The Zen of Go

GopherChina 2021

How should I write good code?

How do you know when you've written good code?

Idiomatic Go


SO I TIED AN ONION TO MY BELT, WHICH WAS THE STYLE AT THE TIME.

idiom (noun): a group of words established by usage as having a meaning not deducible from those of the individual words.


YOU CAN'T SIT WITH US
memegenerator.net

Go Proverbs

proverb (noun): a short, well-known pithy saying, stating a general truth or piece of advice.

Engineering Values

Developing a Culture

- To scale a development team, you need to establish a culture
  - Common way of evaluating designs, making tradeoffs, etc.
  - Common way of developing code and reacting to problems (build breaks, critical bugs, etc.)
  - Common way of establishing ownership of problems
- Goal setting can be the foundation for the culture
- Keeping a culture alive as a team grows is a huge challenge

Values guide your decisions in unknown situations

What are the explicit values of Go?

Perhaps we can look to values from other languages?

C++/Rust

Don’t pay for what you don’t use

Java, Ruby, Smalltalk
Everything is an object

What are a Gopher’s values?

The Zen of ~~Python~~Go

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one—and preferably only one—obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

“Namespaces are one honking great idea–let’s do more of those!”

–Tim Peters, PEP-20

Each package should have a single purpose

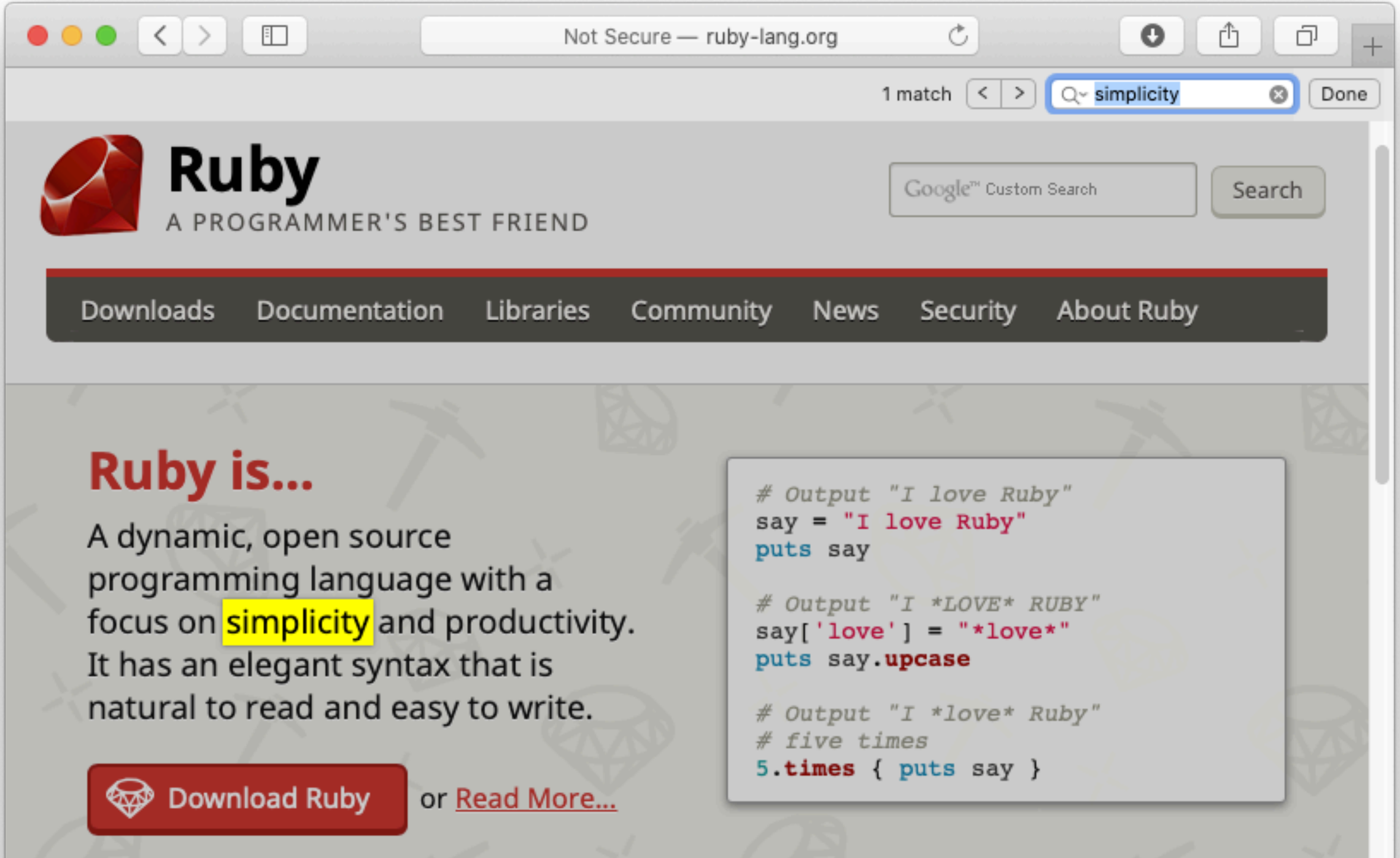
“Design is the art of arranging code to work today, and be changeable forever.”

