





# Erik Corry

## ## A career in language runtimes

- \* 1998: BETA: A pioneering OO language from Scandinavia
- \* 2004: OOVm: Smalltalk for embedded systems
- \* 2007: V8: The JavaScript engine that powers Chrome and Node.js
  - Irregexp: JITted regular expressions in JavaScript
- \* 2010: Dart: A new language for the web (and mobile)
  - Dart2js: A Dart to JavaScript compiler
  - Dart VM: A Dart virtual machine
  - Dartino: A Dart VM for microcontrollers
- \* 2017: Oilpan: A garbage collector for C++ (Chrome)
- \* 2018: Toit: A high level language for IoT
- \* 2024: Working on V8 in the cloud

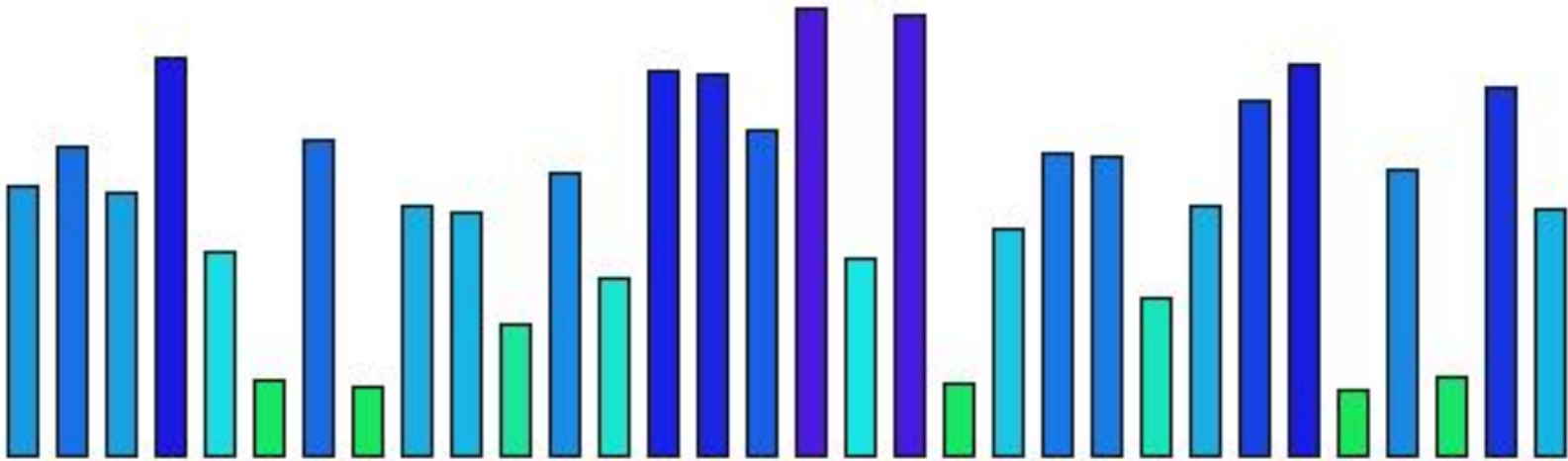
~  
~  
~  
~

```
kind = REGEXP
name = [cga]tatcccgac|[act]gcttacg
compiler = unknown
address = 0x7f0194b904c1
```

|                |    |              |                             |
|----------------|----|--------------|-----------------------------|
| 0x7f0194b90567 | 27 | 83fff9       | cmpl rdi,0xf9               |
| 0x7f0194b9056a | 2a | 0f8d0f030000 | jge 0x7f0194b9087f <+0x33f> |
| 0x7f0194b90570 | 30 | 8b143e       | movl rdx,[rsi+rdi*1]        |
| 0x7f0194b90573 | 33 | b8e8ecfdff   | movl rax,0xffffdece8        |
| 0x7f0194b90578 | 38 | 4823c2       | REX.W andq rax,rdx          |
| 0x7f0194b9057b | 3b | 3d60646174   | cmp rax,0x74616460          |
| 0x7f0194b90580 | 40 | 0f8406000000 | jz 0x7f0194b9058c <+0x4c>   |
| 0x7f0194b90586 | 46 | 4883c701     | REX.W addq rdi,0x1          |
| 0x7f0194b9058a | 4a | ebdb         | jmp 0x7f0194b90567 <+0x27>  |

