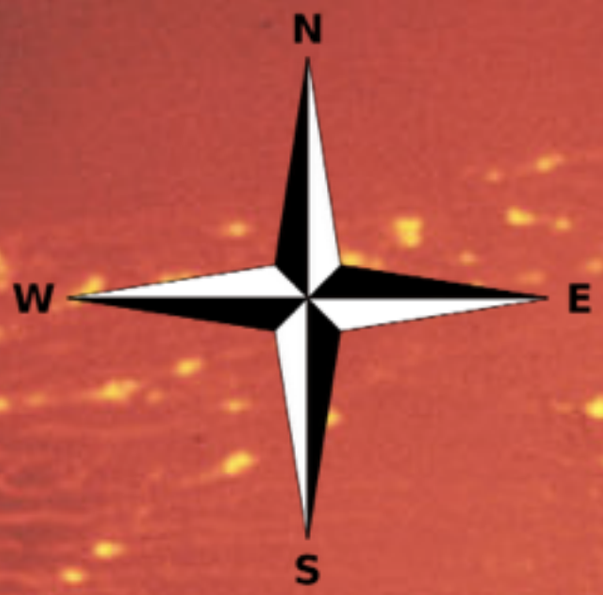
PROGRAMMING
IN
INTERESTING TIMES

@russolsen

YOUR
PROGRAMMING LANGUAGE
IS GOING TO

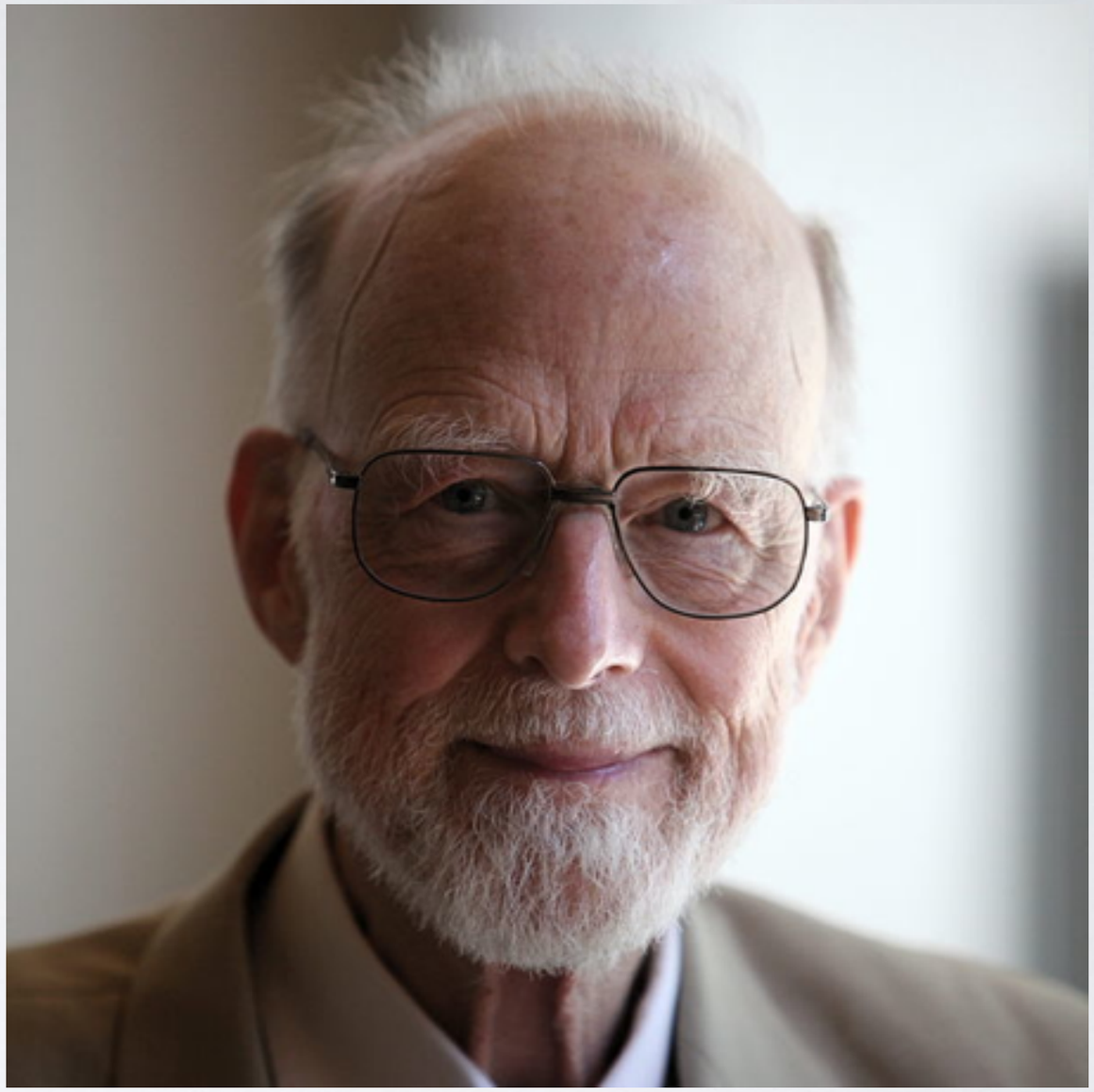
DIE


N
W
E
S

```
MOVE A TO B.
COMPUTE GROSS-PAY = HOURS-WORKED * HOURLY-RATE
MULTIPLY HOURLY-RATE BY HOURS-WORKED GIVING GROSS-PAY
SET MY-INDEX TO 1

SET ADDRESS OF MY-LINKAGE-SECTION-ITEM TO MY-POINTER
READ TRANSACTION-FILE INTO TRANSACTION-RECORD-WS
```

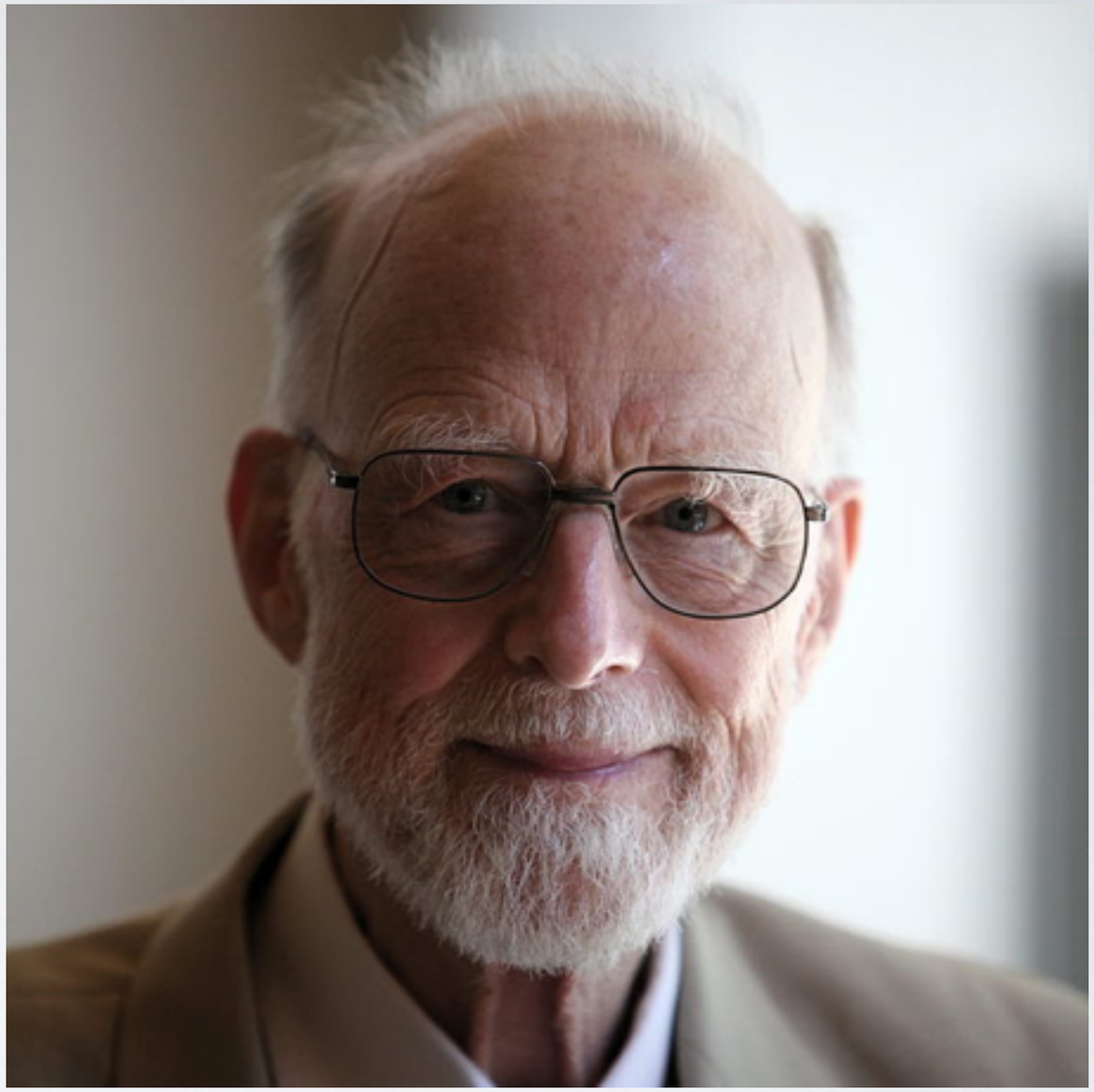
```
C AREA OF A TRIANGLE - HERON'S FORMULA
C INPUT - CARD READER UNIT 5, INTEGER INPUT, NO BLANK CARD FOR END OF DATA
C OUTPUT - LINE PRINTER UNIT 6, REAL OUTPUT
C INPUT ERROR DISPAYS ERROR MESSAGE ON OUTPUT
  501 FORMAT(3I5)
  601 FORMAT(" A= ",I5,"  B= ",I5,"  C= ",I5,"  AREA= ",F10.2,"SQUARE UNITS")
  602 FORMAT("NORMAL END")
  603 FORMAT("INPUT ERROR OR ZERO VALUE ERROR")
      INTEGER A,B,C
   10 READ(5,501,END=50,ERR=90) A,B,C
      IF(A=0 .OR. B=0 .OR. C=0) GO TO 90
      S = (A + B + C) / 2.0
      AREA = SQRT( S * (S - A) * (S - B) * (S - C))
      WRITE(6,601) A,B,C,AREA
      GO TO 10
   50 WRITE(6,602)
      STOP
   90 WRITE(6,603)
      STOP
      END
```

“I DON'T KNOW WHAT THE LANGUAGE OF THE YEAR 2000 WILL LOOK LIKE,



TONY HOARE, 1982

“I DON'T KNOW WHAT THE LANGUAGE OF THE YEAR 2000 WILL LOOK LIKE, BUT I KNOW IT WILL BE CALLED FORTRAN”



TONY HOARE, 1982


FORTRAN
1957
1983
COBOL
1959
1983

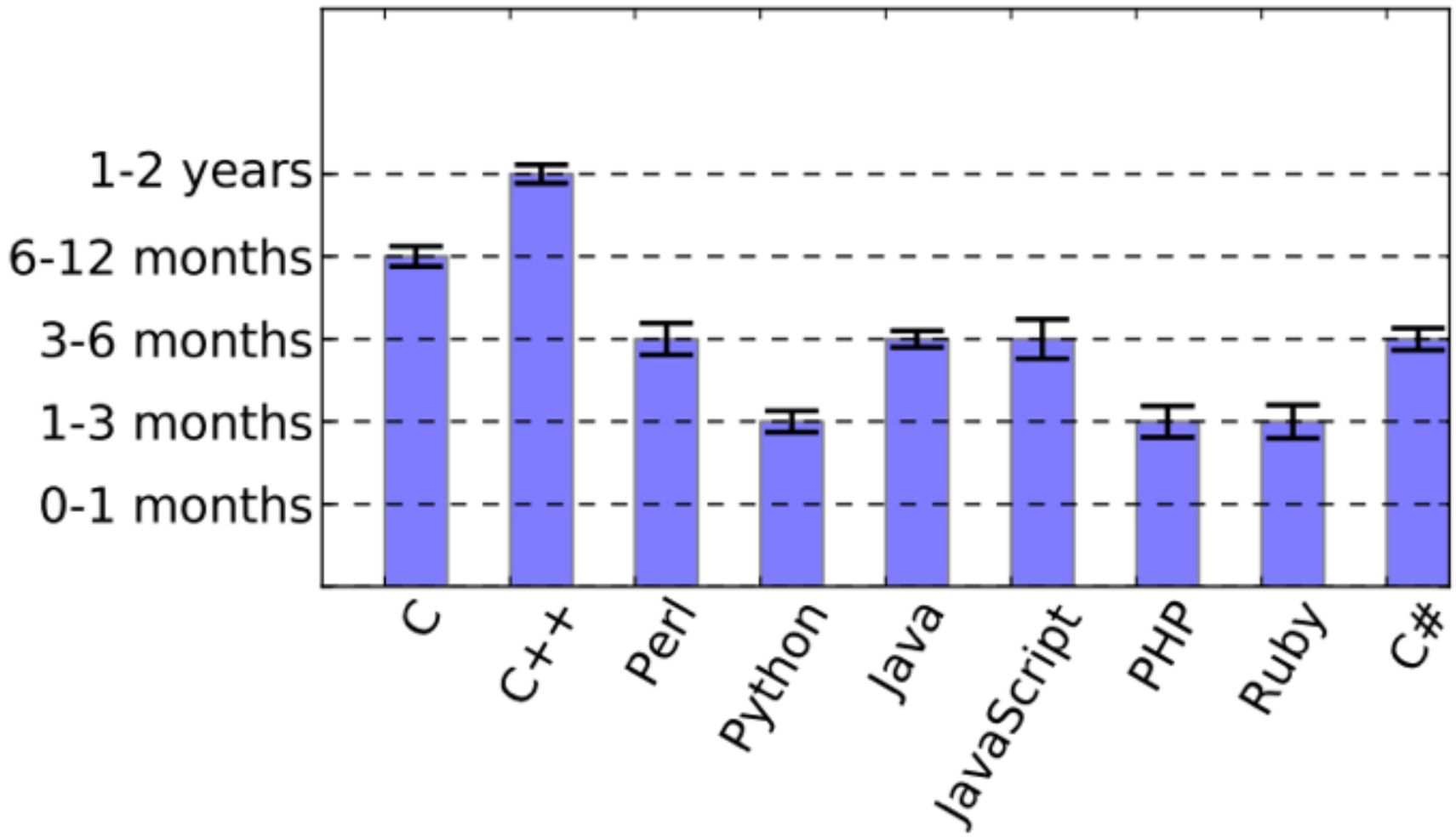
Human sacrifice, dogs and cats living together…

Mass hysteria!

