸν μέτρον, μετρήσει τις τοὺς ΑΒ, ΓΔ ἀριθμοὺς ἀριθμὸς μεῖζον δὲν τοῦ ΓΖ, μετρεῖτα, καὶ ἐστιν ὁ Η. καὶ ἐπεὶ ὁ Η τὸν ΓΔ μετρεῖ, ὁ δὲ ΓΔ τὸν BE μετρεῖ, καὶ ὁ Η ἀρα τὸν BE μετρεῖ· μετρεῖ δὲ καὶ ἑαυτόν· καὶ ἑαυτὸν καὶ τὸν ΑΕ μετρήσει. ὁ δὲ ΔΕ τὸν ΔΖ μετρεῖ· καὶ ὁ Η ἀρα τὸν ΔΖ μετρήσει· μετρεῖ δὲ καὶ ἑαυτὸν· καὶ ἑαυτὸν καὶ τὸν ΓΖ μετρήσει· οἱ μεῖζον τοῦ ἐλάσσονα· ὅπερ ἔστιν ἀδύνατον· οὐκ ἀρα τοὺς ΑΒ, ΓΔ ἀριθμοὺς ἀριθμὸς τις μετρήσει μεῖζον δὲν τοῦ ΓΖ· ὁ ΓΖ ἀρα τῶν ΑΒ, ΓΔ μέγιστον ἕστι κοινὸν μέτρον [ὁπερ ἔδει δεῖξαι].

Πόρισμα.

"Εξ δὴ τούτου φανερῶν, ὅτι ἐὰν ἀριθμὸς δύο ἀριθμοὺς μετρή, καὶ τὸ μέγιστον αὐτῶν κοινὸν μέτρον μετρήσει· ὅπερ ἔδει δεῖξαι.
2008 AD

gcd(a, b) = (b == 0) ? a : gcd(b, a mod b)
Limits on the Greeks

• No notation for zero.
• No variables for unknowns.
• No symbols for operations.
• Long division required Ph.D.
• Irrational numbers punished by death.
Example

• The number such that four of its roots is equal to its three of its square.

• $4x = 3x^2$. 
Indian numerals (596)

- Notation for zero.
- Decimal numerals.
- Calculation easier.

Brahmagupta
Solving quadratics (820)

Muhammad ibn Mūsā al-Khwārizmī
Solving quadratics (820)

\[ x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \]

Muhammad ibn Mūsā al-Khwārizmī
Variables (1570s)

François Viète
Example

• $3 + x$ is equal to 10 times $x^2$.
• $3 + x = 10x^2$
Operations (Early 1600s)

- Letters for variables.
- Symbols for operations.
- Led to slide rule.

William Oughtred
Calculus (Late 1600s)

Isaac Newton

Principia (1687)
Calculus (Late 1600s)

Gottfried Leibniz

Fig. 124.—Facsimile of manuscript of Leibniz, dated Oct. 29, 1675, in which his sign of integration first appears. (Taken from C. I. Gerhardt’s Briefwechsel von G. W. Leibniz mit Mathematikern [1890].)
Euler (1700s)
The end of the reign of numbers
A function transforms an input into an output.

\[ f(x) = x^2 + 3 \]

<table>
<thead>
<tr>
<th>input: ( x )</th>
<th>output: ( f(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
</tr>
</tbody>
</table>
Functions

A **programming language** is a language that can describe a transformation from inputs to outputs.
A **programming language** is a language that can describe functions.
Names for functions

\[ f(x) \]
Names for functions
Names for functions

$\textbf{f}^{(3)}$
Sets and Logic (1800s)

- Sets became objects.
- Logic became math.
- Math began unifying.

Giuseppe Peano
Sets

A set is a collection of objects.

• \{1,2,3\} is a set of three numbers.
• \{} is the empty set.
• \{\{1,2,3\}\} is a set containing a set.
• \(\mathbb{N}\) is the set of natural numbers.
Frege’s unification

- Logic as foundation.
- Sets as atoms.
- Numbers from sets.
- Functions from sets.

Gottlob Frege
Example

- Every even integer greater than two can be written as the sum of two primes.
  \[ \forall n > 2 : \exists a, b : \text{p}(a) \land \text{p}(b) \land a + b = n. \]
Grundgesetze der Arithmetik

- Published in 1903.
- Foundation for math.
Dear Frege,

\[ \exists X \mid X \not\in X \exists X \mid X \not\in X \]

XOXO,

Bertrand R.
Dear Frege,

U FAIL.

XOXO,
Bertrand R.

