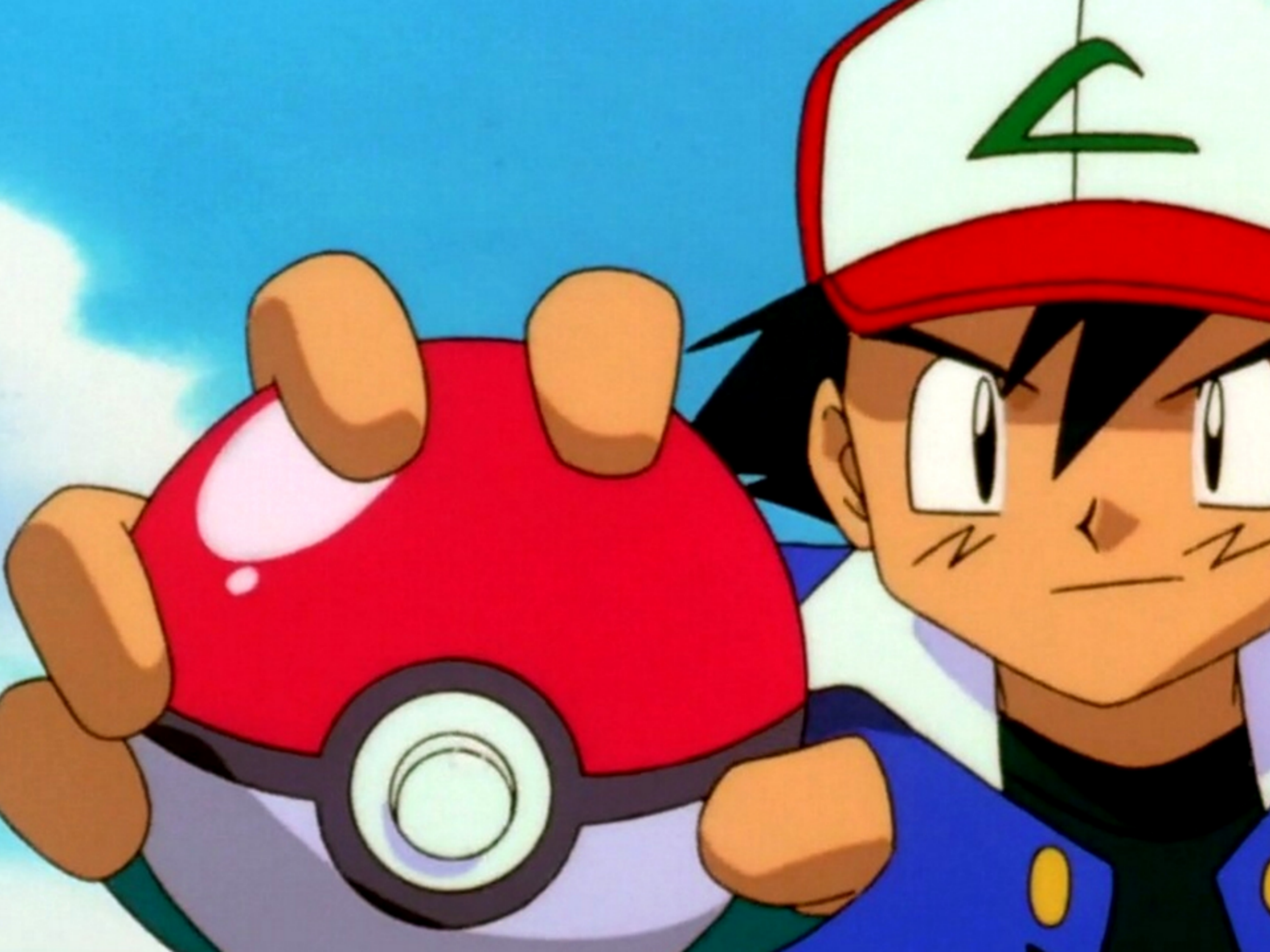


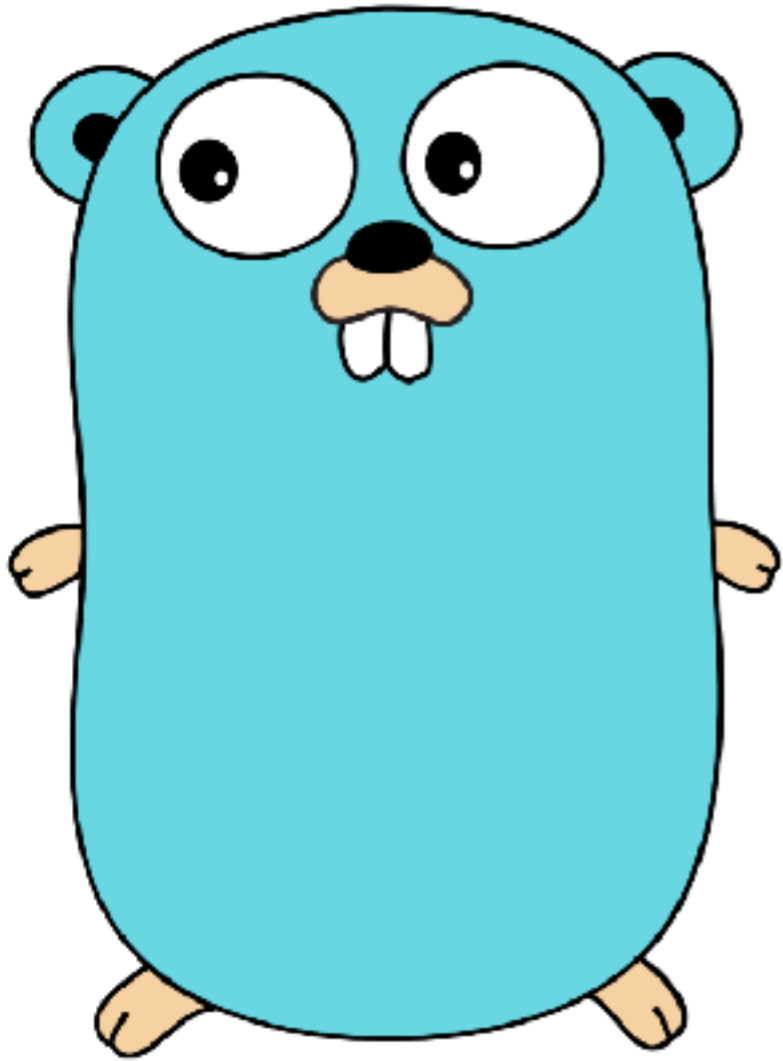
<http://bit.ly/ergo-rr2018>

ergo

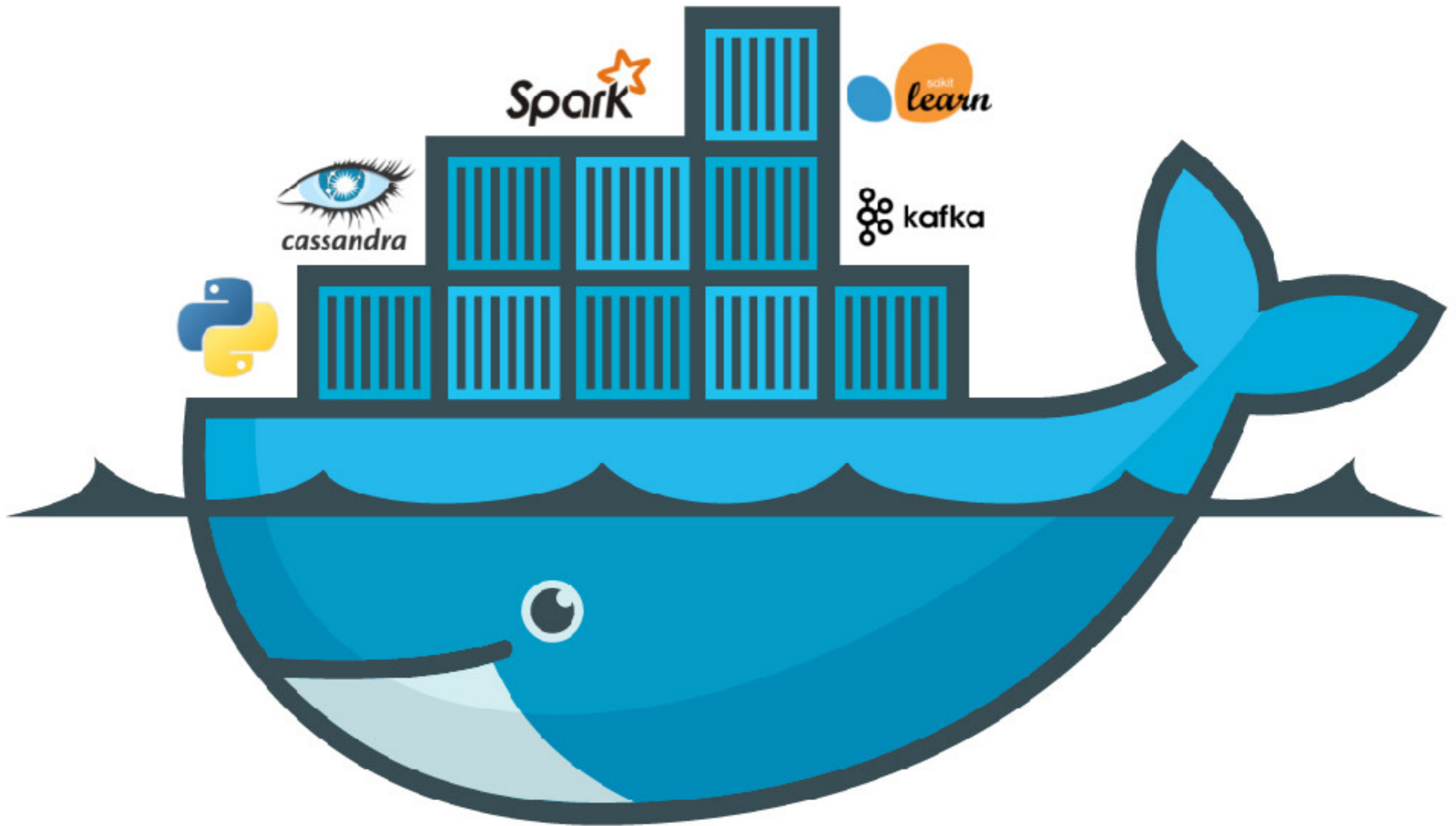
@romain_francois
romain@rstudio.com







- Open source
- Google
- Simple
- Fast enough





<https://bookdown.org/yihui/blogdown/>

RStudio Connect

Content / Access to Care Dashboard

Studio Access to Hospital Care Dashboard State Info Model About in Source Code

Options
Select a State:
California

39M Population
345 Hospitals
58 Counties
3 Underserved Counties

County Map County summary

The map displays California with various counties highlighted in different colors based on hospital density. A legend in the bottom right corner explains the color coding: red for 'Less hospitals than expected', blue for 'More hospitals than expected', green for 'Within Range', and yellow dots for 'Hospital Location'. Major cities like Sacramento, San Francisco, Los Angeles, and San Diego are labeled. The map also shows neighboring states like Nevada, Utah, and Arizona.

Legend

- Less hospitals than expected
- More hospitals than expected
- Within Range
- Hospital Location

Who can view this document
Anyone - no login required

Who can change this document

- ER Edgar Ruiz edgar
- SL Sean Lopp sean
- RD Rachael Dempsey rachael
- AD Andrie de Vries andrie

Add collaborator

Who runs this document on the server
The default user

Administrators can create a custom 'vanity' url to access this document. Your custom url will be appended to your domain to form the complete path to your document.

