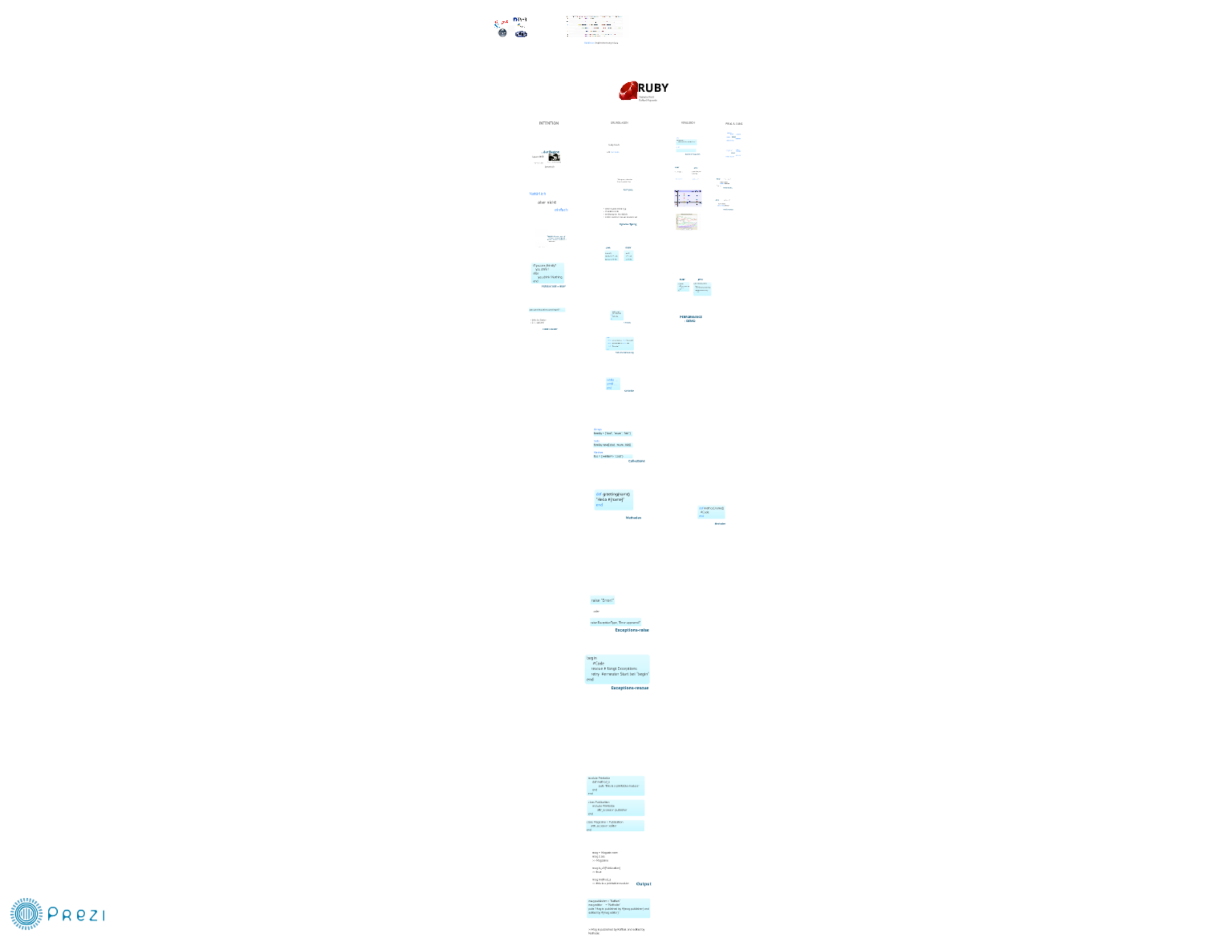

RUBY
INTENTION
SIMULATION
RESEARCH
PRACTICE
PERFORMANCE
- BENCHMARK
Collectionen
Methoden
Exceptions-raise
Exceptions-rescue
Output

```
27.     private static void bubbleSort(int[] unsortedArray, int length) {
28.         int temp, counter, index;
29.
30.         for(counter=0; counter<length-1; counter++) {
31.             for(index=0; index<length-1-counter; index++) {
32                 if(unsortedArray[index] > unsortedArray[index+1]) {
33.                     temp = unsortedArray[index];
34.                     unsortedArray[index] = unsortedArray[index+1];
35.                     unsortedArray[index+1] = temp;
```