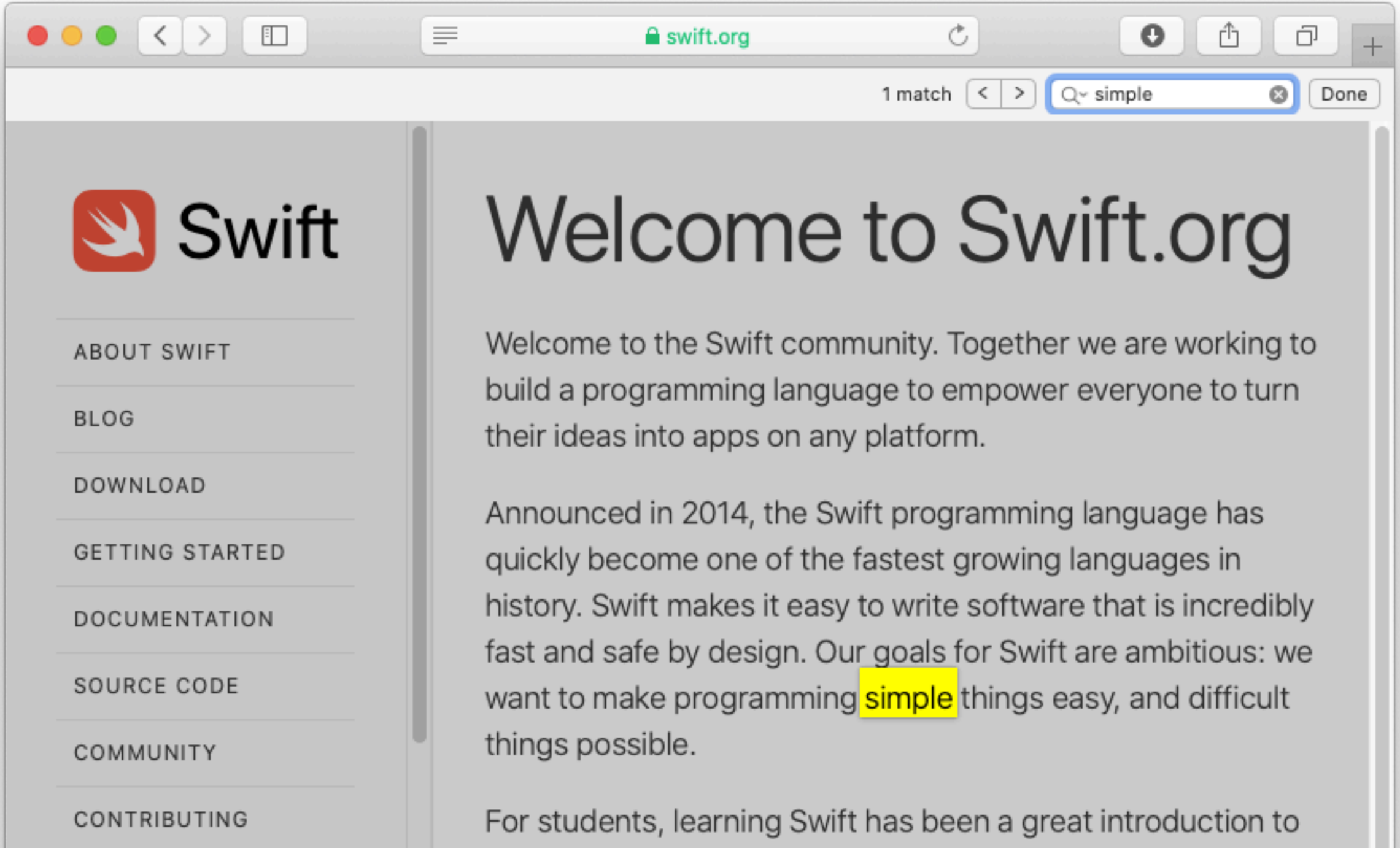
–Sandi Metz

A good package starts with a good name

Simple is better than complex


publish.twitter.com
Dave Cheney
@davecheney
Most programming languages start out aiming to be simple, but end up just settling for being powerful.
31
11:28 AM - Dec 2, 2014