No custom url defined

<https://www.rstudio.com/products/connect/>

ex**am**p**le**


```
package main
```

```
import "fmt"
```

```
func fahrenheit(celcius float64) float64 {  
    return celcius * 1.8 + 32  
}
```

```
func main() {  
    var freezing float64 = fahrenheit(0.0)  
    boiling := fahrenheit(100.0)  
  
    fmt.Printf("Water freezes at %4.2f F\n", freezing)  
    fmt.Printf("Water boils at %4.2f F\n", boiling)  
}
```

```
package main
```

```
import "fmt"
```

```
func fahrenheit(celcius float64) float64 {  
    return celcius * 1.8 + 32  
}
```

```
func main() {  
    var freezing float64 = fahrenheit(0.0)  
    boiling := fahrenheit(100.0)  
  
    fmt.Printf("Water freezes at %4.2f F\n", freezing)  
    fmt.Printf("Water boils at %4.2f F\n", boiling)  
}
```

```
$ go run fahrenheit.go
```

```
Water freezes at 32.00 F
```

```
Water boils at 212.00 F
```

research



2018

gofast

2018/03/23

2017

Go Slices

2017/07/03

Using Go strings in R

2017/06/10

Go packages in R packages

2017/06/09

Calling go from R

2017/05/14

purrple.cat/tags/go/
bit.ly/ergorigin



2018

gofast

2018/03/23

2017

Go Slices

2017/07/03

Using Go strings in R

2017/06/10

Go packages in R packages

2017/06/09

Calling go from R

2017/05/14

purrple.cat/tags/go/
bit.ly/ergorigin

Console

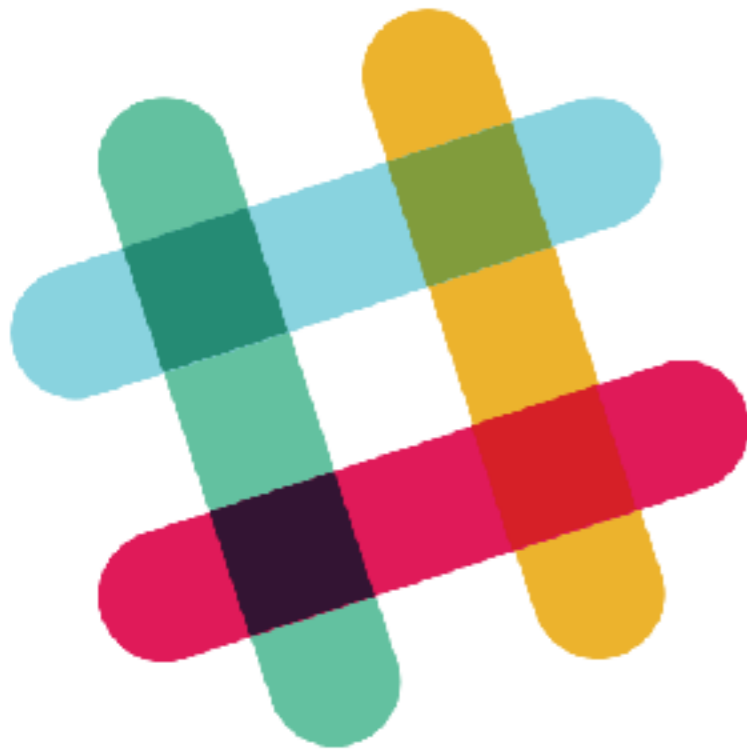
Terminal ×

~/git/rstats-go/rencontres2018/fahrenheit/ ↩


```
> # install_github("rstats-go/rencontres2018/fahrenheit")
> library(fahrenheit)
> fahrenheit(100)
[1] 212
> fahrenheit(37)
[1] 98.6
> fahrenheit(0)
[1] 32
>
```



<http://bit.ly/grrrr-join>






Thread ✕
Robin, Julien, and 2 others

 **romainfrancois** 🏆 Jul 1st at 11:08 AM
in #questions

qqun a une api pour recuperer la temperature actuelle d'une ville ?