Bubblesort Implementierung in Java

RUBY

Nathalie Kröll
Raffael Majewski

INTENTION

GRUNDLAGEN

VERGLEICH

PROS & CONS

INTENTION

...der Beginn

Japan (1995)

Yukihiro Matsumoto

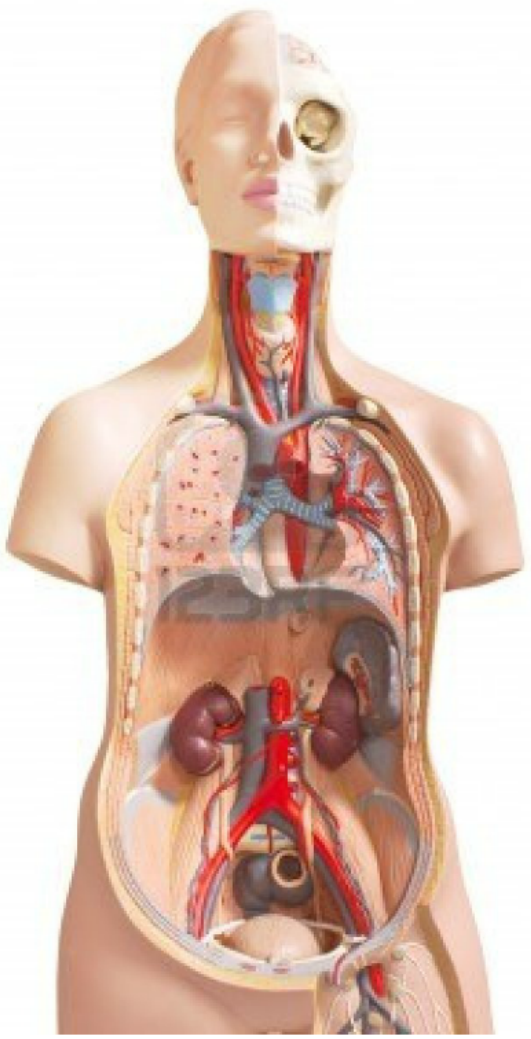
multifunkional

dynamisch

Natürlich

aber nicht

einfach



Human anatomy dummy (us.123rf.com)

"Ruby wirkt simpel, aber ist innen sehr komplex, genau wie der menschliche Körper."

-Yukihiro Matsumoto

```
if you.are_thirsty?
   you.drink!
else
      you.drink! Nothing
end
```

PSEUDOCODE in RUBY

```
puts greeting unless greeting.nil?
```

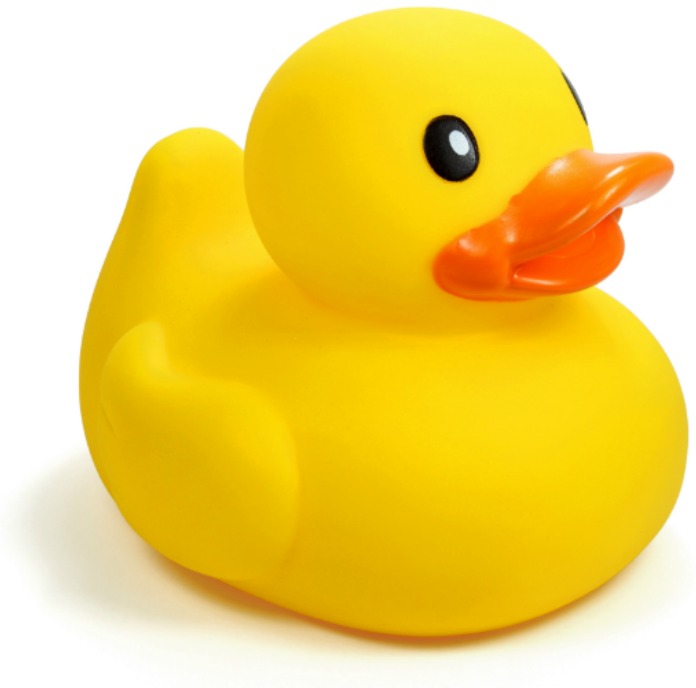
- Einfache Syntax
- () [] ; optional

CODE in RUBY

GRUNDLAGEN

>ruby foo.rb

wird interpretiert



"Was quakt wie eine Ente, ist eine Ente."