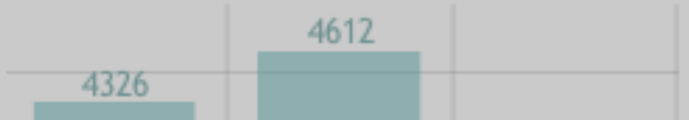

Not Secure — ruby-lang.org
1 match
simplicity
Done
Ruby
A PROGRAMMER'S BEST FRIEND
Google™ Custom Search
Search
Downloads
Documentation
Libraries
Community
News
Security
About Ruby
Ruby is...
A dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write.
Download Ruby
or Read More...
# Output "I love Ruby"
say = "I love Ruby"
puts say
Output "I *LOVE* RUBY"
say['love'] = "*love*"
puts say.upcase
Output "I *love* Ruby"
five times
5.times { puts say }


swift.org
1 match
simple
Done
Swift
ABOUT SWIFT
BLOG
DOWNLOAD
GETTING STARTED
DOCUMENTATION
SOURCE CODE
COMMUNITY
CONTRIBUTING
Welcome to Swift.org
Welcome to the Swift community. Together we are working to build a programming language to empower everyone to turn their ideas into apps on any platform.
Announced in 2014, the Swift programming language has quickly become one of the fastest growing languages in history. Swift makes it easy to write software that is incredibly fast and safe by design. Our goals for Swift are ambitious: we want to make programming simple things easy, and difficult things possible.
For students, learning Swift has been a great introduction to

```
This `toInt` call produces:

    Maybe Int

But `drop` needs the 1st argument to be:

    Int

Hint: Use Maybe.withDefault to handle possible errors.
```

Great Performance

Elm has its own virtual DOM implementation, designed for simplicity and speed. All values are immutable in Elm, and the benchmarks show that this helps us generate particularly fast JavaScript code. (details)