DR. PETER VENKMAN, 1984

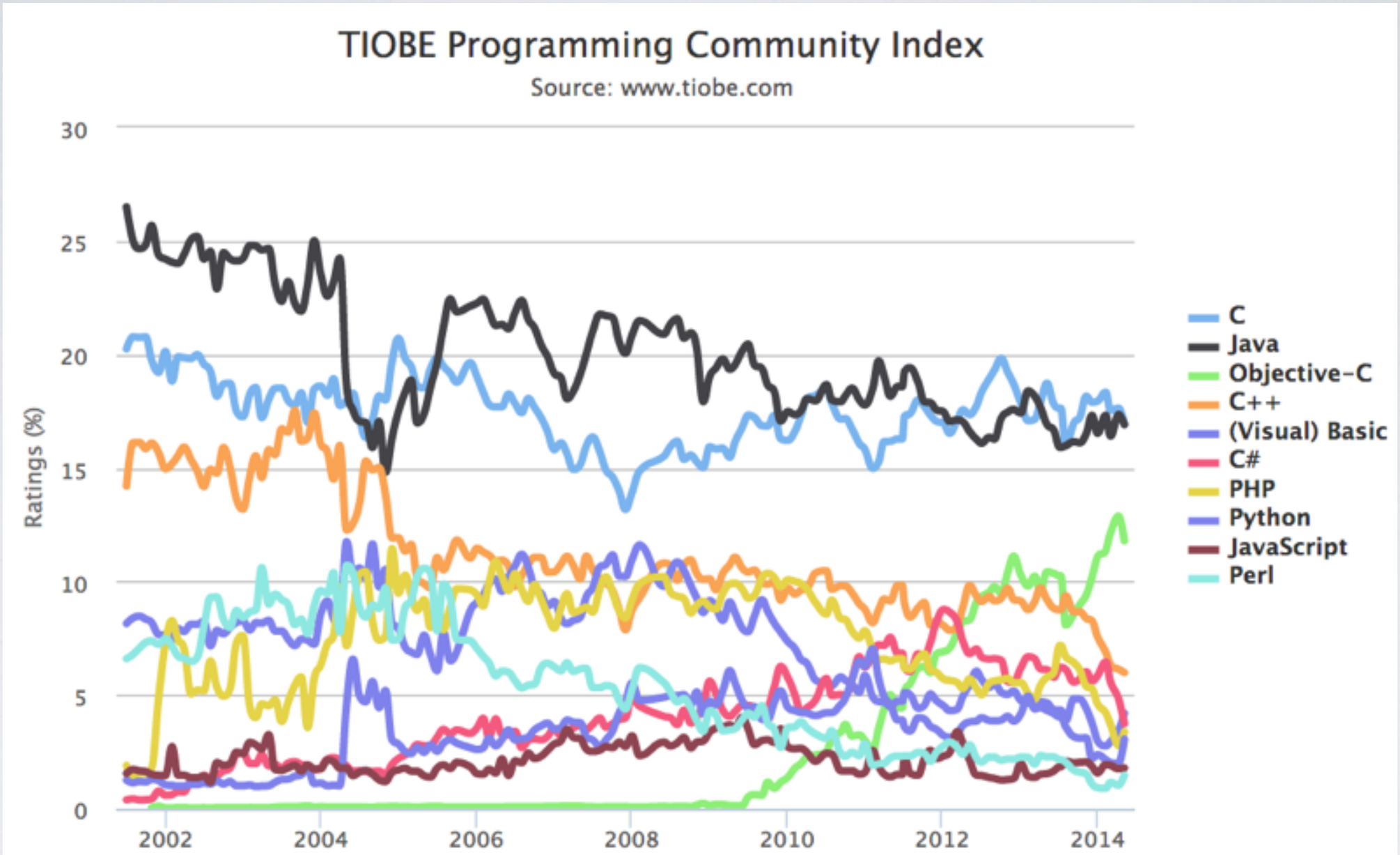
YOUR
PROGRAMMING LANGUAGE
IS GOING TO

DIE

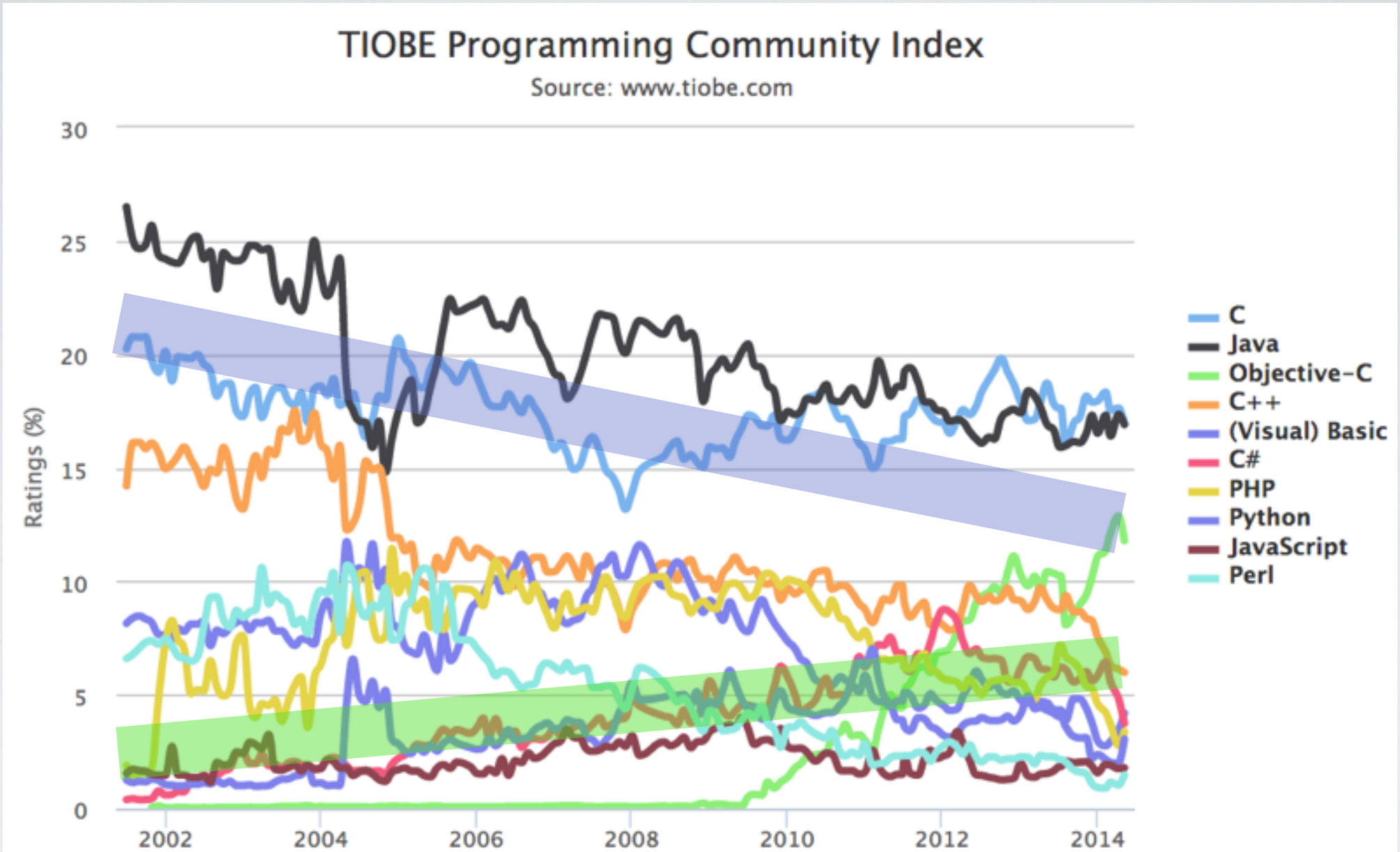


SPEED OF ACQUISITION

Source: Meyerovich & Rabkin 2013



LANGUAGE POPULARITY



LANGUAGE POPULARITY

2002 TOP FIVE

- **Java**
- **C**
- **C++**
- **Visual Basic**
- **Perl**

74%

Source: tiobe.com

2014 TOP FIVE

- **C**
- **Java**
- **Objective C**
- **C++**
- **Visual Basic**

54%

Source: tiobe.com

2014 THE NEXT ELEVEN

- **C#**
- **PHP**
- **Python**
- **JavaScript**
- **Perl**
- **Ruby**
- **F#**
- **Plus Five More!**

20%

Source: tiobe.com

WELCOME TO

INTERESTING TIMES

HOW DO I

PICK?

WHAT AM I PICKING?

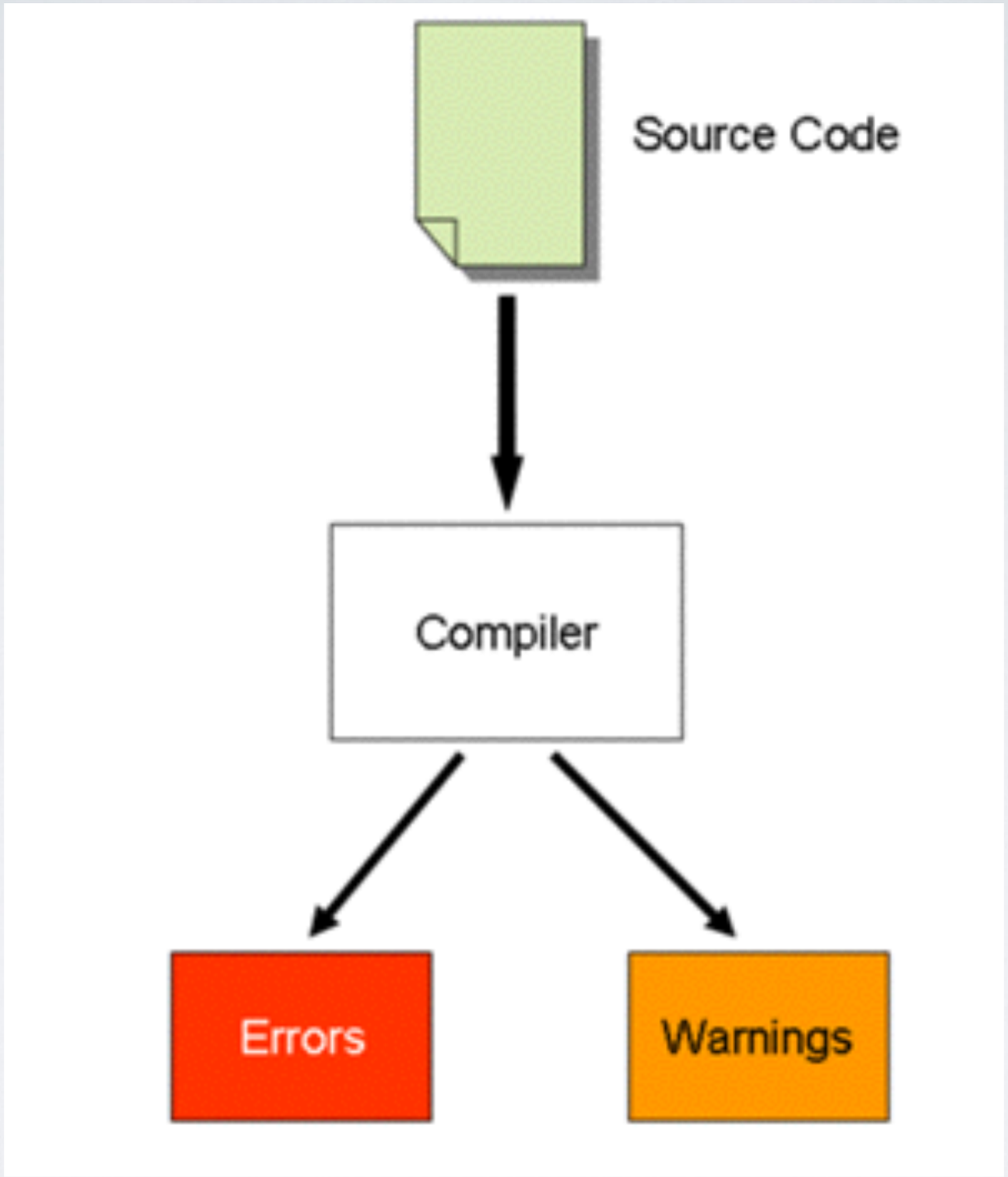
Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.

Source: http://en.wikipedia.org/wiki/Java_(programming_language)

```
CompilationUnit:
    [[Annotations] package QualifiedIdentifier ;]
                                {ImportDeclaration} {TypeDeclaration}

ImportDeclaration:
    import [static] Identifier { . Identifier } [. *] ;

TypeDeclaration:
    ClassOrInterfaceDeclaration
    ;

ClassOrInterfaceDeclaration:
    {Modifier} (ClassDeclaration | InterfaceDeclaration)


ClassDeclaration:
    NormalClassDeclaration
    EnumDeclaration
```

...