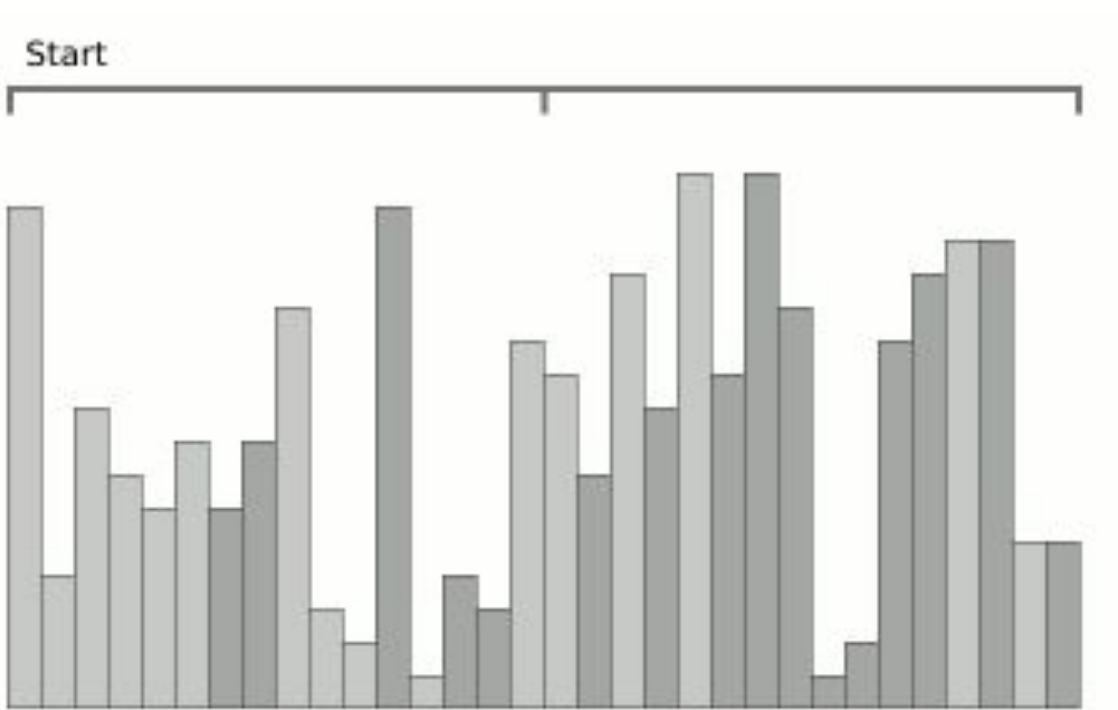
~  
~  
~  
~  
~

# Quicksort - not a stable sorting algorithm



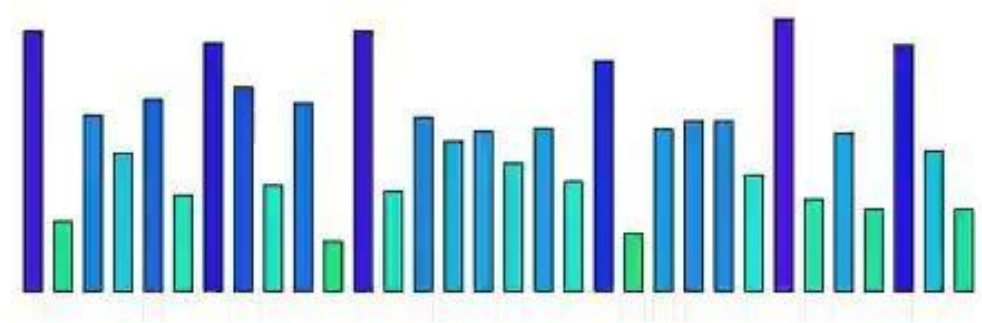
| Name ↕               | Best ↕     | Average ↕  | Worst ↕                  | Memory ↕                 | Stable ↕ |           |
|----------------------|------------|------------|--------------------------|--------------------------|----------|-----------|
| Quicksort            | $n \log n$ | $n \log n$ | $n^2$                    | $\log n$                 | No       |           |
| Merge sort           | $n \log n$ | $n \log n$ | $n \log n$               | $n$                      | Yes      |           |
| In-place merge sort  | —          | —          | $n \log^2 n$             | 1                        | Yes      |           |
| Introsort            | $n \log n$ | $n \log n$ | $n \log n$               | $\log n$                 | No       |           |
| Heapsort             | $n \log n$ | $n \log n$ | $n \log n$               | 1                        | No       |           |
| Insertion sort       | $n$        | $n^2$      | $n^2$                    | 1                        | Yes      |           |
| Block sort           | $n$        | $n \log n$ | $n \log n$               | 1                        | Yes      | Irregular |
| Timsort              | $n$        | $n \log n$ | $n \log n$               | $n$                      | Yes      | Irregular |
| Selection sort       | $n^2$      | $n^2$      | $n^2$                    | 1                        | No       |           |
| Cubesort             | $n$        | $n \log n$ | $n \log n$               | $n$                      | Yes      |           |