Duck Typing

- Werte haben einen Typ
- Variablen nicht
- Erhöhung der Flexibilität
- Fehler tauchen erst zur Laufzeit auf

Dynamic Typing

JAVA

```
int a = 5;

boolean b = true;

double c = 5.43;
```

RUBY

```
a = 5

b = true

c = 5.43
```

Ducktyping - DEMO

```
if   bottle.is_empty?
     take_another bottle
else
     bottle.drink!

end
```

if Anweisung

```
case
  when  password.length < 6  then  "PW too short"
  when  password.length < 10 then  "OK"
  else  "PW too long"
end
```

Fallunterscheidung

while .....
until .....
end

Schleifen

Arrays

```
family = ["dad", "mum", "kid"]
```

Sets

```
family.new([:dad, :mum, :kid])
```

Hashes

```
foo = {:winter=> "cold"}
```

Collections

```
def greeting(name)
"Hello #{name}"
end
```

Methoden

```
def method_name()
  #Code
end
```

Methoden

```
raise "Error!"
```

oder

```
raise ExceptionType, "Error appeared!"
```

Exceptions-raise

```
begin
      #Code
    rescue # fängt Exceptions
    retry  #erneuter Start bei "begin"
end
```

Exceptions-rescue

```
module Printable
    def method_a
        puts "this is a printable module"
    end
end
```

```
class Publication
    include Printable
        attr_accessor :publisher
end
```

```
class Magazine < Publication
    attr_accessor :editor
end
```

```
module Printable
    def method_a
        puts "this is a printable module"
    end
end
```

```
class Publication
    include Printable
```

```
puts "this is a printable module"
    end
end
```

```
class Publication
    include Printable
        attr_accessor :publisher
end
```

```
class Magazine < Publication
    attr_accessor :editor
```

```
    include Printable
        attr_accessor :publisher
end
```

```
class Magazine < Publication
    attr_accessor :editor
end
```

```
mag = Magazin.new
mag.class
>> Magazine

mag.is_a?(Publication)
>> true

mag.method_a
>> this is a printable module
```

Output

```
mag.publisher = "Raffael"
mag.editor    = "Nathalie"
puts "Mag is published by #{mag.publisher} and
edited by #{mag.editor}."
```

>>Mag is published by Raffael and edited by Nathalie.