Source: http://docs.oracle.com/javase/specs/jls/se7/html/jls-18.html


Source Code
Compiler
Errors
Warnings


Профилиране
STEFAN DR CHOLAKOV
Апостол Апостолов
Ангел Грънчаров
Fibank
Fibank
Fibank

HOW DO I

PICK?

POPULAR?

WHY ARE YOU USING THIS LANGUAGE?

1. Availability of open source libraries

2. Extending existing code

3. Already using it

4. Personal familiarity

5. Team familiarity

WHY ARE YOU USING THIS LANGUAGE?

6. Performance

7. Portability/platform

8. Development speed

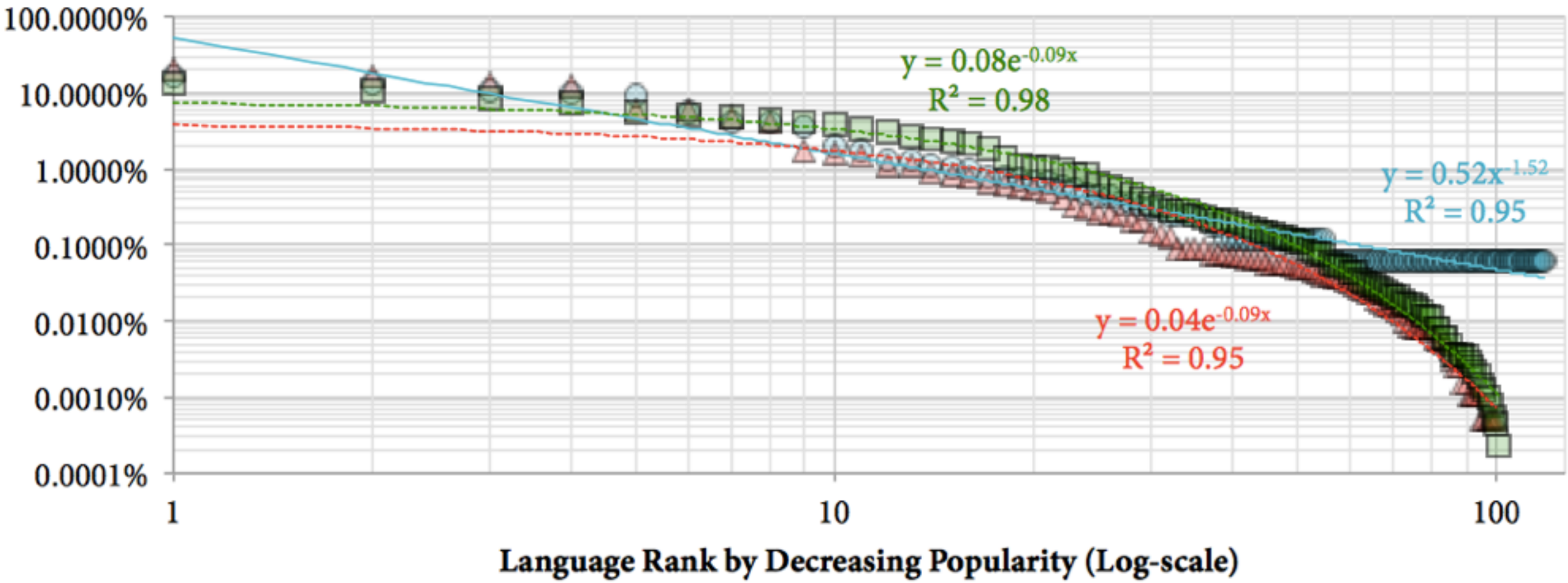
9. Tools

10. Safety/correctness

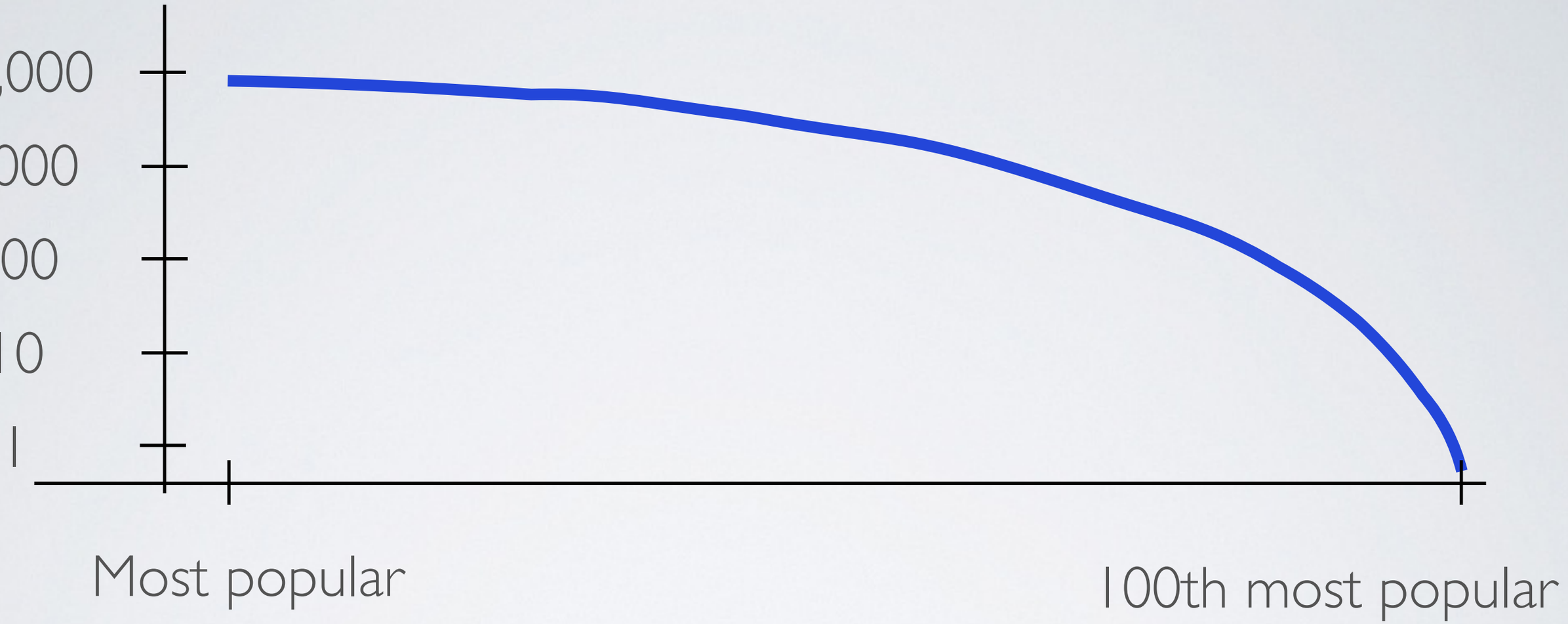
WHY ARE YOU USING THIS LANGUAGE?

I I. Language features

Source: Meyerovich & Rabkin 2013



Source: Meyerovich & Rabkin 2013



Simplified from: Meyerovich & Rabkin 2013


SNAKES
ON A PLANE

You are here

POPULAR

WHERE?

Java

Widget Manufacturing

Gadget Production

Gizmo Assembly

Java

Ruby

DB back web apps;
Fast time to market

FTN

Scientific

Source: Meyerovich & Rabkin 2013

FTN

Scientific

Source: Meyerovich & Rabkin 2013

FTN

Scientific

Source: Meyerovich & Rabkin 2013


R.I.P.

Less popular languages are used in fewer domains

Less popular languages are used in fewer domains

Where they might be quite popular

HOW DO I

PICK?

WHICH FAMILY?

ROB PIKE

Ruby

Ruby

Python

Ruby

Python

Perl

Go

Objective C

C++

Ruby

Python

Perl

2009
Go
Go lang

1983
OC
Objective C

1983
C++
C PlusPlus

1995
Ru
Ruby

1991
Py
Python

1987
Prl
Perl

2007
Clj
Clojure

2009
Go
Go lang

1995
Ru
Ruby

1984
CL
Com Lisp

2012
Elx
Elixir

2000
C#
CSharp

1983
OC
Objective C

1991
Py
Python

1975
Scm
Scheme

1986
Erl
Erlang

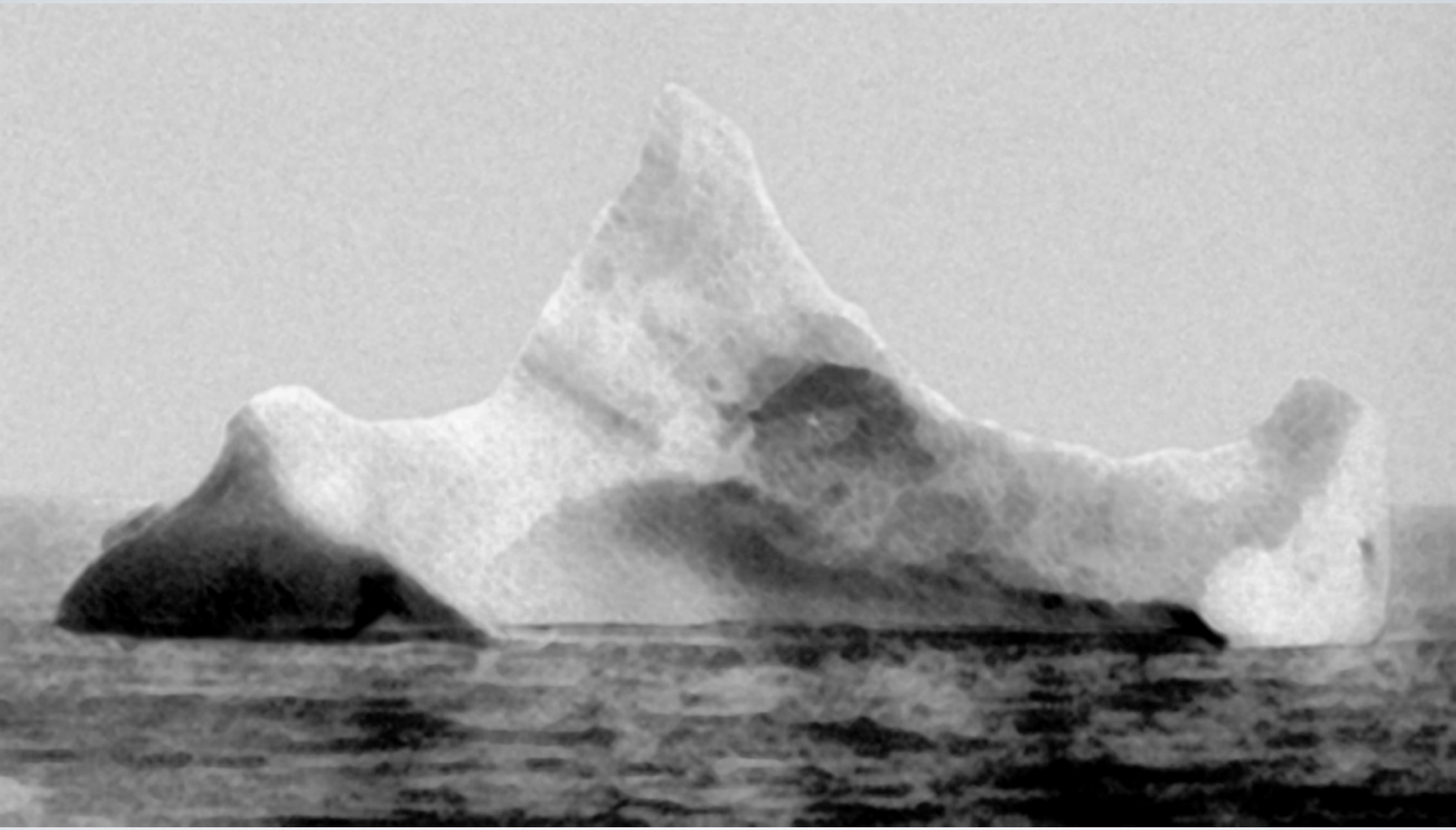
1995
Jv
Java

1983
C++
C PlusPlus

1987
Prl
Perl

1995
JS
JavaScript

SYNTAX



I COULD NEVER WRITE IN A PROGRAMMING LANGUAGE THAT DIDN'T USE BRACES FOR STATEMENT GROUPING.

JAVA DEVELOPER ON RUBY

```
public class Hello {
    public void print_greeting(String name) {
        System.out.println("Hello " + name);
    }

    public static void main(String[] args) {
        Hello hello = new Hello();
        hello.print_greeting("Russ");
    }
}
```

DASHES IN NAMES?

FORGET THIS!

RUBY DEVELOPER ON CLOJURE

```clojure
(defn print-greeting [name]
  (println "Hello" name))

(print-greeting "Russ")
```

Evil

2007
Clj
Clojure

Go
Go lang

1995
Ru
Ruby

1984
CL
Com Lisp

2012
Elx
Elixir

2000
C#
CSharp

1983
OC
Objective C

1991
Py
Python

1975
Scm
Scheme

1986
Erl
Erlang

1995
Jv
Java

1983
C++
C PlusPlus

1987
Prl
Perl

1995
JS
JavaScript

2007
Clj
Clojure

Go
Go lang

1995
Ru
Ruby

1984
CL
Com Lisp

2012
Elx
Elixir

2000
C#
CSharp

1983
OC
Objective C

1991
Py
Python

1975
Scm
Scheme

1986
Erl
Erlang

1995
Jv
Java

1983
C++
C PlusPlus

1987
Prl
Perl

1995
JS
JavaScript

2007
Clj
Clojure

Go
Go lang

1995
Ru
Ruby

1984
CL
Com Lisp

2012
Elx
Elixir

2000
C#
CSharp

1983
OC
Objective C

1991
Py
Python

1975
Scm
Scheme

1986
Erl
Erlang

1995
Jv
Java

1983
C++
C PlusPlus

1987
Prl
Perl

1995
JS
JavaScript

2007
Clj
Clojure

Go
Go lang

1995
Ru
Ruby

1984
CL
Com Lisp

2012
Elx
Elixir

2000
C#
CSharp

1983
OC
Objective C

1991
Py
Python

1975
Scm
Scheme

1986
Erl
Erlang

1995
Jv
Java

1983
C++
C PlusPlus

1987
Prl
Perl

1995
JS
JavaScript

2007
Clj
Clojure

Go
Go lang

1995
Ru
Ruby

1984
CL
Com Lisp

2012
Elx
Elixir

2000
C#
CSharp

1983
OC
Objective C

1991
Py
Python

1975
Scm
Scheme

1986
Erl
Erlang

1995
Jv
Java

1983
C++
C PlusPlus

1987
Prl
Perl

1995
JS
JavaScript

HOW DO I

PICK?