| Shellsort            | $n \log n$ | $n^{4/3}$  | $n^{3/2}$                | 1                        | No       |           |
| Bubble sort          | $n$        | $n^2$      | $n^2$                    | 1                        | Yes      |           |
| Exchange sort        | $n^2$      | $n^2$      | $n^2$                    | 1                        | No       |           |
| Tree sort            | $n \log n$ | $n \log n$ | $n \log n$<br>(balanced) | $n$                      | Yes      |           |
| Cycle sort           | $n^2$      | $n^2$      | $n^2$                    | 1                        | No       |           |
| Library sort         | $n \log n$ | $n \log n$ | $n^2$                    | $n$                      | No       |           |
| Patience sorting     | $n$        | $n \log n$ | $n \log n$               | $n$                      | No       |           |
| Smoothsort           | $n$        | $n \log n$ | $n \log n$               | 1                        | No       |           |
| Strand sort          | $n$        | $n^2$      | $n^2$                    | $n$                      | Yes      |           |
| Tournament sort      | $n \log n$ | $n \log n$ | $n \log n$               | $n^{\lfloor 11 \rfloor}$ | No       |           |
| Cocktail shaker sort | $n$        | $n^2$      | $n^2$                    | 1                        | Yes      |           |
| Comb sort            | $n \log n$ | $n^2$      | $n^2$                    | 1                        | No       |           |
| Gnome sort           | $n$        | $n^2$      | $n^2$                    | 1                        | Yes      |           |
| Odd–even sort        | $n$        | $n^2$      | $n^2$                    | 1                        | Yes      |           |