Prof. Dr. Gottlob Frege
University of Göttingen
Göttingen, Germany
Russell’s paradox

\[ \{X \mid X \not\in X\} \in \{X \mid X \not\in X\} \]
Russell’s paradox

\[ \{ X \mid X \not\in X \} \in \{ X \mid X \not\in X \} \]

Does the set of all sets that do not contain themselves contain itself?
Russell’s paradox

\[ \{ X \mid X \not\in X\} \in \{ X \mid X \not\in X\} \]

Does the set of all sets that do not contain themselves contain itself?

The barber shaves all those that do not shave themselves.
Russell’s paradox

\[ \{ X \mid X \notin X \} \in \{ X \mid X \notin X \} \]

Does the set of all sets that do not contain themselves contain itself?

The barber shaves all those that do not shave themselves.

But, then who shaves the barber?
Russell’s solution: Orders

- Problem is self-reference.
- Example: This sentence is false.
- Solution: Order sentences.
- Must reference lower orders.
- Seems to avoid paradox.

Bertrand Russell
Functions as foundation?

- Notation for functions.
- \( f(x) = x^2 \)
- \( f(2) = 4 \)
- \( f = \lambda x. x^2 \)
- \( (\lambda x. x^2)(2) = 4 \)
Lambda Calculus (1920s)

Throw away everything in math, except:

<table>
<thead>
<tr>
<th>$x$</th>
<th>variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(e)$</td>
<td>function application</td>
</tr>
<tr>
<td>$\lambda x. e$</td>
<td>function definition</td>
</tr>
</tbody>
</table>
Lambda Calculus

\[(\lambda x. e) v = \{v/x\} e.\]
The Lambda Calculus

Can encode:

- Numbers.
- True and false.
- Propositions.
- Sets.
- Recursive functions.
- Logic.
- ...
Another paradox

\[ k = (\lambda x. \neg (x \ x)) \ (\lambda x. \neg (x \ x)) = \neg k \]
Turing machine (1936)

- Student of Church.
- Defined computability.
- Showed $\lambda = \text{computer}$.
Modern programming languages
Lisp (1958)

- $\lambda$ as programming language.
- S-Expression notation.
- Code as data.
- Data as code.
- Self-evaluating.

John McCarthy
Example

(define (factorial x)
  (if (= x 0)
      1
      (* n (factorial (- n 1)))))))
Kinds of languages

• Declarative: Describes relationship.

• Imperative: Describes process.
Example: Declarative

- A PBJ is the result of placing peanut butter and jelly between two slices of bread.
Example: Imperative

- Place slice of bread on table.
- Add peanut butter.
- Add jelly.
- Place slice of bread on top.
Declarative languages

- Roots in the lambda calculus.
- Examples: Haskell, SML, Lisp/Scheme
- Easy to reason about correctness.
- Generally safe and bug free.
- Can be inefficient.
- Need to think mathematically to use them.
Imperative languages

• Trace roots to Babbage’s engine (1800s).
• Examples: C, C++, Java, Python.
• Hard to know if correct.
• Generally very efficient.
• Extremely dangerous.
<exploding-rocket-video />
A fatal exception 0E has occurred at 0137:BFFA21C9. The current application will be terminated.

* Press any key to terminate the current application.
* Press CTRL+ALT+DEL again to restart your computer. You will lose any unsaved information in all applications.

Press any key to continue
<exploding-rocket-video />
The future of programming languages
Front lines of research

• Get the best of both worlds.
• Write program in both; prove equivalent.
• Use a mixed language: Scala, Ynot.
The future

- Programs won’t crash.
- Programs won’t hang.
- Programs will always do the right thing.
Thank you

http://matt.might.net/
Example: Numbers

• \( 0 = \lambda s.\lambda z.z \)

• \( n+1 = \lambda s.\lambda z.s(n \ s \ z) \)
Example: True, false, if

- $true = \lambda t.\lambda f.t$
- $false = \lambda t.\lambda f.f$
- $if = \lambda b.\lambda t.\lambda f.(b \ t \ f)$
\[ \omega = (\lambda x.x \ x) \ (\lambda x.x \ x) = \omega \]
Fixed-point Combinator

$$\text{Fix}(F) = F(\text{Fix}(F))$$
Fixed points and self reference

• $x^2 = 1 + x$
• $x = x^2 - 1$
• $x = F(x)$, where $F = \lambda x.x^2 - 1$
• $x = \text{Fix}(F) = \pm 1$
Recursion

- \( f(n) = \text{if } n=0 \text{ then } 1 \text{ else } n \times f(n-1) \)
- \( F(f) = \lambda n. n=0 \text{ then } 1 \text{ else } n \times f(n-1) \)
- \( ! = \text{Fix}(F) \)
Fixed-point Combinator

\[ \text{Fix} = \lambda f. (\lambda x. f (x x)) (\lambda x. f (x x)) \]