KNOW THE

QUESTIONS

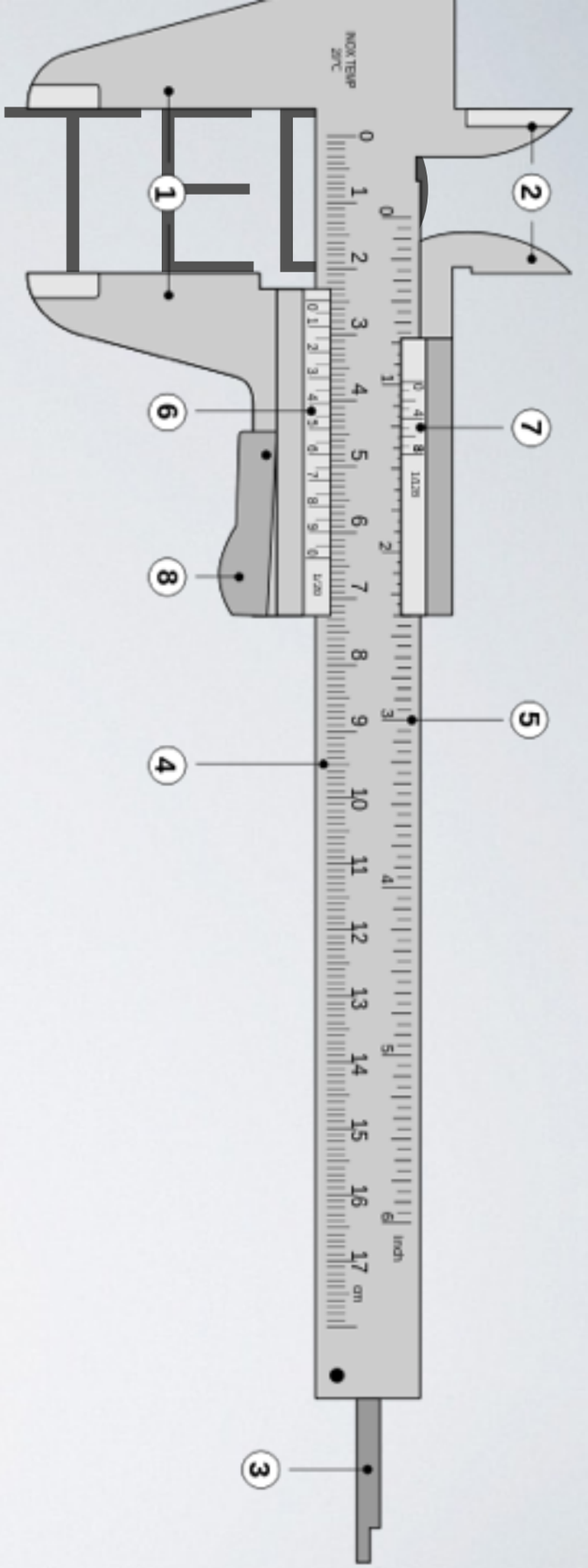
CHASE THE

ANSWERS

ORGANIZE?

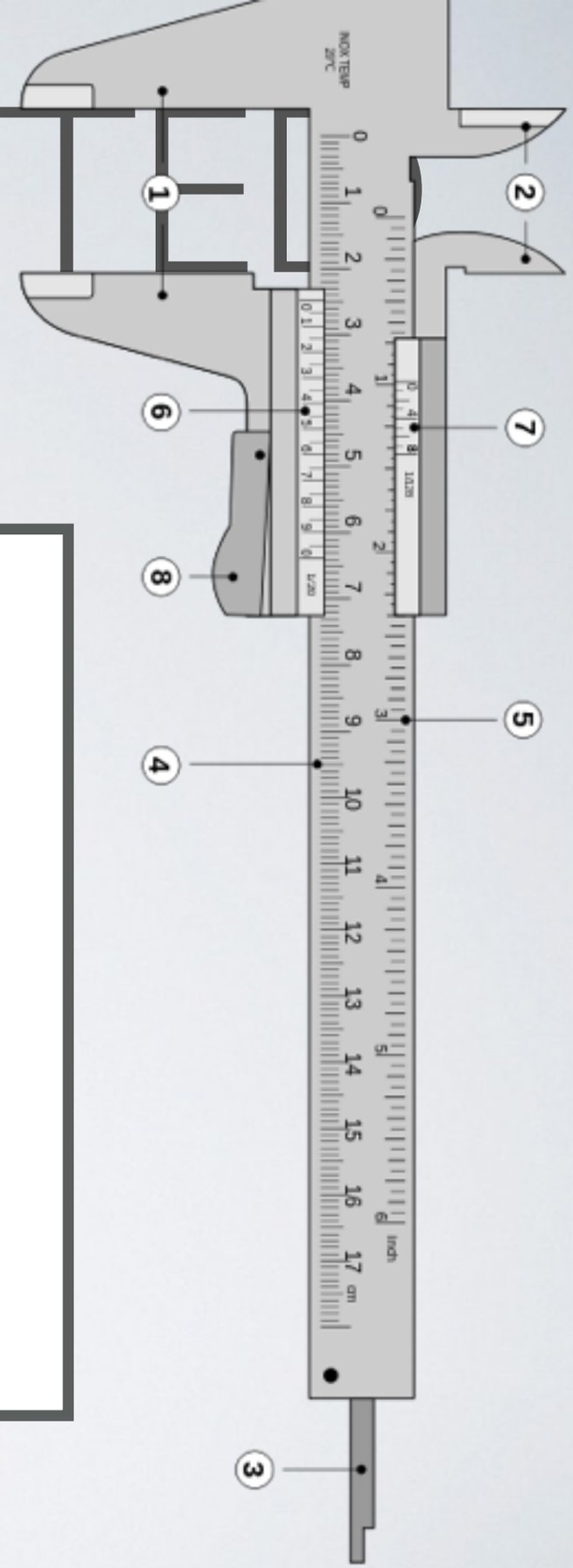
OBJECT ORIENTED

OBJECT ORIENTED

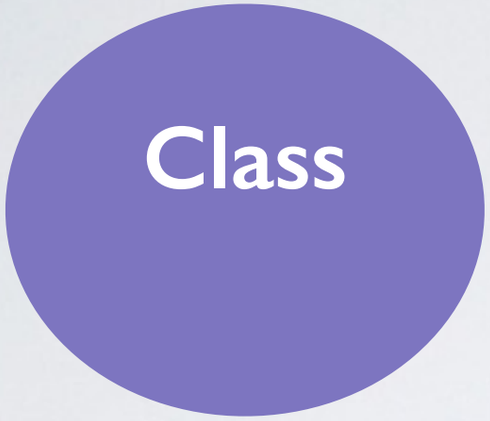


OBJECT ORIENTED

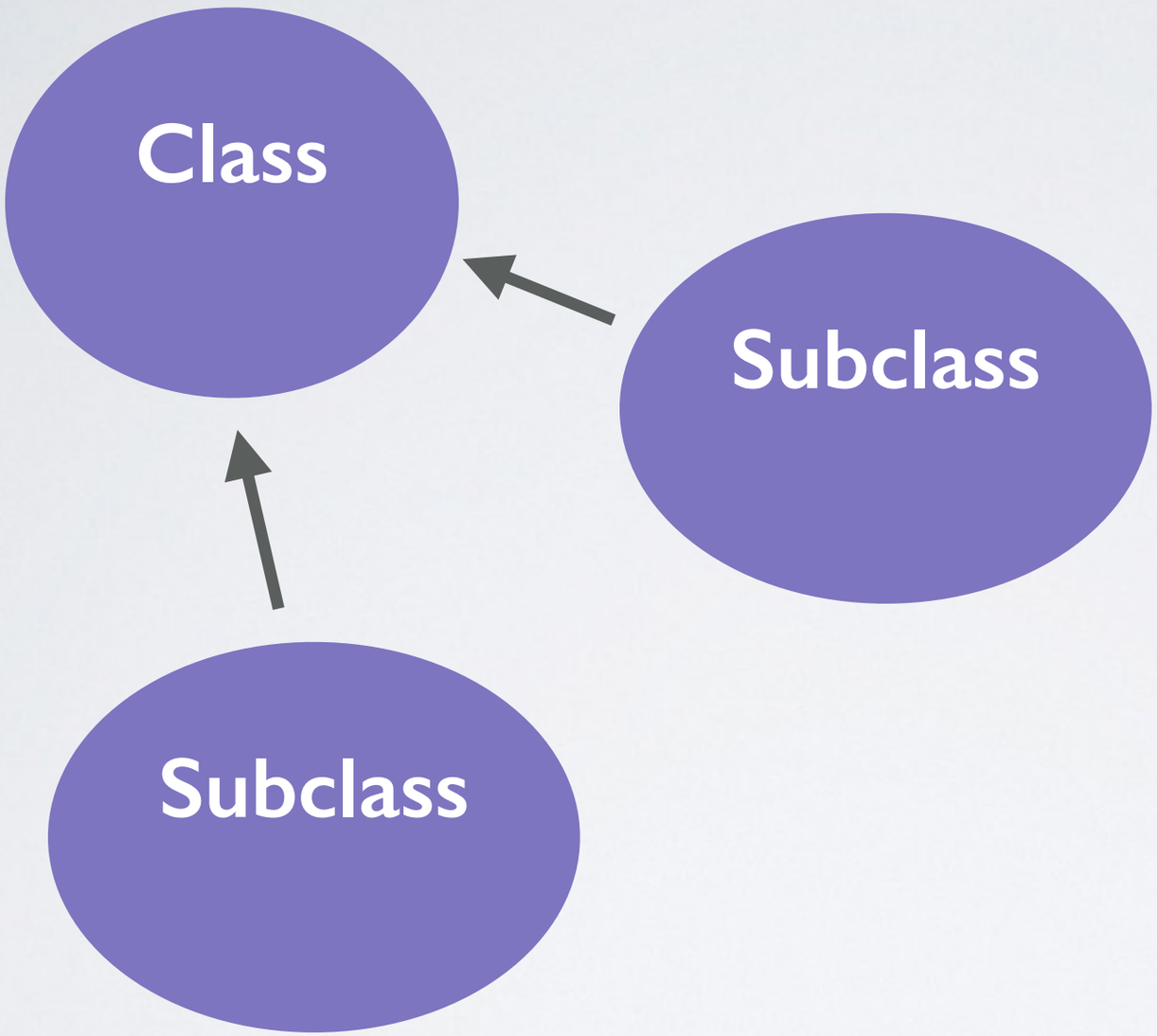
```ruby
class Document
 def initialize(title, text)
    @title = title
    @text = text
  end
  #…
end

class Book < Document
  #... Add publisher and isbn
end

book = Book.new("Gateway", "Pohl")
```