| Name ↕               | Best ↕     | Average ↕  | Worst ↕                  | Memory ↕   | Stable ↕ |           |
|----------------------|------------|------------|--------------------------|------------|----------|-----------|
| Quicksort            | $n \log n$ | $n \log n$ | $n^2$                    | $\log n$   | No       |           |
| Merge sort           | $n \log n$ | $n \log n$ | $n \log n$               | $n$        | Yes      |           |
| In-place merge sort  | —          | —          | $n \log^2 n$             | 1          | Yes      |           |
| Introsort            | $n \log n$ | $n \log n$ | $n \log n$               | $\log n$   | No       |           |
| Heapsort             | $n \log n$ | $n \log n$ | $n \log n$               | 1          | No       |           |
| Insertion sort       | $n$        | $n^2$      | $n^2$                    | 1          | Yes      |           |
| Block sort           | $n$        | $n \log n$ | $n \log n$               | 1          | Yes      | Irregular |
| Timsort              | $n$        | $n \log n$ | $n \log n$               | $n$        | Yes      | Irregular |
| Selection sort       | $n^2$      | $n^2$      | $n^2$                    | 1          | No       |           |
| Cubesort             | $n$        | $n \log n$ | $n \log n$               | $n$        | Yes      |           |
| Shellsort            | $n \log n$ | $n^{4/3}$  | $n^{3/2}$                | 1          | No       |           |
| Bubble sort          | $n$        | $n^2$      | $n^2$                    | 1          | Yes      |           |
| Exchange sort        | $n^2$      | $n^2$      | $n^2$                    | 1          | No       |           |
| Tree sort            | $n \log n$ | $n \log n$ | $n \log n$<br>(balanced) | $n$        | Yes      |           |
| Cycle sort           | $n^2$      | $n^2$      | $n^2$                    | 1          | No       |           |
| Library sort         | $n \log n$ | $n \log n$ | $n^2$                    | $n$        | No       |           |
| Patience sorting     | $n$        | $n \log n$ | $n \log n$               | $n$        | No       |           |
| Smoothsort           | $n$        | $n \log n$ | $n \log n$               | 1          | No       |           |
| Strand sort          | $n$        | $n^2$      | $n^2$                    | $n$        | Yes      |           |
| Tournament sort      | $n \log n$ | $n \log n$ | $n \log n$               | $n^{[11]}$ | No       |           |
| Cocktail shaker sort | $n$        | $n^2$      | $n^2$                    | 1          | Yes      |           |
| Comb sort            | $n \log n$ | $n^2$      | $n^2$                    | 1          | No       |           |
| Gnome sort           | $n$        | $n^2$      | $n^2$                    | 1          | Yes      |           |
| Odd–even sort        | $n$        | $n^2$      | $n^2$                    | 1          | Yes      |           |



Thanks to BonzaiThePenguin

# Mergesort with $\frac{1}{4}$ sized temporary buffer



```
kind = REGEXP
name = [cga]tatcccgac|[act]gcttacg
compiler = unknown
address = 0x7f0194b904c1
```

|                |    |              |                             |
|----------------|----|--------------|-----------------------------|
| 0x7f0194b90567 | 27 | 83fff9       | cmpl rdi,0xf9               |
| 0x7f0194b9056a | 2a | 0f8d0f030000 | jge 0x7f0194b9087f <+0x33f> |
| 0x7f0194b90570 | 30 | 8b143e       | movl rdx,[rsi+rdi*1]        |
| 0x7f0194b90573 | 33 | b8e8ecfdff   | movl rax,0xffffdece8        |
| 0x7f0194b90578 | 38 | 4823c2       | REX.W andq rax,rdx          |
| 0x7f0194b9057b | 3b | 3d60646174   | cmp rax,0x74616460          |
| 0x7f0194b90580 | 40 | 0f8406000000 | jz 0x7f0194b9058c <+0x4c>   |
| 0x7f0194b90586 | 46 | 4883c701     | REX.W addq rdi,0x1          |
| 0x7f0194b9058a | 4a | ebdb         | jmp 0x7f0194b90567 <+0x27>  |

~  
~  
~  
~  
~

```
$ node sort.js  
[ 'dog', 'cat', 'hen', 'duck', 'goat', 'horse' ]  
$
```

```
TERMINAL /bin/bash ln :1/3  
var a = ["dog", "cat", "hen", "horse", "duck", "goat"];  
  
a.sort((a, b) => a.length - b.length);  
  
console.log(a);  
~  
~  
~  
~  
sort.js jav... utf-8[unix] 20% ln :1/5  
:term
```

```
$ node order.js  
foo: 1  
bar: 42  
$
```

TERMINAL /bin/bash

ln :1/4

```
var m = {  
  "foo": 1,  
  "bar": 42  
}  
  
for (var k in m) { console.log(k + ": " + m[k]) }
```

~  
~  
~

order.js jav... utf-8[unix] 50% ln :3/6

:term

```
$ node order.js
foo: 1
bar: 42
$ node order.js
2: 103
foo: 1
bar: 42
$
```

```
TERMINAL /bin/bash ln:1/8≡
var m = {
  "foo": 1,
  "bar": 42,
  2: 103
}

for (var k in m) { console.log(k + ": " + m[k]) }
~
~

NORMAL  main! order.js jav... utf-8[unix] 57% ln:4/7≡%:8
```

```
$ node order.js
2: 103
foo: 1
bar: 42
$ node order.js
2: 103
foo: 1
bar: 42
$
```

