

# Introduction to Shakti

## AquaQuarantine Tech Talk Series



# Happy 10th Birthday AquaQ!



Gifson.net

SHAKTI

# Agenda

- Why?
- What?
- How?
- Who?
- Where?
- When?

Why?

# Setting the Scene



# Today's Enterprise Software Stack

## Multiple Layers

- Analytics
- Connectivity
- Database
- Presentation
- Security
- Streaming/Messaging
- Web

# Hardware Overhead

- On-Premises
- Public Cloud
- Private Cloud
- Multi Cloud
- Hybrid Cloud
- Co-Location
- DR/Backup

# Software Overhead

- Multiple Programming Languages
- Multiple APIs
- Multiple Versions
- Multiple Dependencies

# Operational Overhead

- More Developers
- More Code
- More Points of Failure
- Longer Procurement Cycles
- Vendor Lock-In

# Performance Overhead

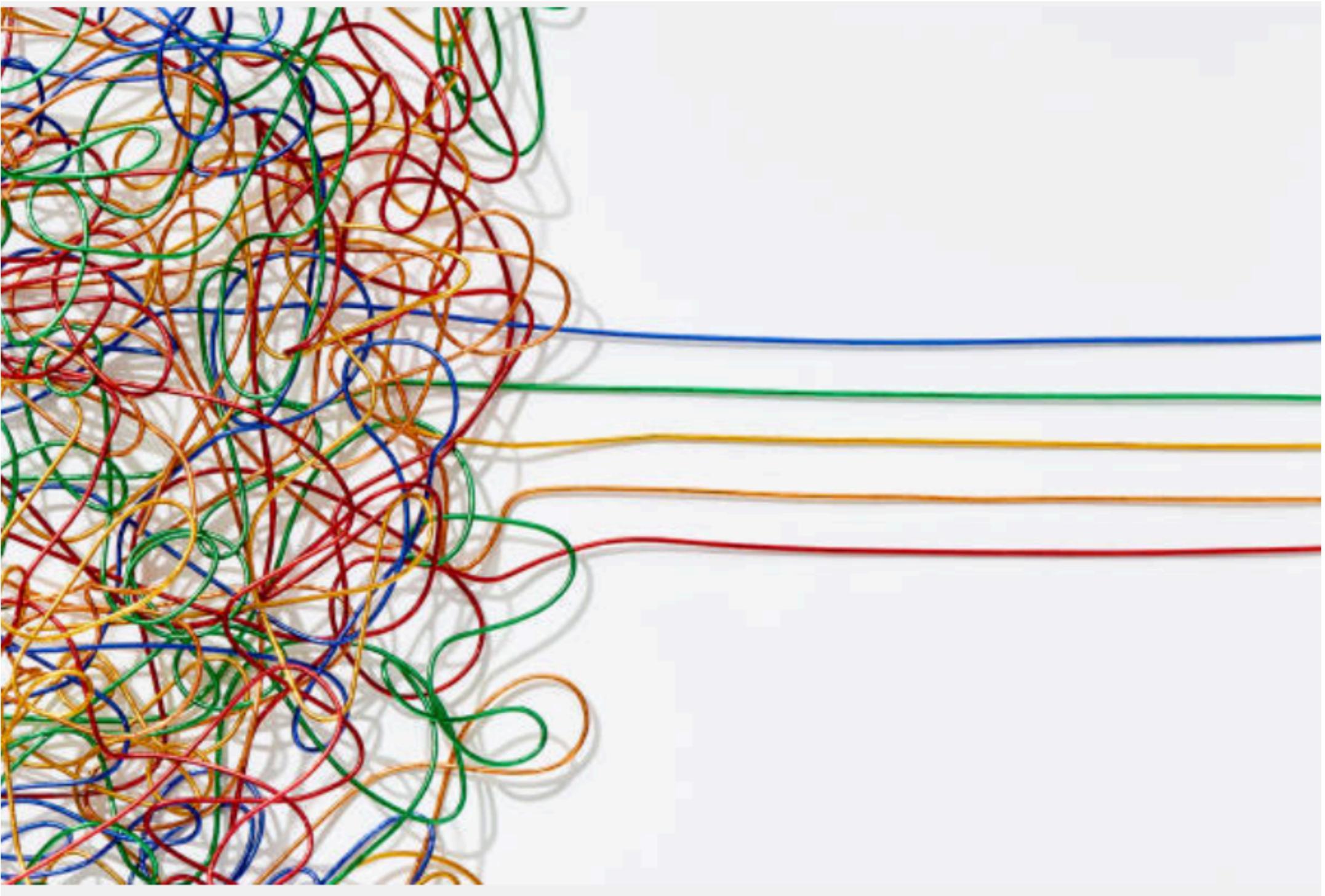
- More Network Hops
- Serialization/Deserialization Overhead
- Increased Latency
- Increased Storage