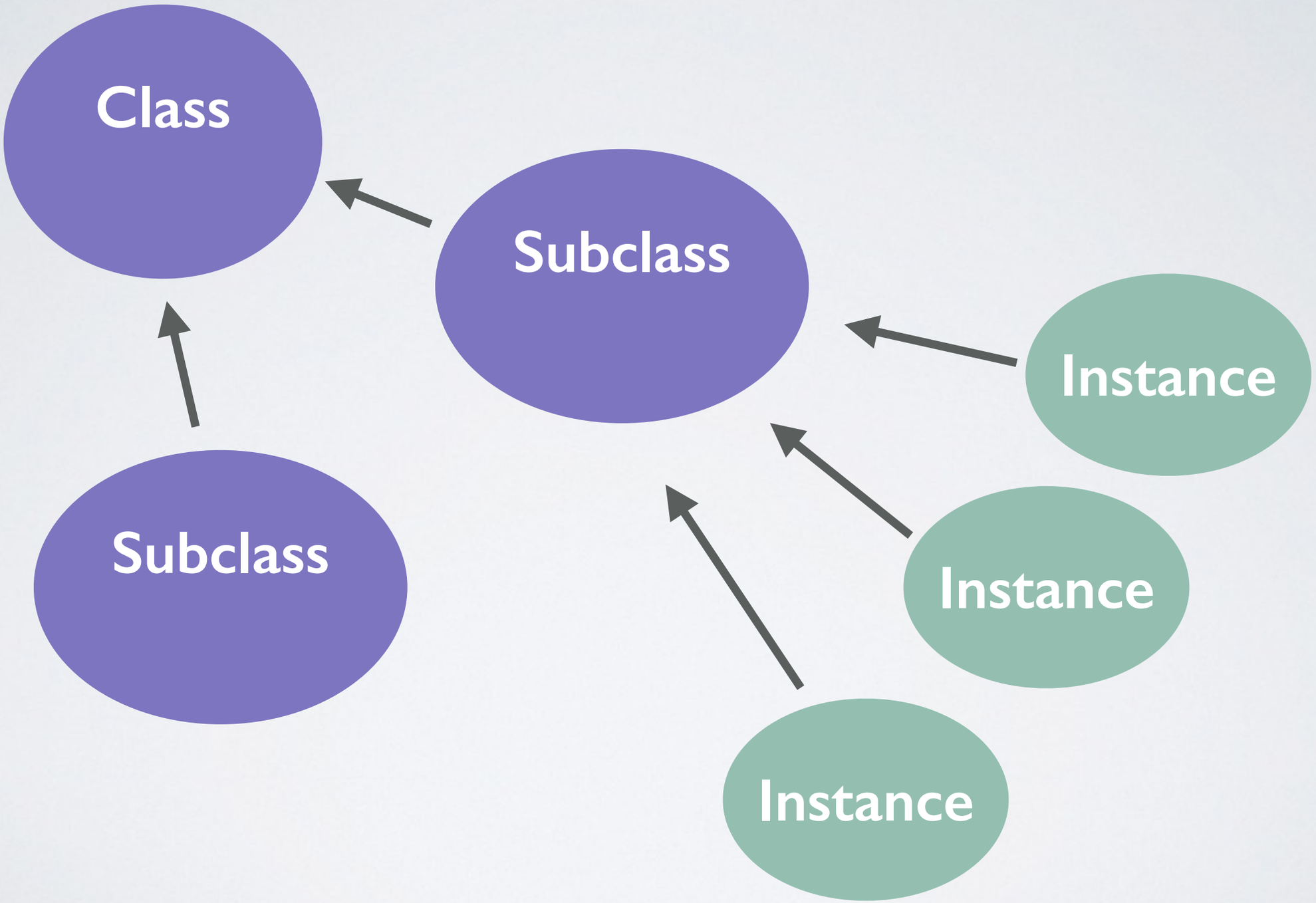
CLASSIC OO



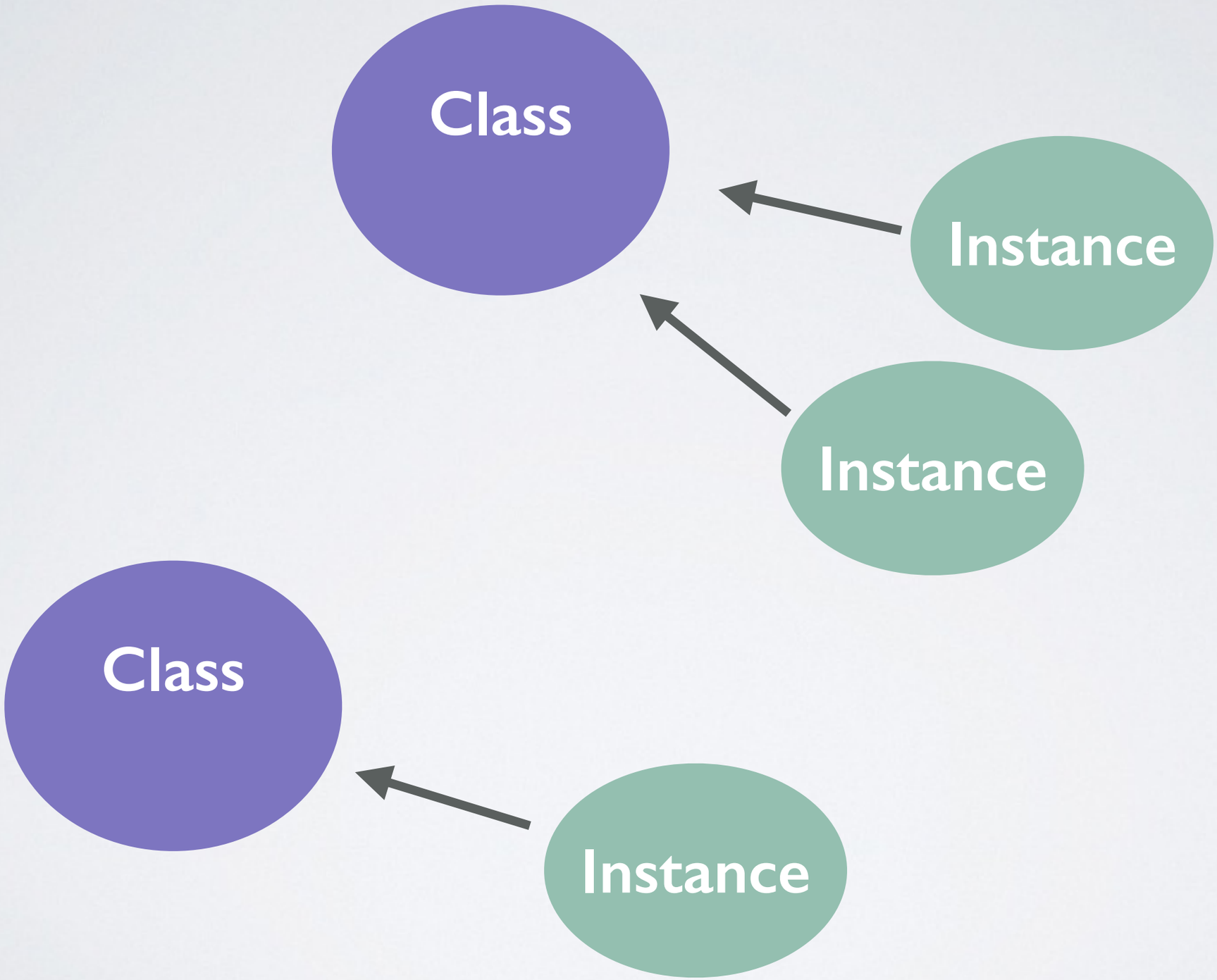
CLASSIC OO



CLASSIC OO


Class
Subclass
Instance
Subclass
Instance
Instance
Instance

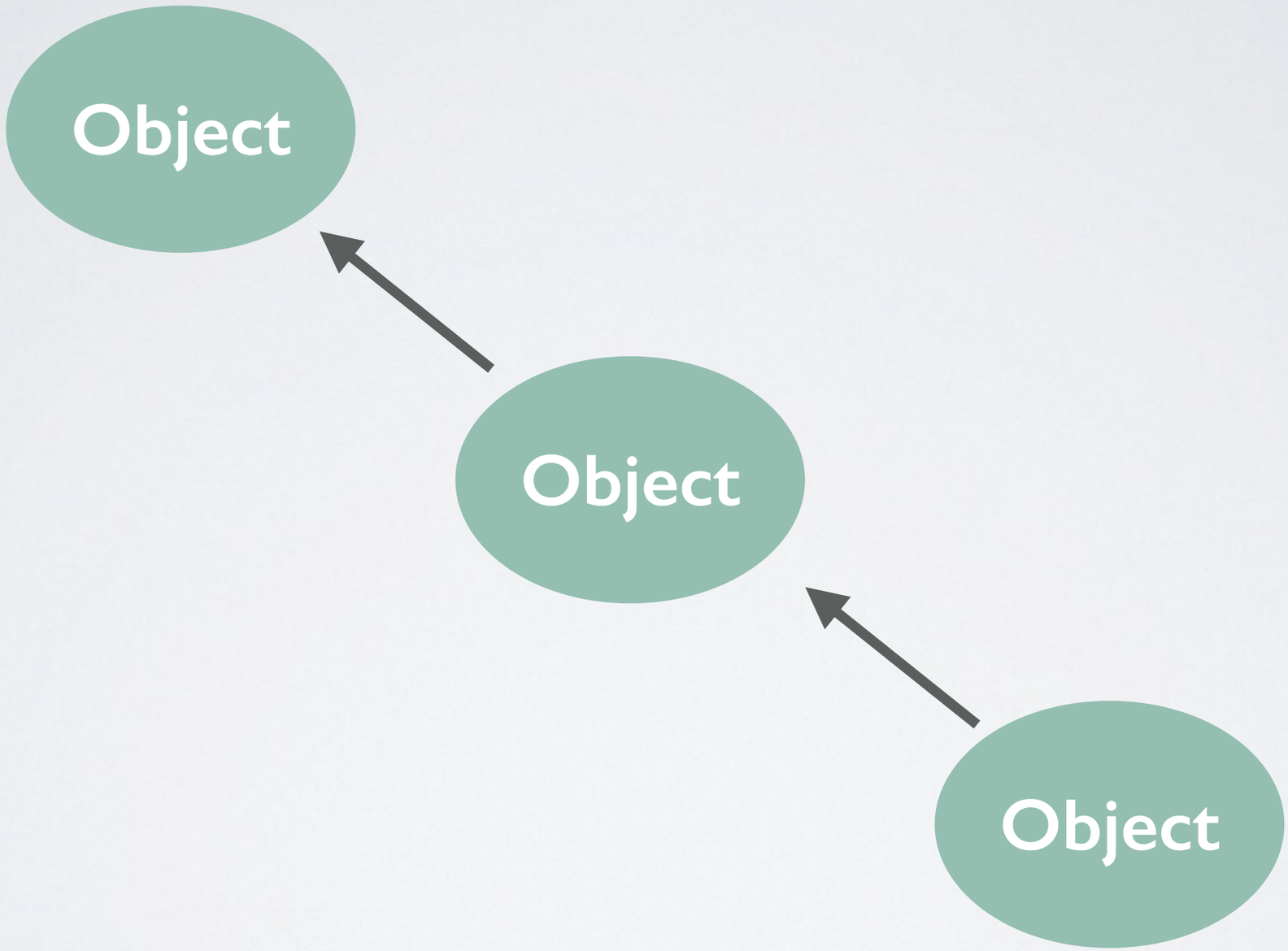
GO: NO INHERITANCE



JAVASCRIPT

```
doc = {
  "title": "Gateway",
  "author": "Pohl"
};

book = {};
book.__proto__ = doc;

book.title  // “Gateway”
```

PROTOTYPES?



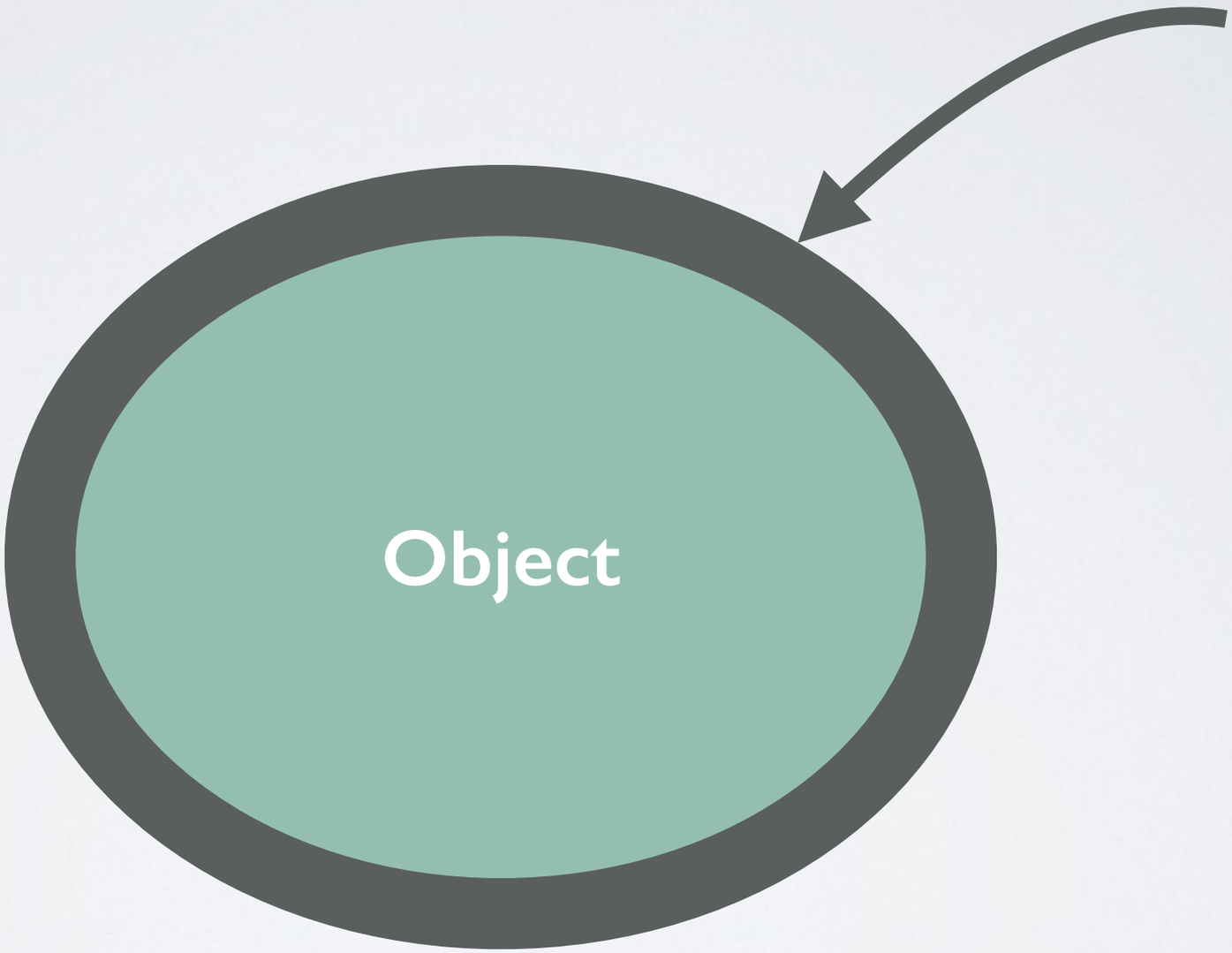
INFORMATION HIDING

```
public class Document {
    private String title;
    private String text;

    public Document(String title, String text) {
        this.title = title;
        this.text = text;
    }

    public String getTitle() {
        return title;
    }

    // …
}
```