TERMINAL /bin/bash

ln :1/12

```
var m = {
  "foo": 1,
  "bar": 42,
  "2": 103
}

for (var k in m) { console.log(k + ": " + m[k]) }
```

~  
~

order.js

jav...

utf-8[unix]

57% ln :4/7

```
$ go run order.go  
map[bar:42 foo:1]  
$
```

TERMINAL /bin/bash

ln :1/3

```
package main  
import "fmt"  
func main() {  
    m := map[string]int{  
        "foo": 1,  
        "bar": 42,  
    }  
    fmt.Println(m)  
}
```

order.go

go utf-8[unix] 11% ln :1/9

:term

```
package main
```

```
import "fmt"
```

```
func main() {  
    for i := 1; i <= 10; i++ {  
        fmt.Println(DoesItAlias(i))  
    }  
}
```

```
func DoesItAlias(length int) bool {  
    a := []int{}  
    for i := 0; i < length; i++ {  
        a = append(a, i)  
    }  
    b := append(a[:], 42)  
    a = append(a, 103)  
    return a[length] == b[length]  
}
```

```
NORMAL  ↵ main  alias.go  go  utf-8[unix]  5%  ln:1/19  %:1
```

```
"alias.go" 19L, 285B
```

- A) true, true, true, true, true, true, true, true, true, true
- B) false, false, true, false, true, true, true, false, true, true
- C) false, false, false, false, false, false, false, false, false, false
- D) true, true, true, false, false, false, false, true, true, true

~  
~  
~  
~  
~

alias.txt tex... utf-8[unix] 44W 25% ln:1/4≡%:1

```
func DoesItAlias(length int) bool {  
    a := []int{}  
    for i := 0; i < length; i++ {  
        a = append(a, i)  
    }  
    b := append(a[:], 42)  
    a = append(a, 103)  
    return a[length] == b[length]
```

}

NORMAL ↗ main! alias.go go utf-8[unix] 100% ln:19/19≡%:1

"alias.txt" 4L, 266B

- A) true, true, true, true, true, true, true, true, true, true
- B) false, false, true, false, true, true, true, false, true, true
- C) false, false, false, false, false, false, false, false, false, false
- D) true, true, true, false, false, false, false, true, true, true

alias.txt

tex...

utf-8[unix]

44W 25%

ln :1/4

```
func DoesItAlias(length int) bool {  
    a := []int{}  
    for i := 0; i < length; i++ {  
        a = append(a, i)  
    }  
    b := append(a[0:len(a)], 42)  
    a = append(a, 103)  
    return a[length] == b[length]  
}
```

NORMAL

main!

alias.go

go

utf-8[unix]

84%

ln :16/19

```
$ go run alias.go | fmt
false false true false true true true false true true
$
```

TERMINAL /bin/bash

ln :1/3

```
func DoesItAlias(length int) bool {
    a := []int{}
    for i := 0; i < length; i++ {
        a = append(a, i)
    }
    b := append(a[0:len(a)], 42)
    a = append(a, 103)
    return a[length] == b[length]
}
```

alias.go

go

utf-8[unix]

84% ln :16/19

```
$ go run alias.go | fmt
false false true false true true true false true true
$ go run alias.go | fmt
false false false false false false false false false
$
```

TERMINAL /bin/bash

ln :1/5

```
func DoesItAlias(length int) bool {
    a := []int{}
    for i := 0; i < length; i++ {
        a = append(a, i)
    }
    b := append(a[0:len(a):len(a)], 42)
    a = append(a, 103)
    return a[length] == b[length]
}
```

alias.go

go

utf-8[unix]

84% ln :16/19

> Undefined behavior" means that anything can happen. An implementation is  
> allowed to wrap the integer, throw an exception, reformat your hard drive,  
> etc.