# Underperforming Infrastructure

- More Software
- More Hardware
- Higher Operational Cost
- **Lower Performance**
- **Increased TCO**

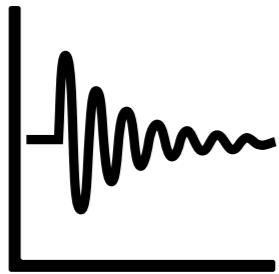


# Solving the Complexity

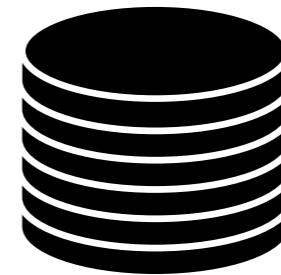


What?

# The Shakti Solution



**Analytics**



**Database**



**Connectivity**



**Streaming**

# Analytics

- Language: **k9**
- Type: Vector
- Query: Simplified & Extended SQL
- Functional: System & User-defined

# Database

- CRUD (insert/select/update/delete)
- Joins (simplified)
- Time-Series (data types & arithmetic)
- Structured
- Semi-Structured/Document

# Connectivity

- Language Support
- Data Formats
- Wire Formats
- CPU Architectures
- Operating Systems

# Streaming

- 10s of millions of inserts/updates per second per core
- Sync
- Async
- Fork-Exec

How?

ALMOST DONE WITH  
THIS MONSTER  
PROJECT!

