

Beautiful IO

A tour through standard library `pkg/io` and various implementations of its interfaces.

Golab 2019, 2019–10–21, Firenze

Martin Czygan

About me

SWE [@ubleipzig](#) working mostly with Python and Go.

Taming data – open source – writing.

| [Explore IO](#) workshop at Golab 2017.

The IO package

- contains basic, widely used interfaces (within and outside standard library)
- simple, flexible helpers

Why beautiful?

La bellezza è negli occhi di chi guarda

- small, versatile interfaces
- composable

Love and praise

This article aims to convince you to use `io.Reader` in your own code wherever you can. -- [@matryer](#)

"Crossing Streams: a love letter to Go `io.Reader`" -- [@jmoiron](#)

Which brings me to `io.Reader`, easily my favourite Go interface. --
[@davecheney](#)

What's in pkg/io?

- 25 types - of which 21 are interfaces, 12 functions, 3 constants, 6 errors
- Four concrete types are: `LimitedReader`, `PipeReader`, `PipeWriter`, `SectionReader`.
- Functions: `Copy`, `CopyN`, `CopyBuffer`, `Pipe`, `ReadAtLeast`, `ReadFull`, `WriteString`, `LimitReader`, `MultiReader`, `TeeReader`, `NewSectionReader`, `MultiWriter`

How many readers, writers are in the standard library?

- TODO: count

How do you implement them?

```
type Reader interface {  
    Read(p []byte) (n int, err error)  
}
```

```
type Writer interface {  
    Write(p []byte) (n int, err error)  
}
```