Reddit, 2020

> the behavior of this action is undefined. It might work, it might not work,  
> it might crash, it might not crash, it might launch nuclear missiles and end  
> all humanity.

Reddit, 2024

> Who knows, the compiler writers might consider it to be an important  
> language-extension to be able to summon nasal demons by taking the sizeof an  
> incomplete type, or to be able to format a hard drive by simply running the C  
> compiler on a program missing a colon (I can never remember the arguments to  
> format programs, that actually sounds like a helpful thing to me... ;-).

John F. Woods, Usenet 1992

~

~

~

NORMAL

main

c++-undefined.mdx

mar...

utf-8[unix]

140W 6% ln:1/16 3:1

"c++-undefined.mdx" 16L, 771B



```
// Read array at index + offset (returns 0 if out of bounds).
int index(int* array, int length, int index, unsigned short offset) {
    if (index < 0) return 0;
    index += offset;
    if (index < 0) return 0;
    if (index >= length) return 0;
    return array[index];
}
```

~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~

**NORMAL** main! c++-bounds.cc cpp utf-8[unix] 66% ln:6/9 %:1 [7]trai...

"c++-bounds.cc" 9L, 265B

// Uninitialized locals should be zeroed: <https://wg21.link/P2723R1>

// This would mitigate or fix around 10% of security issues in security-  
// relevant code bases.

```
int main() {  
    int x;  
    char b[100];  
  
    if (predicate()) {  
        x = 1;  
        memset(b, 0, sizeof(b));  
    }  
  
    bar(b);  
  
    return x;  
}
```

~

**NORMAL**  main  **++-uninit.cc[+]** **cpp** **utf-8[unix]** **5% ln : 1/18** **3% : 1**

2 fewer lines; before #3 85 seconds ago

```
// Uninitialized locals should be zeroed:  https://wg21.link/P2723R1

// This would mitigate or fix around 10% of security issues in security-
// relevant code bases.

// Enable zeroing of locals with:
// -enable-trivial-auto-var-init-zero-knowing-it-will-be-removed-from-clang

int main() {
    int x;
    char b[100];

    if (predicate()) {
        x = 1;
        memset(b, 0, sizeof(b));
    }

    bar(b);
}
```

**NORMAL** main! **c++-uninit.cc** **cpp** utf-8[unix] 33% ln : 7/21 %: 1

"c++-uninit.cc" 21L, 409B

#

#

p:

~ ~ ~ ~ ~

1



```
# A new hope
```

```
## Erroneous behaviour for uninitialized reads
```

p2795r5 introduces a new concept in the C++ standard: Erroneous behaviour.

Default-initialization of an automatic-storage object initializes the object with a **\*\*fixed value defined by the implementation\*\***; however, **\*\*reading that value is a conceptual error\*\***. Implementations are **\*\*allowed and encouraged to diagnose this error\*\***, but they are also allowed to ignore the error and **\*\*treat the read as valid\*\***.

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

```
NORMAL  P main$ <ous.mdx mar... utf-8[unix] /fix[1/1] 69W 100% ln :11/11%:1  
"erroneous.mdx" 11L, 474B written
```

# A new hope

## Erroneous behaviour for uninitialized reads

p2795r5 introduces a new concept in the C++ standard: Erroneous behaviour.

Default-initialization of an automatic-storage object initializes the object with a **\*\*fixed value defined by the implementation\*\***; however, **\*\*reading that value is a conceptual error\*\***. Implementations are **\*\*allowed and encouraged to diagnose this error\*\***, but they are also allowed to ignore the error and **\*\*treat the read as valid\*\***.

Reading an uninitialized value is never intended and a definitive sign that the code is not written correctly and needs to be fixed. At the same time, we do give this code well-defined behaviour, and if the situation has not been diagnosed, we want the program to be stable and predictable. This is what we call *\*erroneous behaviour\**.