4 replies

 **Robin** 4 days ago
Pas vraiment une API, mais ça c'est top :
<https://github.com/chubin/wttr.in>

 GitHub
[chubin/wttr.in](https://github.com/chubin/wttr.in)
wttr.in - 🌤️ The right way to check the weather 

romain@purrplex ~ \$ curl wttr.in/Rennes

Weather report: Rennes, France

wttr.in/Rennes

```

  \ /      Sunny
  -.-      19 °C
 - ( ) -   → 0 km/h
  ^-^      10 km
  / \      0.0 mm

```

Thu 05 Jul

Morning

```

  \ /      Partly cloudy
 - /"".-.  19 °C
  \ ( ) .  → 7-9 km/h
 / (___)   16 km
           0.0 mm | 0%

```

Noon

```

  \ /      Partly cloudy
 - /"".-.  23-24 °C
  \ ( ) .  ↑ 6-7 km/h
 / (___)   15 km
           0.0 mm | 0%

```

Evening

```

 - /"".-.  Light rain sho...
 , \ ( ) .  23-25 °C
 / (___)   ↘ 9-12 km/h
 ' ' ' '   16 km
 ' ' ' '   0.3 mm | 31%

```

Night

```

  \ /      Partly cloudy
 - /"".-.  20 °C
  \ ( ) .  ↓ 11-21 km/h
 / (___)   17 km
           0.3 mm | 61%

```

Fri 06 Jul

Morning

```

  \ /      Partly cloudy
 - /"".-.  20 °C
  \ ( ) .  ↑ 7-9 km/h
 / (___)   17 km
           0.0 mm | 0%

```

Noon

```

  \ /      Partly cloudy
 - /"".-.  24-25 °C
  \ ( ) .  ↑ 8-9 km/h
 / (___)   17 km
           0.0 mm | 0%

```

Evening

```

  \ /      Partly cloudy
 - /"".-.  24-26 °C
  \ ( ) .  ↑ 13-19 km/h
 / (___)   15 km
           0.0 mm | 0%

```

Night

```

  \ /      Partly cloudy
 - /"".-.  22 °C
  \ ( ) .  ↗ 14-26 km/h
 / (___)   16 km
           0.0 mm | 0%

```

Sat 07 Jul

Morning

```

  \ /      Partly cloudy
 - /"".-.  22-23 °C
  \ ( ) .  ↗ 8-10 km/h
 / (___)   18 km
           0.0 mm | 0%

```

Noon

```

  \ /      Partly cloudy
 - /"".-.  26-27 °C
  \ ( ) .  ← 9-10 km/h
 / (___)   16 km
           0.0 mm | 0%

```

Evening

```

 - /"".-.  Light rain sho...
 , \ ( ) .  25-27 °C
 / (___)   ↗ 12-16 km/h
 ' ' ' '   13 km
 ' ' ' '   1.0 mm | 66%

```

Night

```

 - /"".-.  Patchy rain po...
 , \ ( ) .  21 °C
 / (___)   ↗ 11-21 km/h
 ' ' ' '   15 km
 ' ' ' '   0.5 mm | 70%

```

```

library(rvest)
library(glue)
library(fahrenheit)
library(tibble)
library(dplyr)

temperature <- function(when = "Rennes"){
  glue("https://wttr.in/{when}") %>%
    read_html() %>%
    html_node("span:nth-child(3)") %>%
    html_text() %>%
    as.numeric()
}

```

```

temperature("Rennes") %>%
  tibble(celcius = .) %>%
  mutate(fahrenheit = fahrenheit(celcius))

```

```

#> # A tibble: 1 x 2
#>   celcius fahrenheit
#>   <dbl>     <dbl>
#> 1      19         66.2

```

```


#' Created on 2018-07-05 by the [reprex package](http://reprex.tidyverse.org)
(v0.2.0).