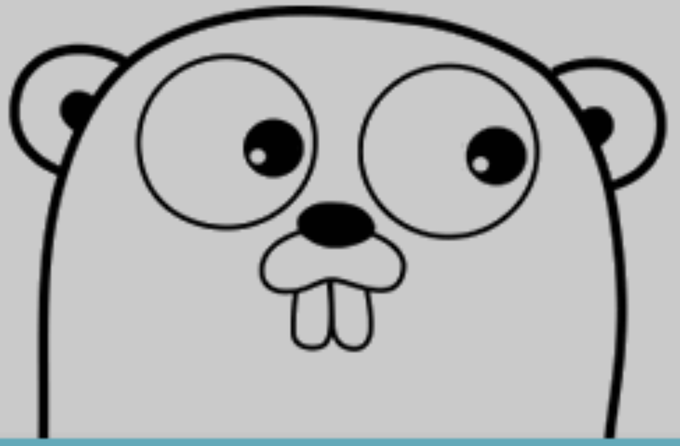
Benchmark Times

In milliseconds on Chrome 52

golang.org

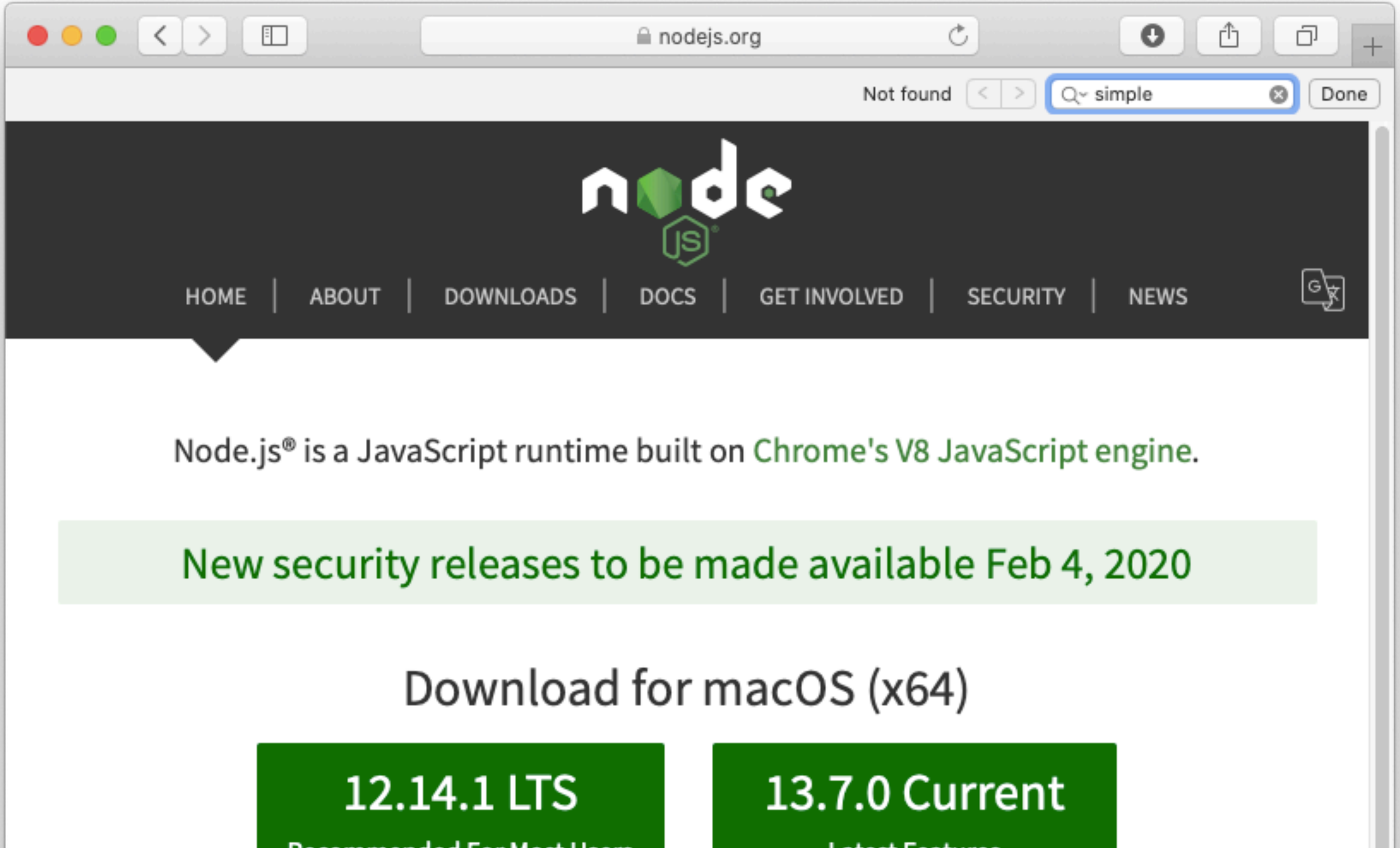
1 match simple Done