~

~

NORMAL P main erroneous.mdx mar... utf-8[unix] 127W 100% ln :17/17 %:1  
"erroneous.mdx" 17L, 810B



|                                                                                                             |                                                  |                                                                                                             |
|-------------------------------------------------------------------------------------------------------------|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| C++23                                                                                                       | P2723R1 (default-init zero)                      | This proposal                                                                                               |
| -----                                                                                                       | -----                                            | -----                                                                                                       |
| undefined behaviour                                                                                         | well-defined behaviour                           | erroneous behaviour                                                                                         |
| -----                                                                                                       | -----                                            | -----                                                                                                       |
| definitely a bug                                                                                            | may be intentionally using<br>0, or may be a bug | definitely a bug                                                                                            |
| -----                                                                                                       | -----                                            | -----                                                                                                       |
| common compilers<br>allow rejecting<br>(e.g. -Werror),<br>this selects a<br>non-conforming<br>compiler mode | conforming compilers<br>cannot diagnose anything | conforming compilers<br>generally have to<br>accept, but can<br>reject as QoI<br>in non-conforming<br>modes |

~  
~  
~  
~

```
public class Point {  
    private int x;  
    private int y;  
    public Point(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
    public int getX() {  
        return x;  
    }  
    public int getY() {  
        return y;  
    }  
    public void setX(int x) {  
        this.x = x;  
    }  
    public void setY(int y) {  
        this.y = y;  
    }  
}
```

NORMAL

main

<.java[+]

JavaSE-1...

jav...

[unix]

/^\$[0/4]

42%

ln :17/40

3:1

```
record Point (int x, int y) {  
    public Point {  
        if (x < 0 || y < 0) {  
            throw new IllegalArgumentException("x and y must be non-negative");  
        }  
    }  
}
```

```
}
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
class Point(val x: Int, val y: Int) {  
    operator fun plus(other: Point): Point {  
        return Point(x + other.x, y + other.y)  
    }  
}
```

```
}
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

INSERT main! point.kt[+] kot... [unix] 100% ln : 5/5 %:2

-- INSERT --

## # Conclusions

\* Your language, and your libraries should not have foot-guns.



~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

INSERT P main conclusion.mdx[+] mar... [unix] 12W 100% ln :4/4%:1

-- INSERT --

## # Conclusions

- \* Your language, and your libraries should not have foot-guns.
- \* The performance penalty of removing a foot-gun is often negligible.

22

2

2

2

2

2

2

2

2

2

2

2

2

2

2

## # Conclusions

- \* Your language, and your libraries should not have foot-guns.
- \* The performance penalty of removing a foot-gun is often negligible.
- \* In fact, the performance penalty is often negative!



~

~

~

~

~

~

~

~

~

~

~

~

~

100

- \* Your language, and your libraries should not have foot-guns.
- \* The performance penalty of removing a foot-gun is often negligible.
- \* In fact, the performance penalty is often negative!
- \* Language and library implementers should respect their users.

2 2 2 2 2 2 2 2

## # Conclusions

- \* Your language, and your libraries should not have foot-guns.
- \* The performance penalty of removing a foot-gun is often negligible.
- \* In fact, the performance penalty is often negative!
- \* Language and library implementers should respect their users.
- \* Users should respect themselves - demand more from your tools!

--

Erik Corry



~

~

~

~

~

~

~

~

INSERT P main <sion.mdx[+] mar... [unix] 55W 100% ln :11/11Ξ:1 ≡ [12]tra...

-- INSERT --



```
$ vim title.txt
$ vim introduction.mdx
$ vim dna.js
$ node --print-regexp-code dna.js > code.txt
$ vim code.txt
$ 
```

\$ ed 020-regexp.gif

668431

/NETSCAPE

Q-W 7E/O\$+F.6Kd n i y

5f(6o,g\_\_R0Ya]

Ko2Nq!leDDDIEGSKJnTixdQF0rUotprpZa^ % 9 / / 44 ' ( ( 0 ( 2 7  
Y x: + J U U J g B f ^

n K #

ThKG, ŪP ndp7n w p x Y g Al Y m Ru i v  
6y nt w x 2 y  
H U  
y

NETSCAPE2.0! x80@

\ Ō #J

3j C (S n 0c s (

s/NETSC/NOTSC/

w

668431

q

\$