VERGLEICH

JAVA

```
Class Example {
    public static void main (String[] args) {}
}
```

RUBY

das leere Programm

RUBY

>ruby foo.rb

interpretiert

JAVA

>javac foo.java
>java foo

kompiliert

Ruby interpretiert

Fehler während
Runtime behandelt

flexibel

Performance

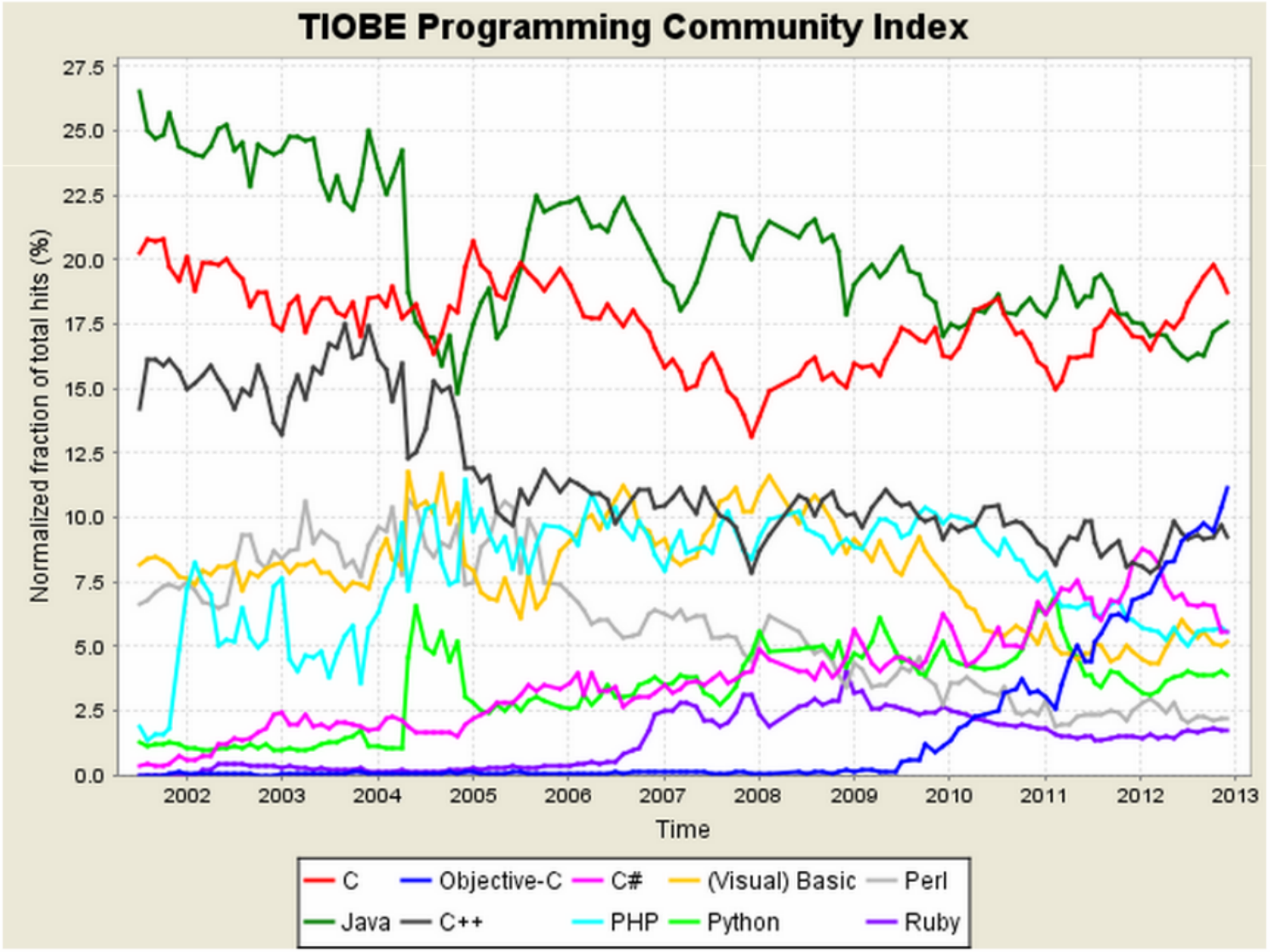
JAVA kompiliert

Fehler während
Kompilierung behandelt

Performance

Position Dec 2012	Position Dec 2011	Delta in Position	Programming Language	Ratings Dec 2012	Delta Dec 2011
1	2	↑	C	18.696%	+1.64%
2	1	↓	Java	17.567%	+0.01%
3	5	↑↑	Objective-C	11.116%	+4.31%
4	3	↓	C++	9.203%	+0.95%
5	4	↓	C#	5.547%	-2.66%
6	6	=	PHP	5.541%	-0.46%
7	7	=	(Visual) Basic	5.174%	+0.42%
8	8	=	Python	3.848%	+0.36%
9	9	=	Perl	2.174%	-0.30%
10	11	↑	Ruby	1.728%	+0.23%

3	5	⬆⬆	Objective-C	11.116%	+4.31%
4	3	⬇	C++	9.203%	+0.95%
5	4	⬇	C#	5.547%	-2.66%
6	6	=	PHP	5.541%	-0.46%
7	7	=	(Visual) Basic	5.174%	+0.42%
8	8	=	Python	3.848%	+0.36%
9	9	=	Perl	2.174%	-0.30%
10	11	⬆	Ruby	1.728%	+0.23%


TIOBE Programming Community Index
Normalized fraction of total hits (%)
27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
Time
C
Objective-C
C#
(Visual) Basic
Perl
Java
C++
PHP
Python
Ruby

RUBY

```
def len (list)
    x=0
    list.each do |element|
        x+=1
    end
end
```

JAVA

```
public static int len (List list)
 {
    int x = 0;
    Iterator listIterator = list.iterator();

    while(listIterator.hasNext()){
        x+=1;}
}
```

PERFORMANCE - DEMO

PROS & CONS

weniger Code

dynamisch

natürlich

flexibel

PROS

multifunktional

einfach zu lesen

Cons

langsamer

wenige Entwickler

komplex

Einstiegssprache?