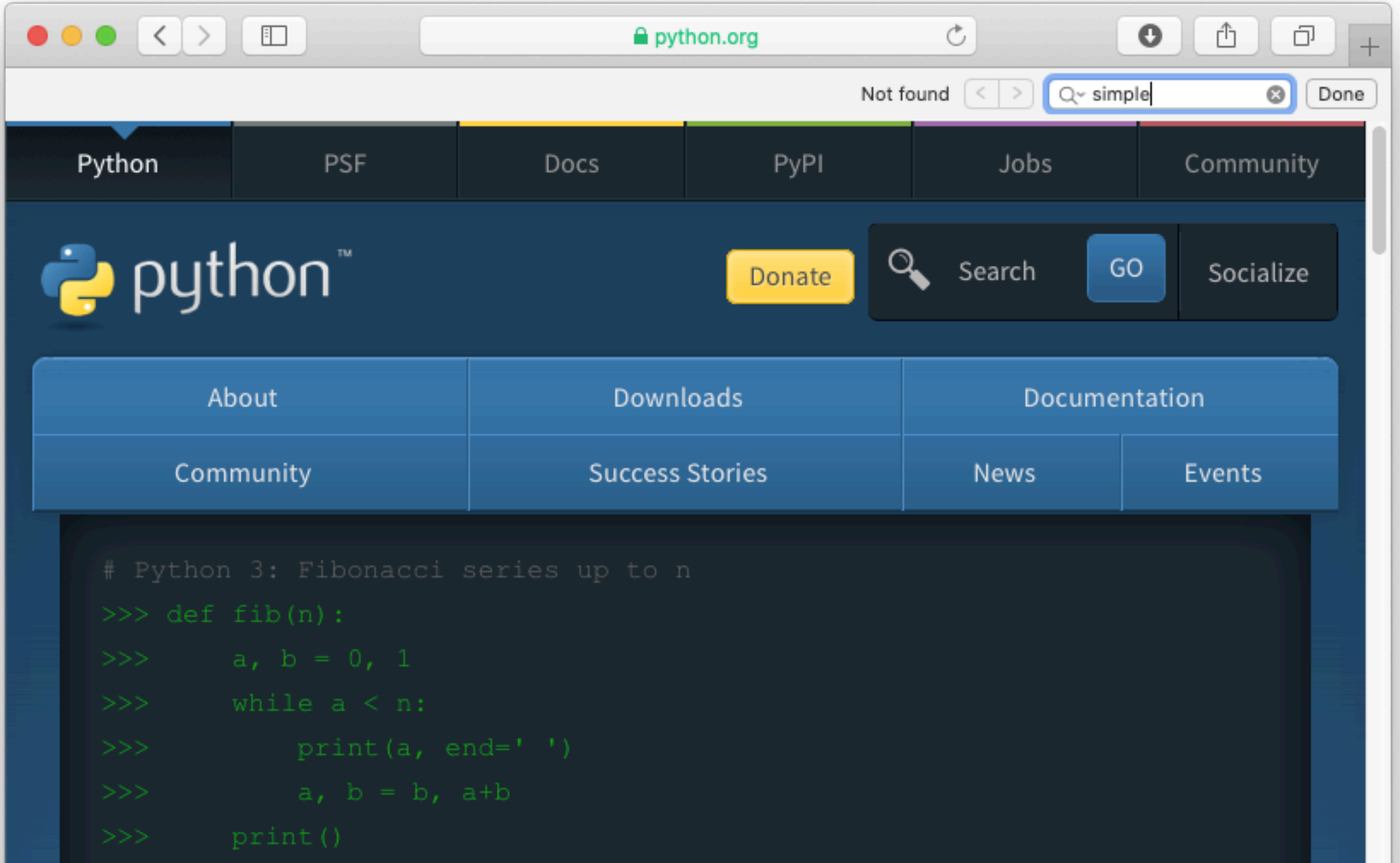
Go is an open source programming language that makes it easy to build **simple**, **reliable**, and **efficient** software.