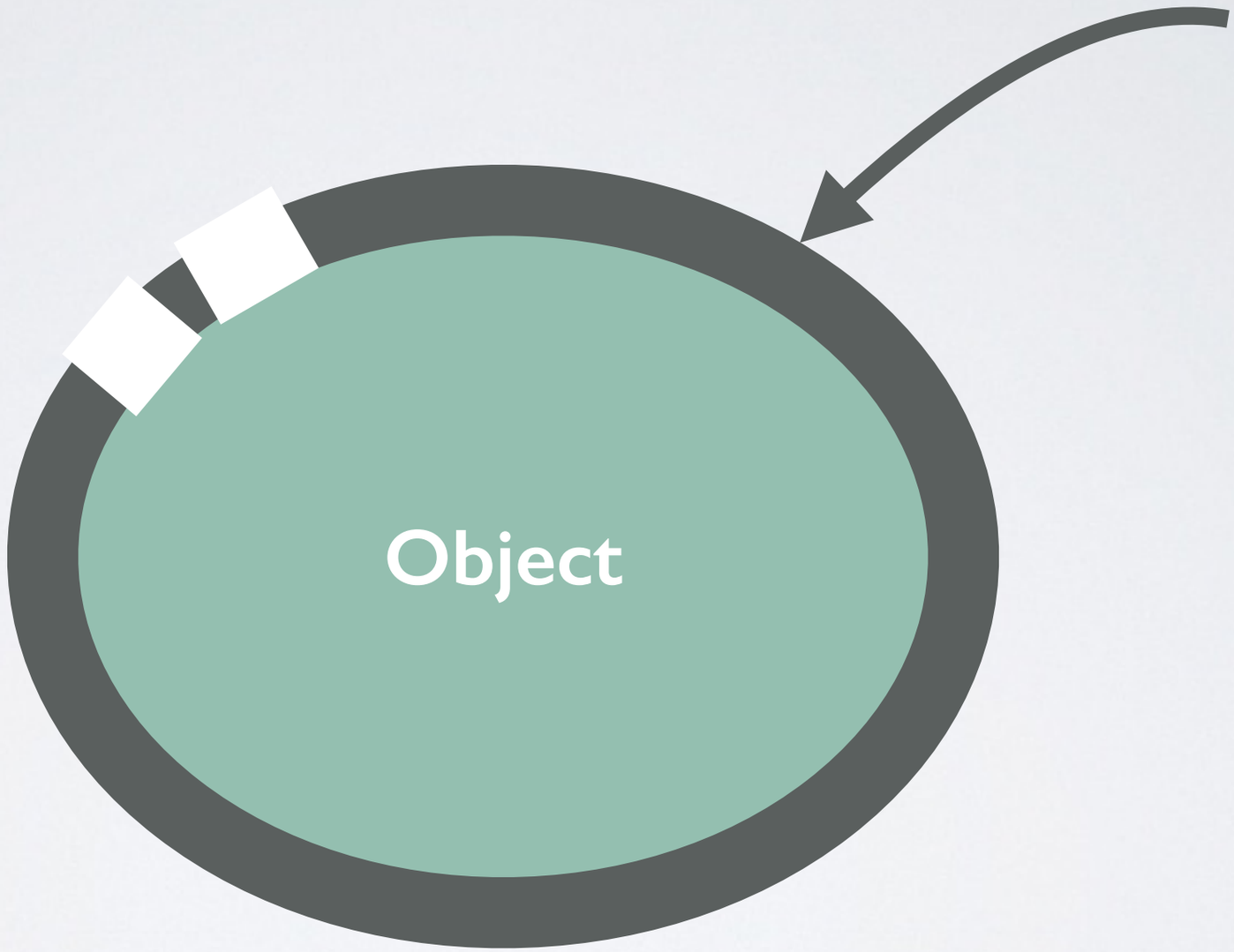
INFORMATION HIDING

```
public class Document {
    private String title;
    private String text;

    public Document(String title, String text) {
        this.title = title;
        this.text = text;
    }

    public String getTitle() {
        return title;
    }

    // …
}
```

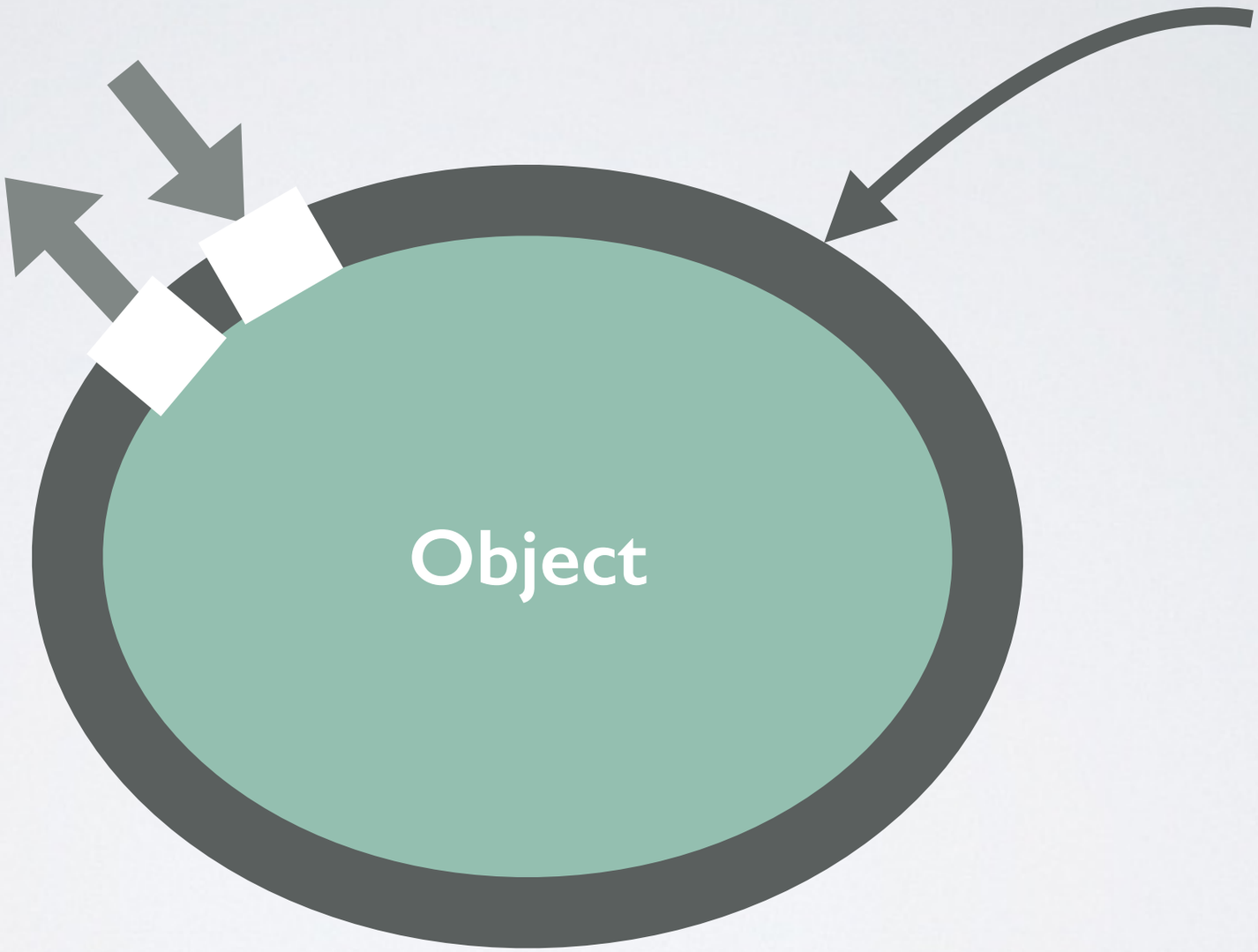
INFORMATION HIDING



INFORMATION HIDING



INFORMATION HIDING



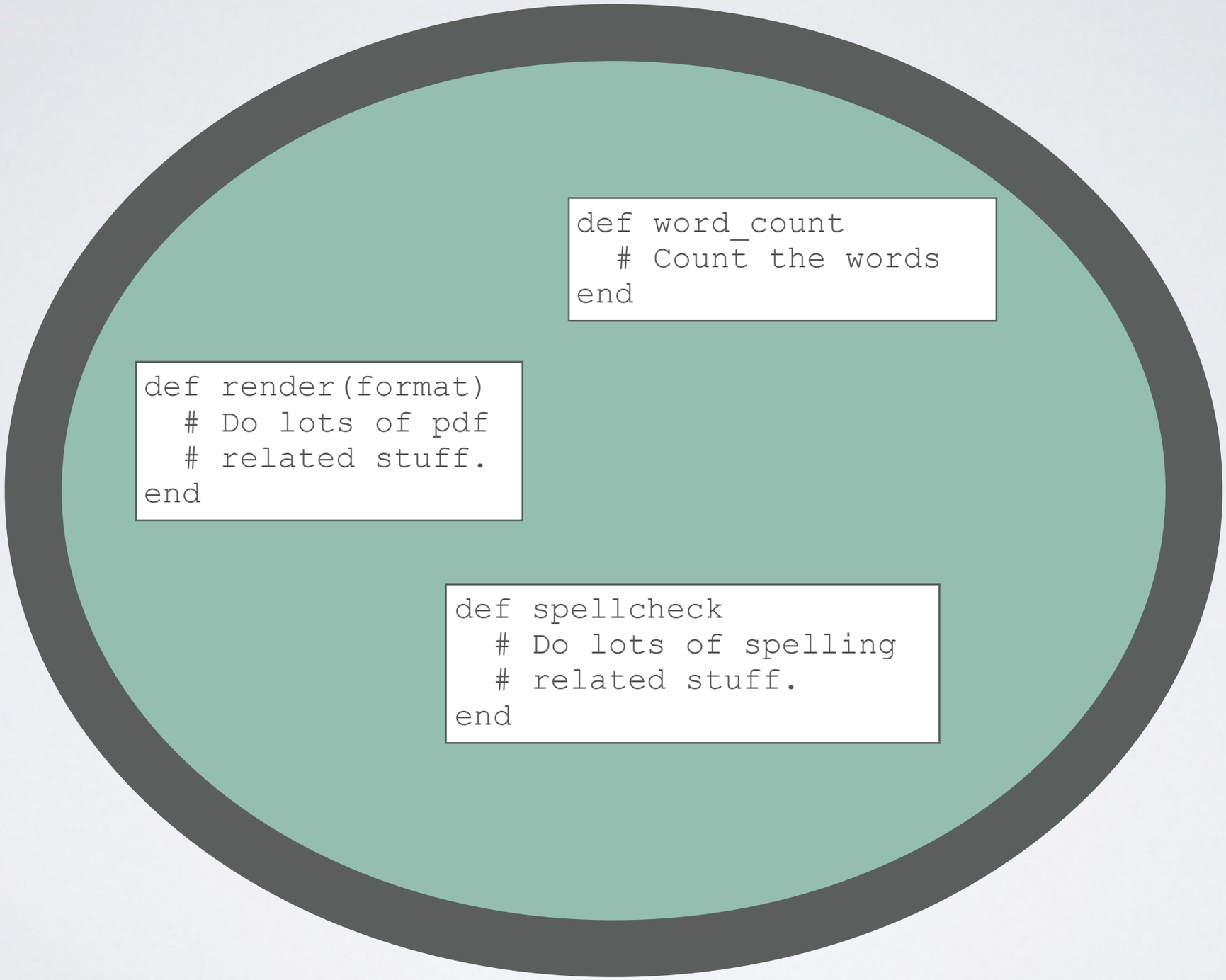
INFORMATION SHARING?

```
(def books
    [{:title "Gateway", :author “Pohl"}
     {:title "Great Gatsby", :author "Fitzgerald"}
     {:title "Hobbit", :author “Tolkien"}
     {:title "1984", :author “Orwell"}
     {:title "2001", :author “Clarke”}])

(defn find-books-by-author [author]
  (filter #(= (:author %) author) books))
```

INFORMATION SHARING?

title	author
Gateway	Pohl
Great Gatsby	Fitzgerald
Hobbit	Tokien
1984	Orwell
2001	Clarke

OBJECT ORIENTED?



FUNCTIONAL?

```
(defn count-words(doc)
  ;; Count the words in a doc
   )
```

```
(defn render(doc)
  ;; Render doc
  )
```

```
(defn spellcheck(doc)
  ;; Spellcheck the doc
  )
```

title:	Tale of Two Cities
author:	Dickens
text:	It was the best of times…

A MIX?

```
(defn count-words(doc)
  ;; Count the words in a doc
   )
```

```
(defn render(doc)
  ;; Render doc
  )
```

```
(defn spellcheck(doc)
  ;; spellcheck the doc
  )
```