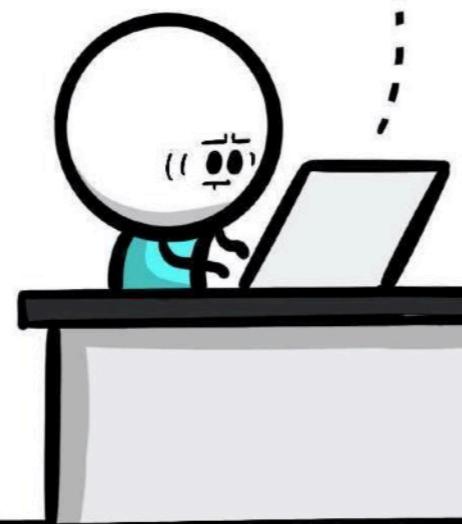


\*SOFTWARE IS NOT  
RESPONDING\*

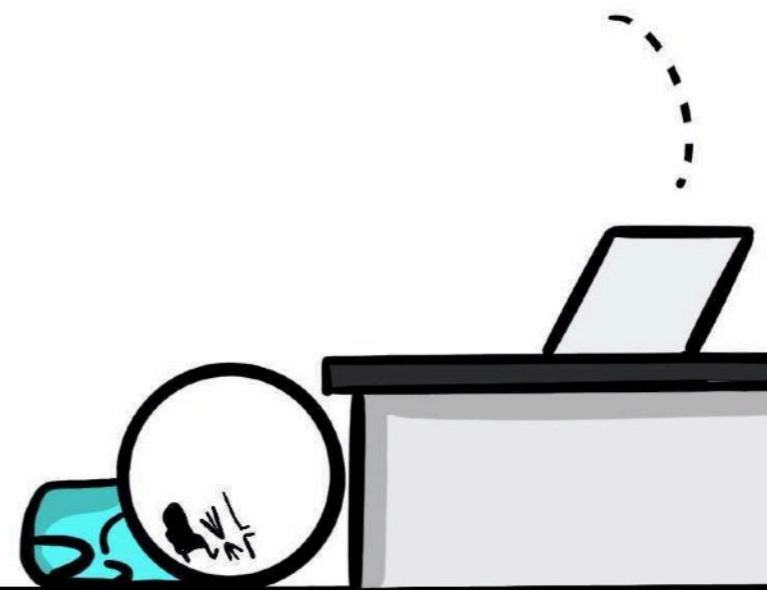


workchronicles.com

\*SOFTWARE  
CRASHED  
UNEXPECTEDLY.\*



\*RESTORING VERSION  
SAVED 3HRS BACK.\*



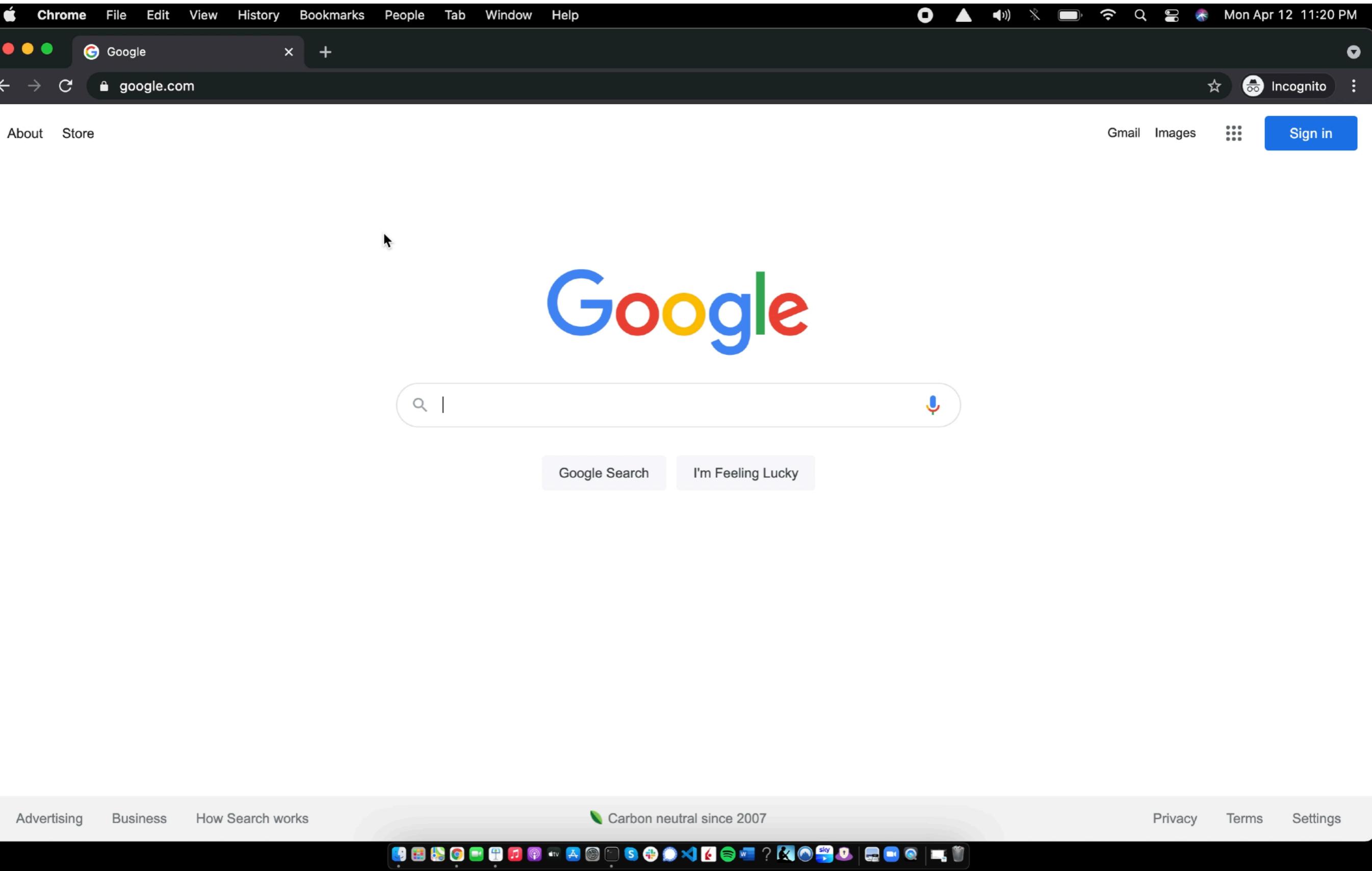
Hello. I make comics about work.