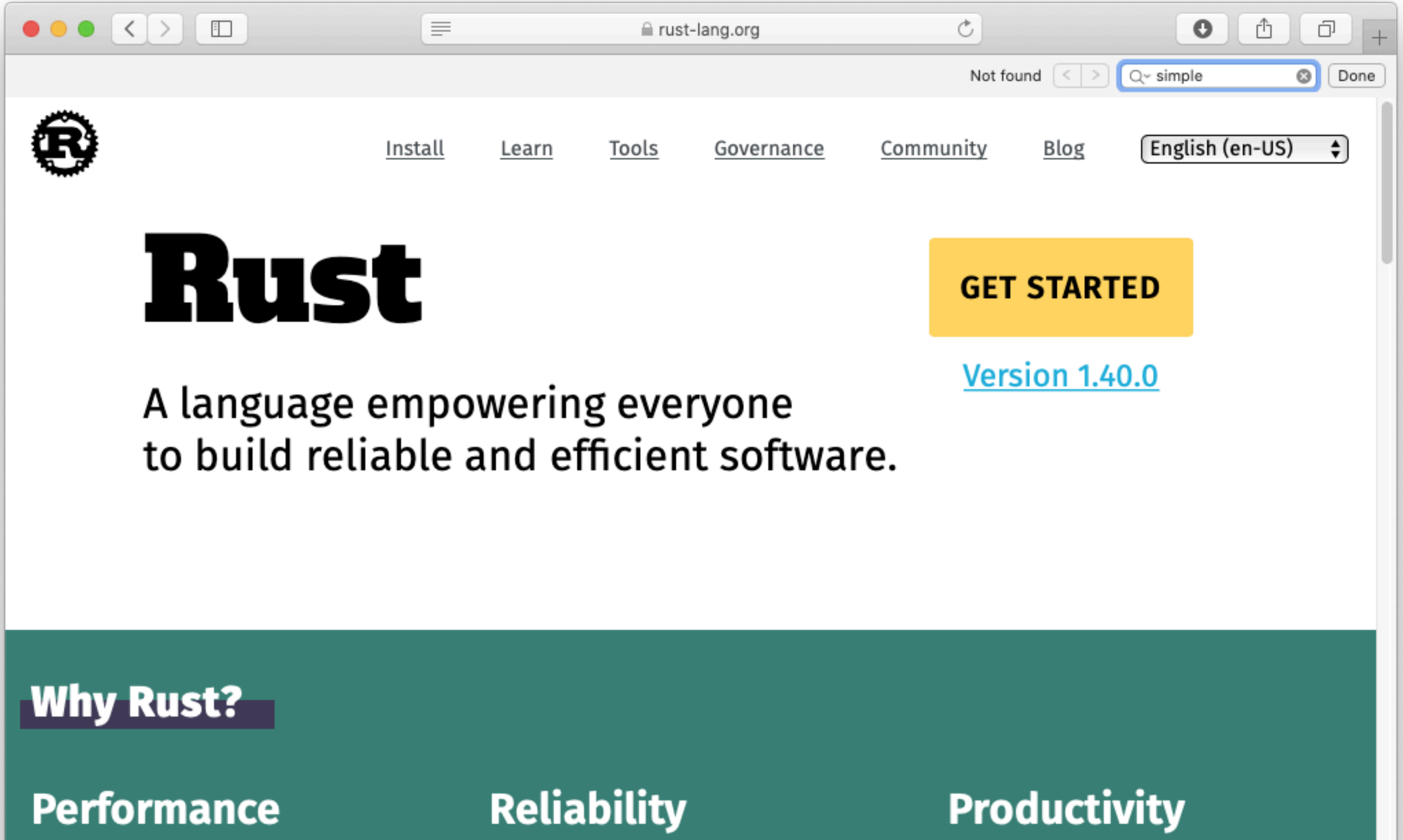


Download Go

Binary distributions available for
Linux, macOS, Windows, and more.


nodejs.org
Not found
simple
Done
HOME
ABOUT
DOWNLOADS
DOCS
GET INVOLVED
SECURITY
NEWS
Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.
New security releases to be made available Feb 4, 2020
Download for macOS (x64)
12.14.1 LTS
13.7.0 Current


python.org
Not found
simple
Done
Python
PSF
Docs
PyPI
Jobs
Community
python™
Donate
Search
GO
Socialize
About
Downloads
Documentation
Community
Success Stories
News
Events
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>   a, b = 0, 1
>>>   while a < n:
>>>   print(a, end=' ')
>>>   a, b = b, a+b
>>>   print()


rust-lang.org
Not found
simple
Done
Install
Learn
Tools
Governance
Community
Blog
English (en-US)
Rust
GET STARTED
Version 1.40.0
A language empowering everyone
to build reliable and efficient software.
Why Rust?
Performance
Reliability
Productivity

Simplicity matters

“There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult.”

–C.A.R. Hoare

“Simplicity is prerequisite for reliability.”

–Edsger W. Dijkstra

“Controlling complexity is the essence of computer programming.”

–Brian Kernighan

Explicit is better than implicit

Seek to be explicit

State

The problem of computer science

```
package counter

var count int

func Increment(n int) int {
        count+=n
        return count
}
```

```
package counter

type Counter struct {
        count int
}

func (c *Counter) Increment(n int) int {
        c.count+=n
        return c.count
}
```

Avoid package level state

Errors should never pass silently

The Samurai Principle

Return victorious or not at all

Plan for failure, not success

```
if err != nil {
        return err
}
```

“I think that error handling should be explicit, this should be a core value of the language.”

–Peter Bourgon, GoTime #91

Handle errors explicitly

Flat is better than nested

Return early rather than nesting deeply

“Line of sight is a straight line along which an observer has unobstructed vision.”

–Mat Ryer
Align the happy path to the left edge

Return early rather than nesting deeply

In the face of ambiguity, refuse the temptation to guess

What does this variable do?

What does this parameter do?

What happens if I pass nil here?

What happens if I call Register twice?

“APIs should be easy to use and hard to misuse.”

–Josh Bloch

If you think it’s slow, prove it with a benchmark

Going beyond

“You type g o, a space, and then a function call. Three keystrokes, you can’t make it much shorter than that. Three keystrokes and you’ve just started a sub process.”

–Rob Pike, Simplicity is Complicated, dotGo 2015

Goroutines are the key to resource ownership in Go

Under what condition will a goroutine stop?

What is required for that condition to arise?

What signal will you use to know the goroutine has stopped

Before you launch a goroutine, know when it will stop

Leave concurrency to the caller

Write tests to lock in the behaviour of your package's API

Moderation is a virtue

Readability counts

Maintainability counts

the-zen-of-go.netlify.com

Thank you!