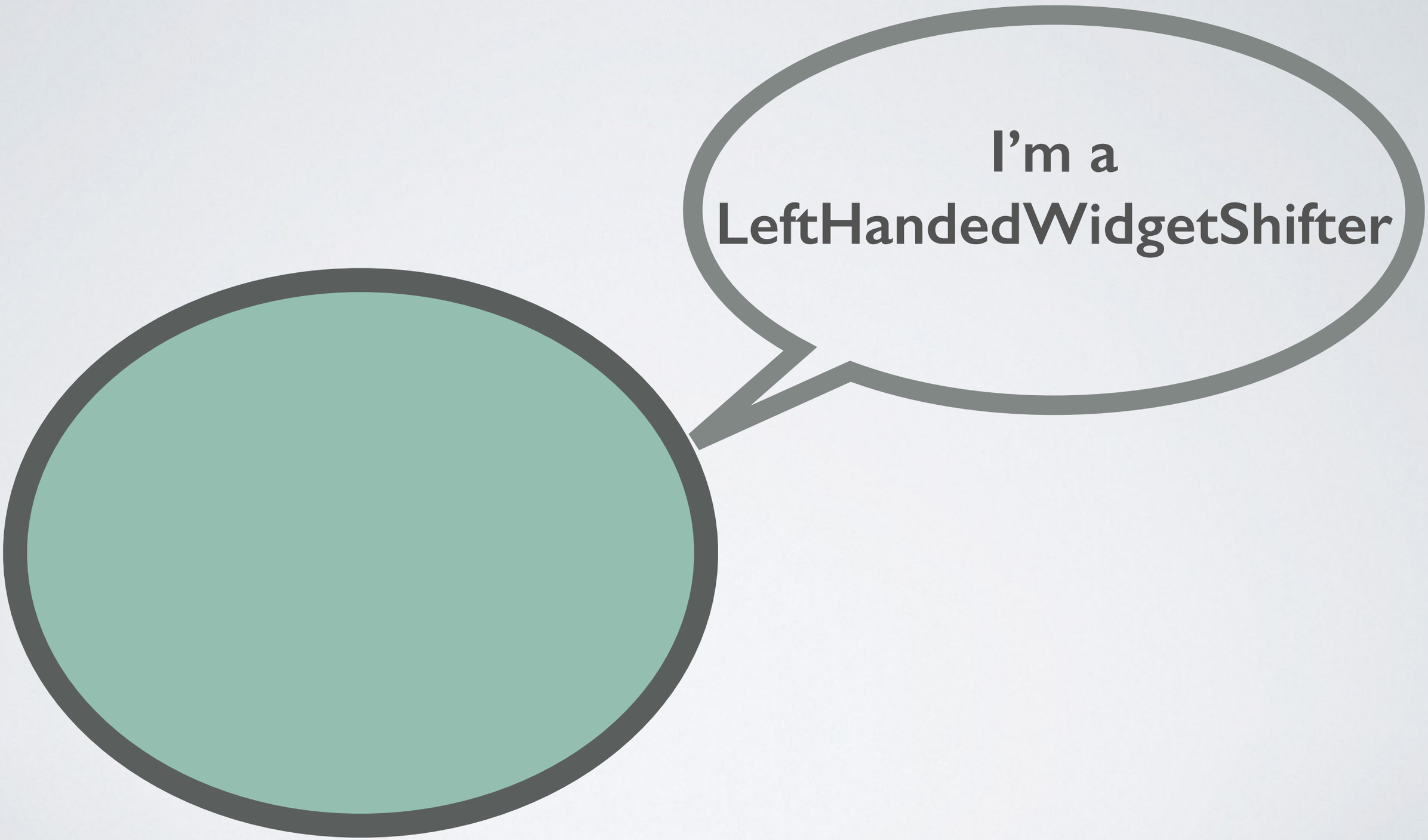
title:	Tale of Two Cities
author:	Dickens
text:	It was the best of times…

```
def word_count
  # Count the words
end
```

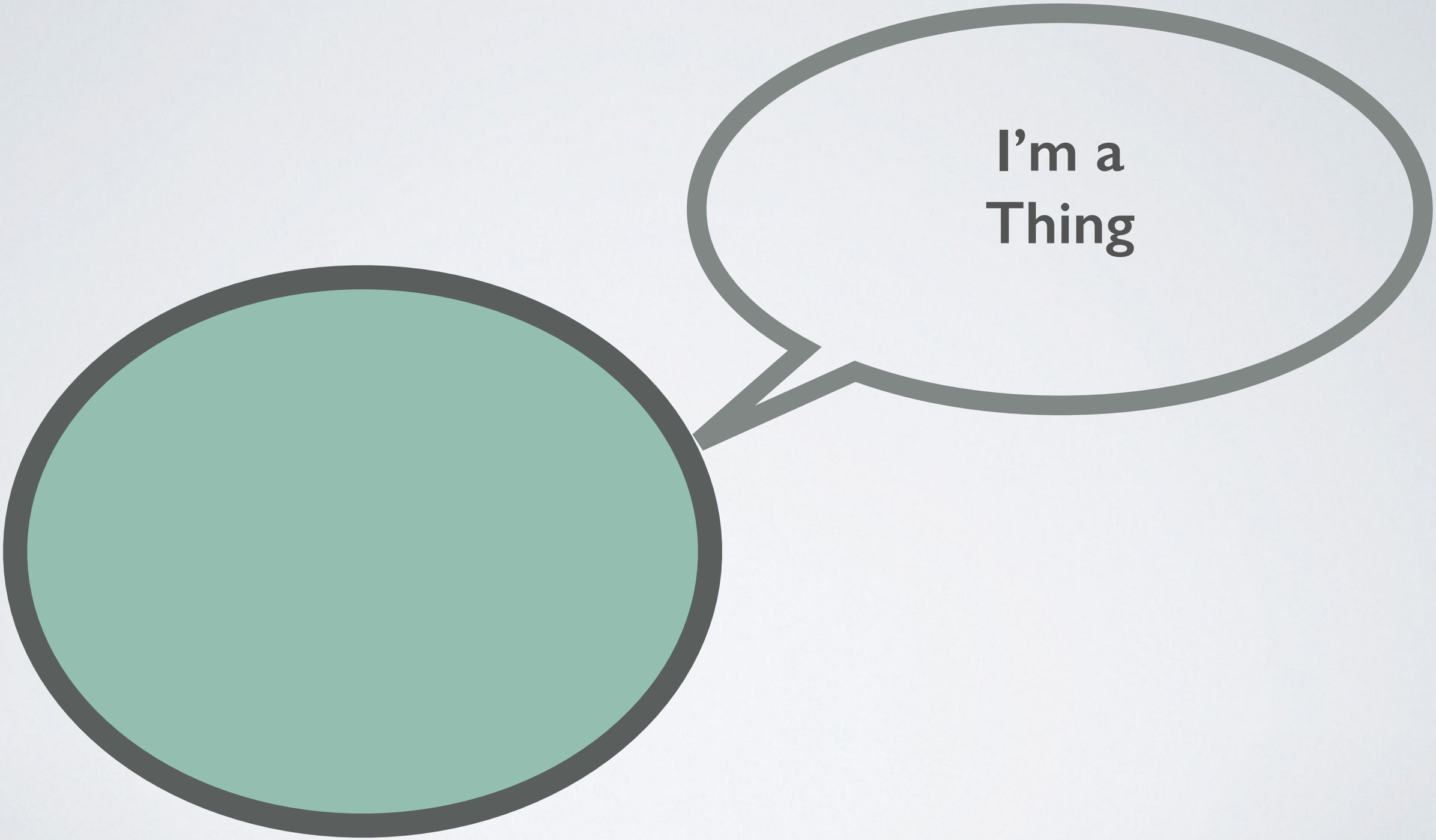
```
def render(format)
  # Do lots of pdf
  # related stuff.
end
```

```
def spellcheck
  # Do lots of spelling
  # related stuff.
end
```

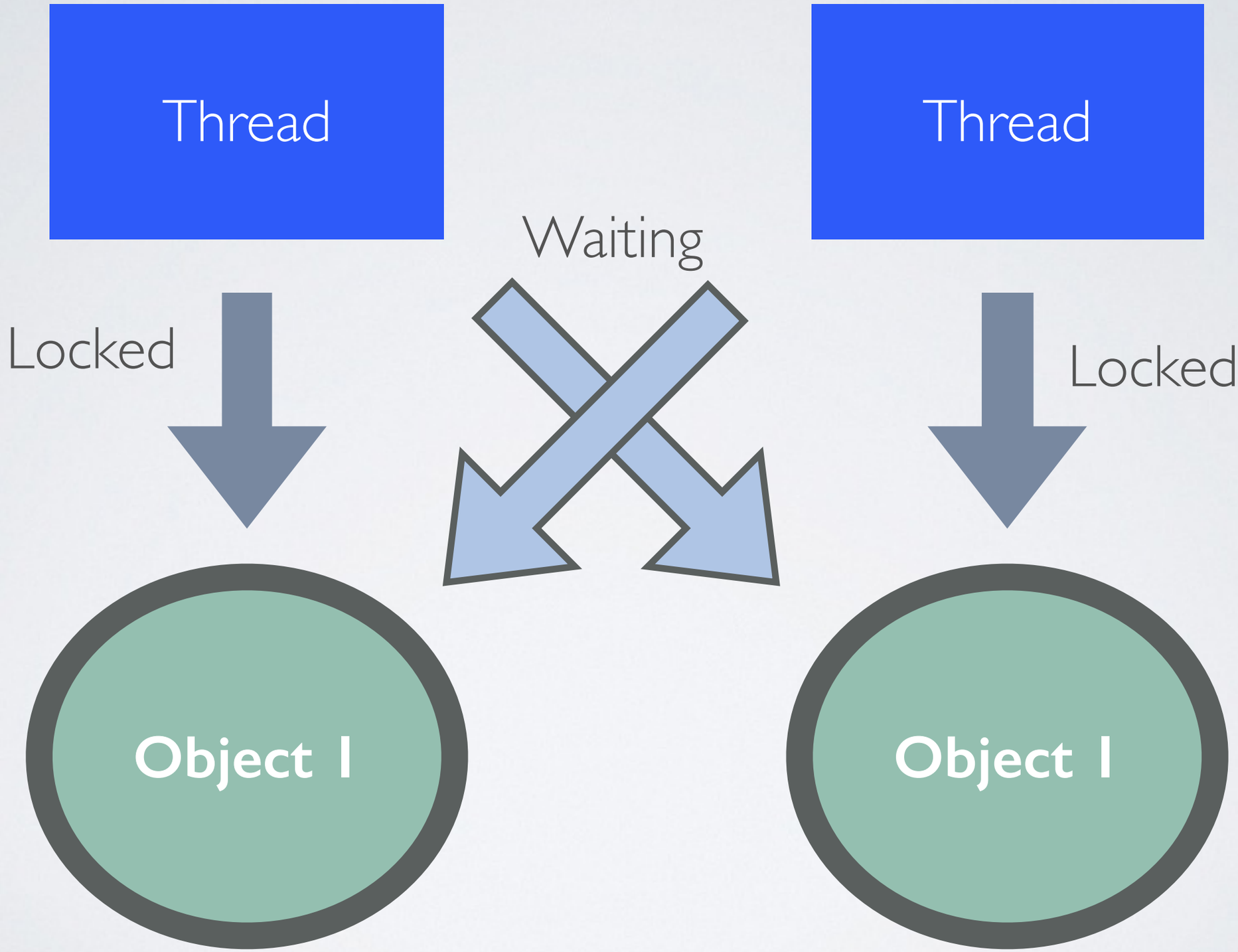
DO YOU NEED TYPES?



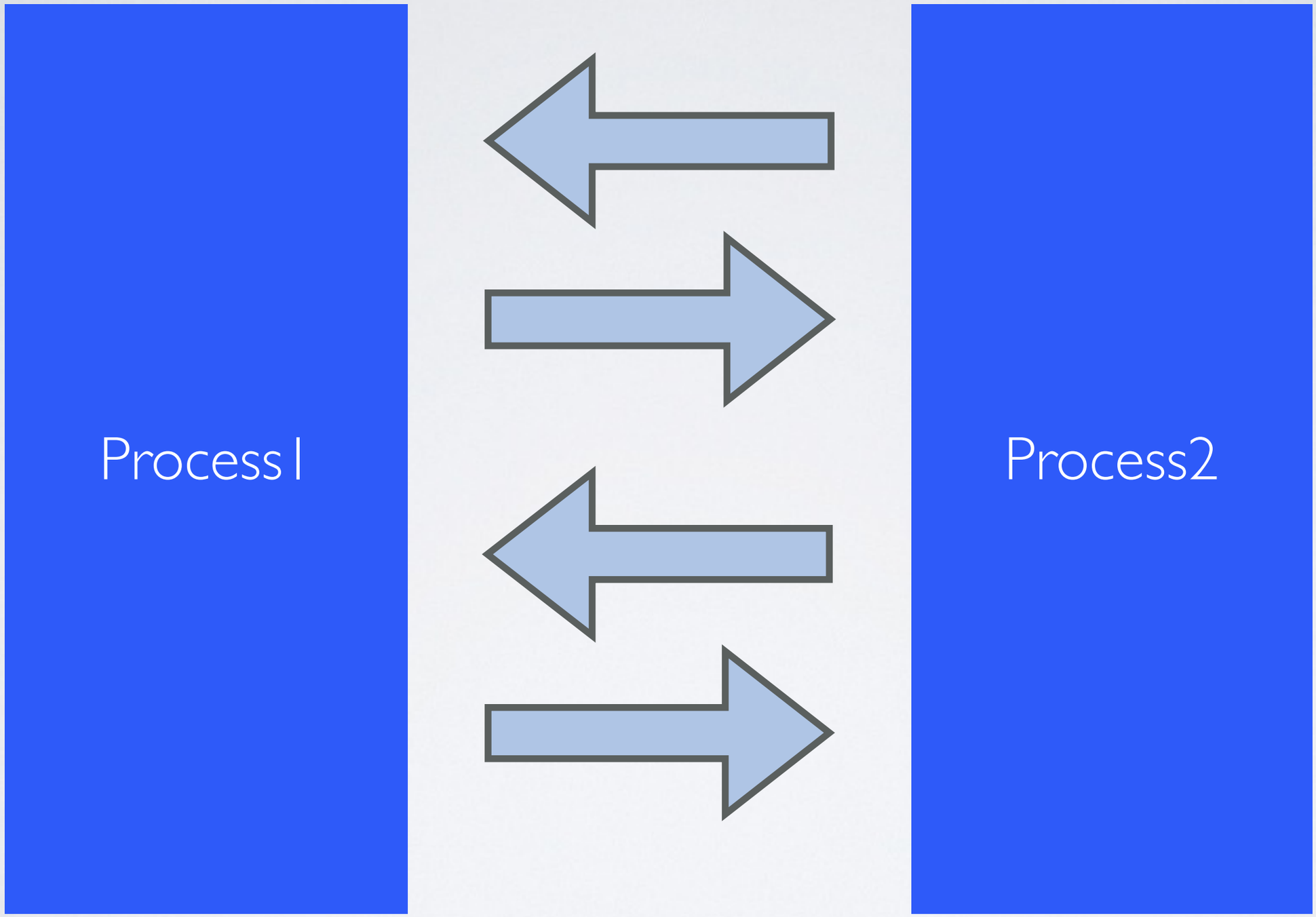
DO YOU NEED TYPES?



CONCURRENCY?