```



```
romain@purrplex ~/git/rstats-go/rencontresr2018/fahrenheit $ tree
```

```
.
├── DESCRIPTION
├── NAMESPACE
├── R
│   └── fahrenheit.R
├── fahrenheit.Rproj
├── man
└── src
    ├── Makevars
    ├── fahrenheit.h
    ├── fahrenheit.so
    └── go
        └── src
            ├── fahrenheit
            │   └── fahrenheit.go
            └── main
                ├── main.c
                └── main.go
```

}  **fahrenheit**

}  **main**

```
7 directories, 10 files
```

```
package fahrenheit
```

```
func Fahrenheit(celcius float64) float64 {  
    return celcius * 1.8 + 32  
}
```

```
package main
```

```
import "C"
```

```
import "fahrenheit"
```

```
//export Fahrenheit
```

```
func Fahrenheit(x float64) float64 {  
    return fahrenheit.Fahrenheit(x) ;  
}
```

```
func main() {}
```

```
#include <R.h>
#include <Rinternals.h>
#include "_cgo_export.h"
```

```
SEXP _fahrenheit(SEXP x) {
    return Rf_ScalarReal(Fahrenheit(REAL(x)[0]));
}
```

```
# ' @useDynLib fahrenheit
# ' @export
fahrenheit <- function(celcius) {
  .Call("_fahrenheit",
        celcius,
        PACKAGE = "fahrenheit"
  )
}
```

```
.PHONY: go
```

```
CGO_CFLAGS = "$(ALL_CPPFLAGS)"
```

```
CGO_LDFLAGS = "$(PKG_LIBS) $(SHLIB_LIBADD) $(LIBR)"
```

```
GOPATH = $(CURDIR)/go
```

```
go:
```

```
CGO_CFLAGS=$(CGO_CFLAGS) CGO_LDFLAGS=$(CGO_LDFLAGS) \
```

```
  GOPATH=$(GOPATH) /usr/local/go/bin/go \
```

```
  build -o $(SHLIB) -x -buildmode=c-shared main
```


future





romainfrancois last(?) changes 4e6f563 on 31 Mar

6 contributors

197 lines (142 sloc) | 8.81 KB

Raw Blame History

title	author	date	output
ergo: high level interface between R and Go	Romain François	2018/03/31	pdf_document

```
knitr::opts_chunk$set(echo = TRUE)
```

The problem

R is an amazing interpreted language, giving a flexible and agile foundation for Data Science. Efforts such as Rcpp and reticulate have established that it can be an advantage to pair R with another programming language. Sometimes for speed, sometimes to have alternative options of expression, sometimes to have access to existing libraries.

Go (<https://golang.org>) is an open source programming language that makes it easy to build simple, reliable and efficient software. It is sometimes said to be the language C++ should have been, in particular if it did not carry a strong commitment to backwards compatibility to C and a taste for complexity.

Go is beautiful and simple, its standard library is one of the most impressive for a programming language. It comes with concurrency built in, which includes (but is not limited to) running code in parallel. The static site generator [huugo](#), the

Hadley Wickham

ISC proposal: ergo: high level interface between R and Go...

À : Romain Francois, Cc : John Mertic

Dear Romain François,

Thank you for sharing your proposal with us. Unfortunately the ISC did not elect to fund your proposal as many proposals were submitted to the ISC and only a limited number could be funded. Although the committee agrees that a Go interface for R would be useful, we don't feel that this is of broad enough need to be considered an infrastructure project for the R community. The consortium does not generally fund individual package development work

Regards,

Hadley Wickham
Chair, Infrastructure Steering Committee
R Consortium



<http://bit.ly/ergo-rr2018>

ergo

@romain_francois
romain@rstudio.com