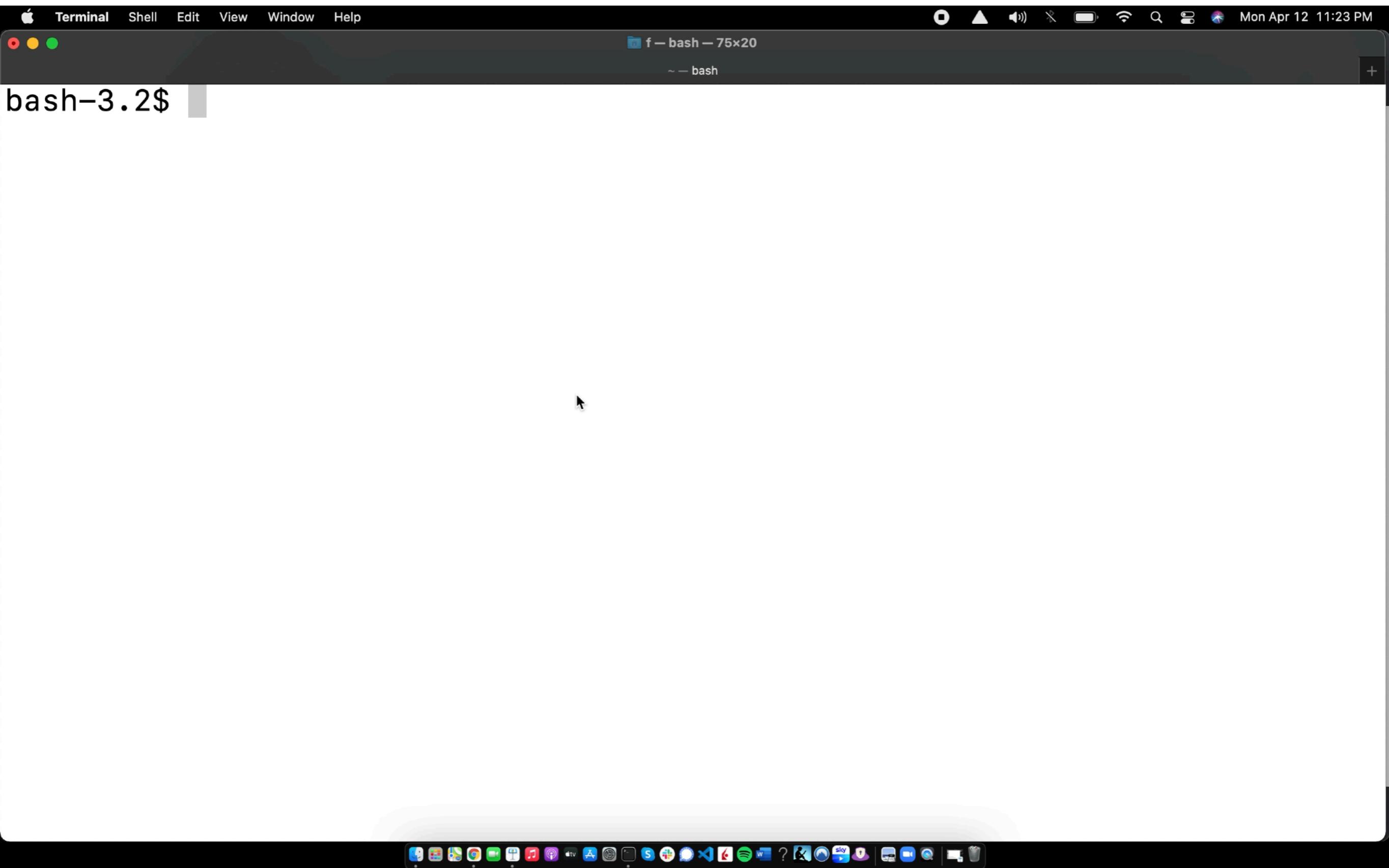
Join r/workchronicles or follow on Instagram/Twitter/FB