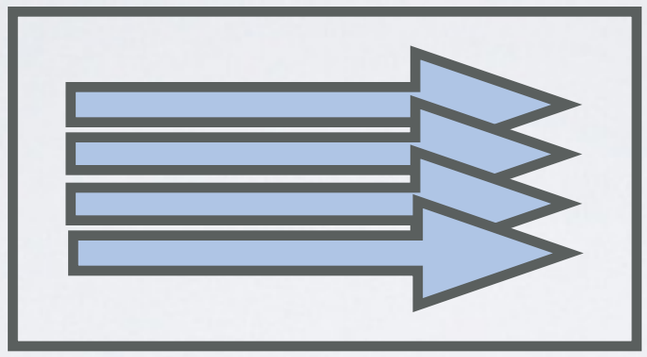
CSP?

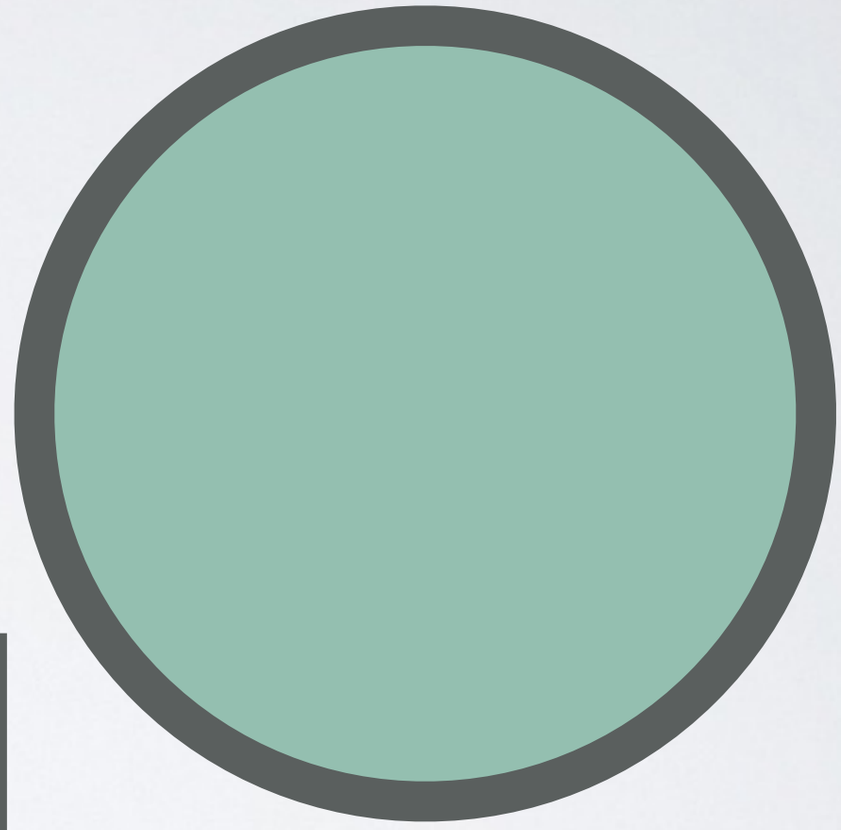


TRANSACTIONS?

Thread1

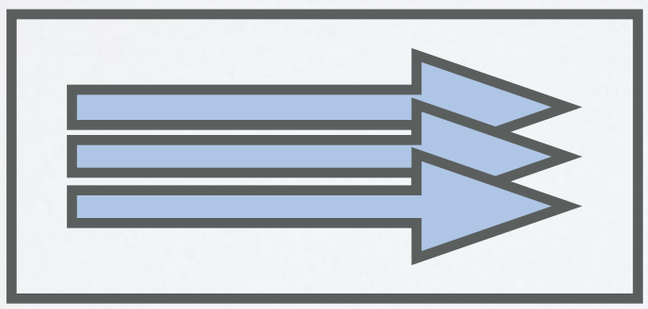
Transaction1





Thread2

Transaction2



MUTABLE STATE?

```
fred = Employee.new("Fred Smith", 50000)

fred.salary = 100000
```

MUTABLE STATE?

```
(def fred {:name "Fred Smith", :salary 50000})

(def new-fred (assoc fred :salary 100000))
```

WHICH LANGUAGE?

WHICH LANGUAGE?

QUESTIONS?

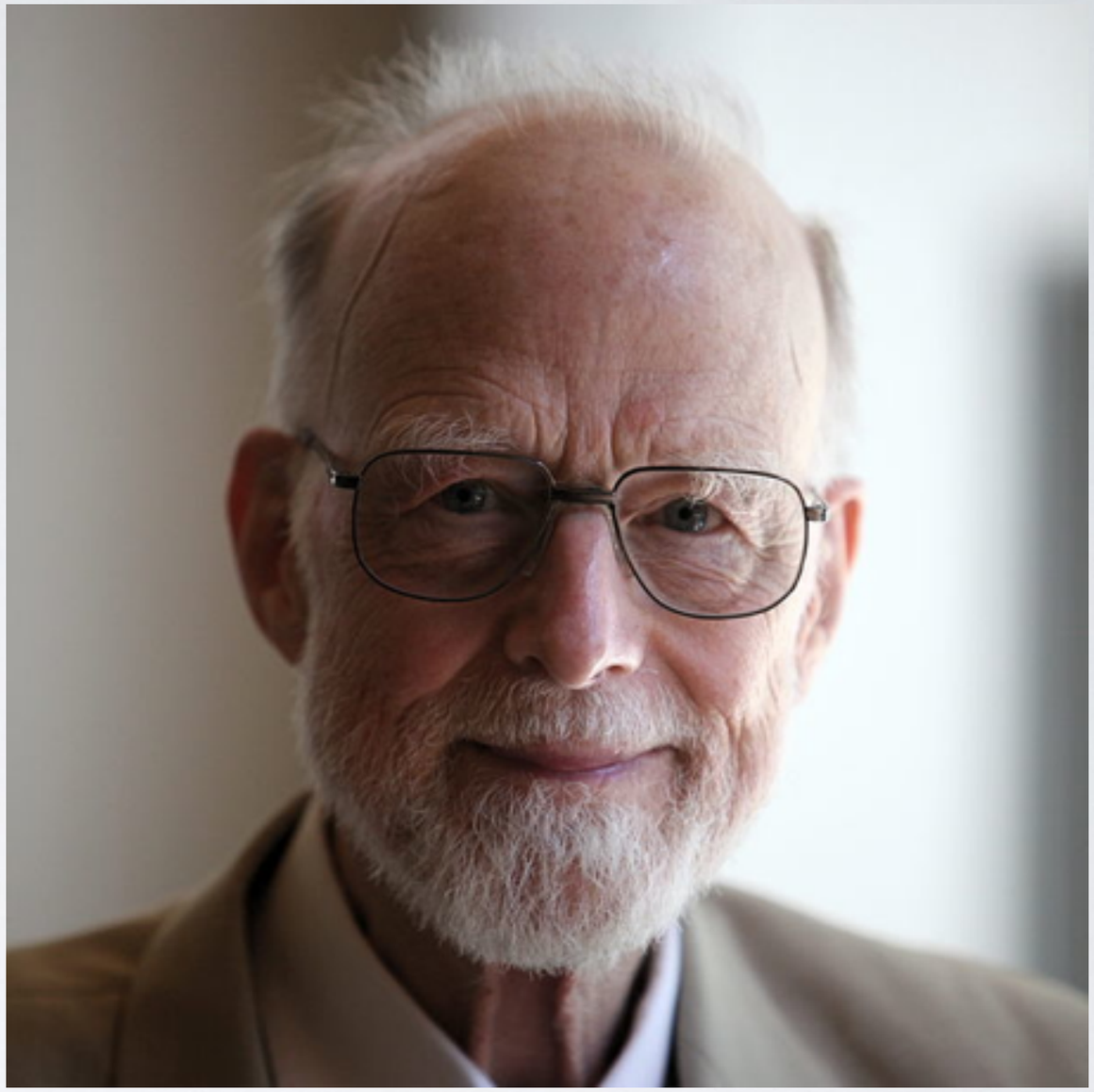
WHICH LANGUAGE?

ANSWERS!

YOUR
PROGRAMMINGLANGUAGE
IS GOING TO

DIE

“I DON'T KNOW WHAT THE LANGUAGE OF THE YEAR 2000 WILL LOOK LIKE, BUT I KNOW IT WILL BE CALLED FORTRAN”



TONY HOARE, 1982

“I DON'T KNOW WHAT THE LANGUAGE OF THE YEAR 2030 WILL LOOK LIKE,

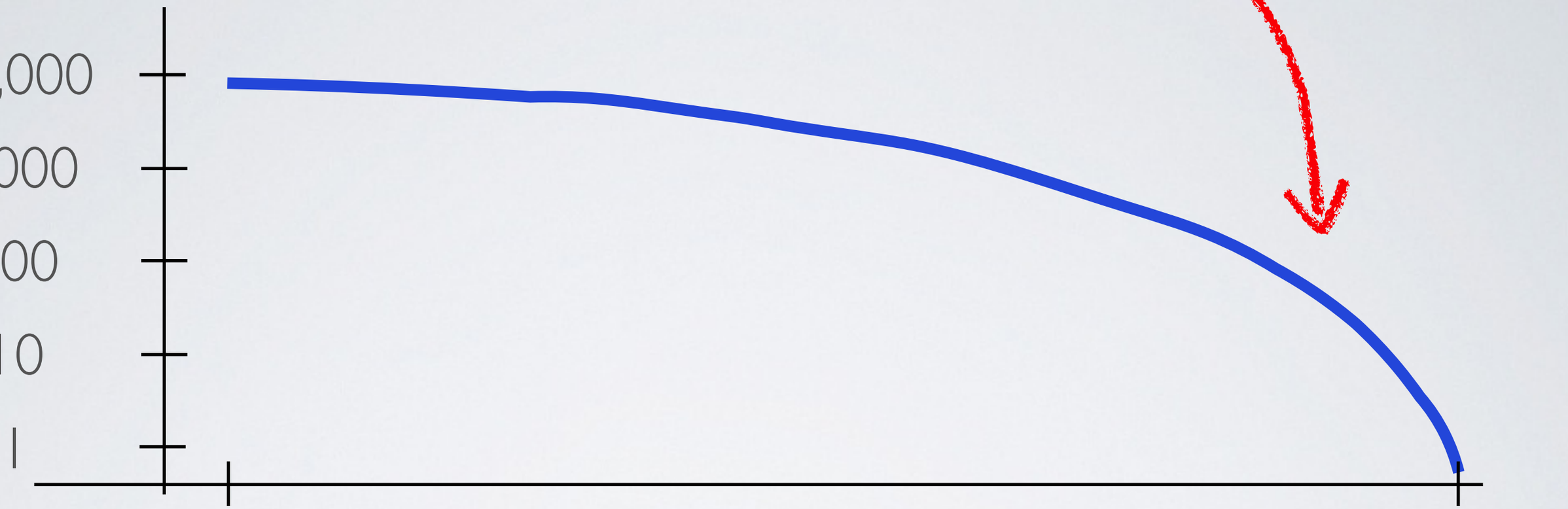
RUSS OLSEN, 2014

“I DON'T KNOW WHAT THE LANGUAGE OF THE YEAR 2030 WILL LOOK LIKE

BUT I DO KNOW WHERE TO LOOK **FOR** IT

RUSS OLSEN, 2014

Here!



Most popular

100th most popular

Simplified from : Meyerovich & Rabkin 2013

CHANGE HAPPENS

TO

THE MAINSTREAM

JAMES H. CARROT, 2013

PROGRAMMING
IN
INTERESTING TIMES

@russolsen

REFERENCES

- The Tony Hoare Fortran quote is from http://c2.com/cgi/wiki?FutureOfProgrammingLanguages
- The Rob Pike quote is from http://commandcenter.blogspot.com/2012/06/less-is-exponentially-more.html
- The programming language stats came from *Empirical Analysis of Programming Language Adoption* by Meyerovich & Rabkin, retrieved from http://www.cs.princeton.edu/~asrabkin/papers/oopsla13.pdf on May 21, 2014.
- A similar and interesting paper is *A Study of Language Usage Evolution in Open Source Software* by Siim Karus and Harald Gall, retrieved from http://arxiv.org/pdf/1102.2262v1.pdf on May 13, 2014.
- The James H. Carrott quote is from the book *Vintage Tomorrows*, http://www.amazon.com/Vintage-Tomorrows-Historian-Steampunk-Technology/dp/1449337996

IMAGE CREDITS

- Volcano: http://en.wikipedia.org/wiki/File:Eruption_1954_Kilauea_Volcano.jpg
- Tony Hoare: http://en.wikipedia.org/wiki/File:Sir_Tony_Hoare_IMG_5125.jpg
- Tombstone: https://s3.amazonaws.com/fotor.onlineresource.w/458d60c37ec94a8c8d72bb96ddcaf465/458d60c37ec94a8c8d72bb96ddcaf465_p_400.png
- Dr. Peter Venkman: http://www.spoutnik.info/uploads/1369597868_ghostbusters-bill-murray-peter-venkman-HD-Wallpapers.jpg
- Compass: http://upload.wikimedia.org/wikipedia/commons/8/8d/Simple_compass_rose.svg
- Compiler diagram: http://img2.wikia.nocookie.net/__cb20070117104131/uncyclopedia/images/thumb/3/31/Compiler_io.PNG/300px-Compiler_io.PNG
- Conference: http://upload.wikimedia.org/wikipedia/commons/e/e6/WordCamp_2011_Bulgaria.jpg
- Rob Pike: http://en.wikipedia.org/wiki/File:Rob-pike.jpg

IMAGE CREDITS

- Grass: http://en.wikipedia.org/wiki/File:Grass_dsc08672-nevit.jpg
- Crowd: http://upload.wikimedia.org/wikipedia/commons/8/8d/Folla_in_piazza_del_campo.jpg
- Zombie hand: http://legaciesbydesign.com/wp-content/uploads/2012/06/rising-from-grave001.jpg
- Iceberg: http://upload.wikimedia.org/wikipedia/commons/5/5d/Titanic_iceberg.jpg