Work Chronicles

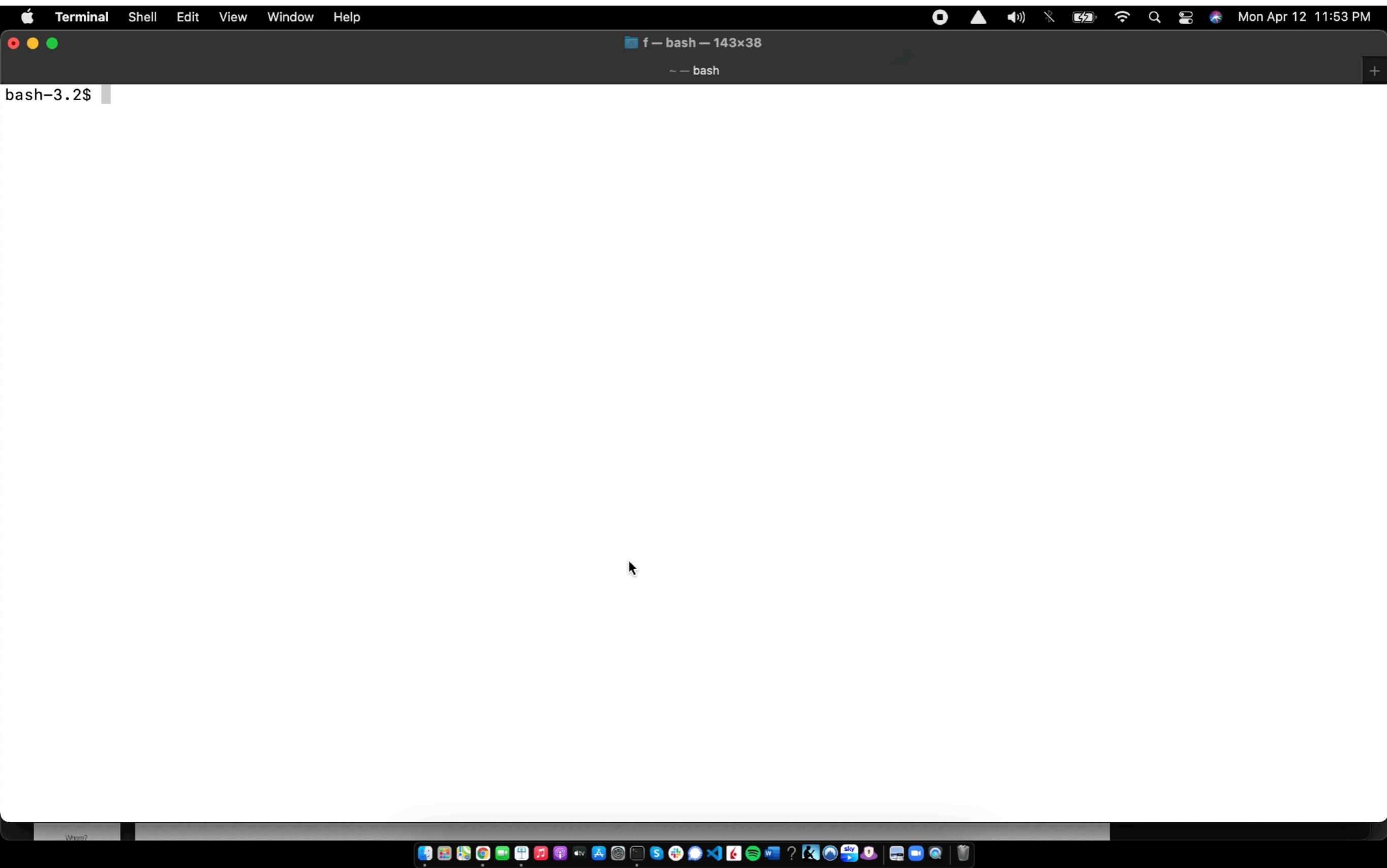
workchronicles.com

# Download

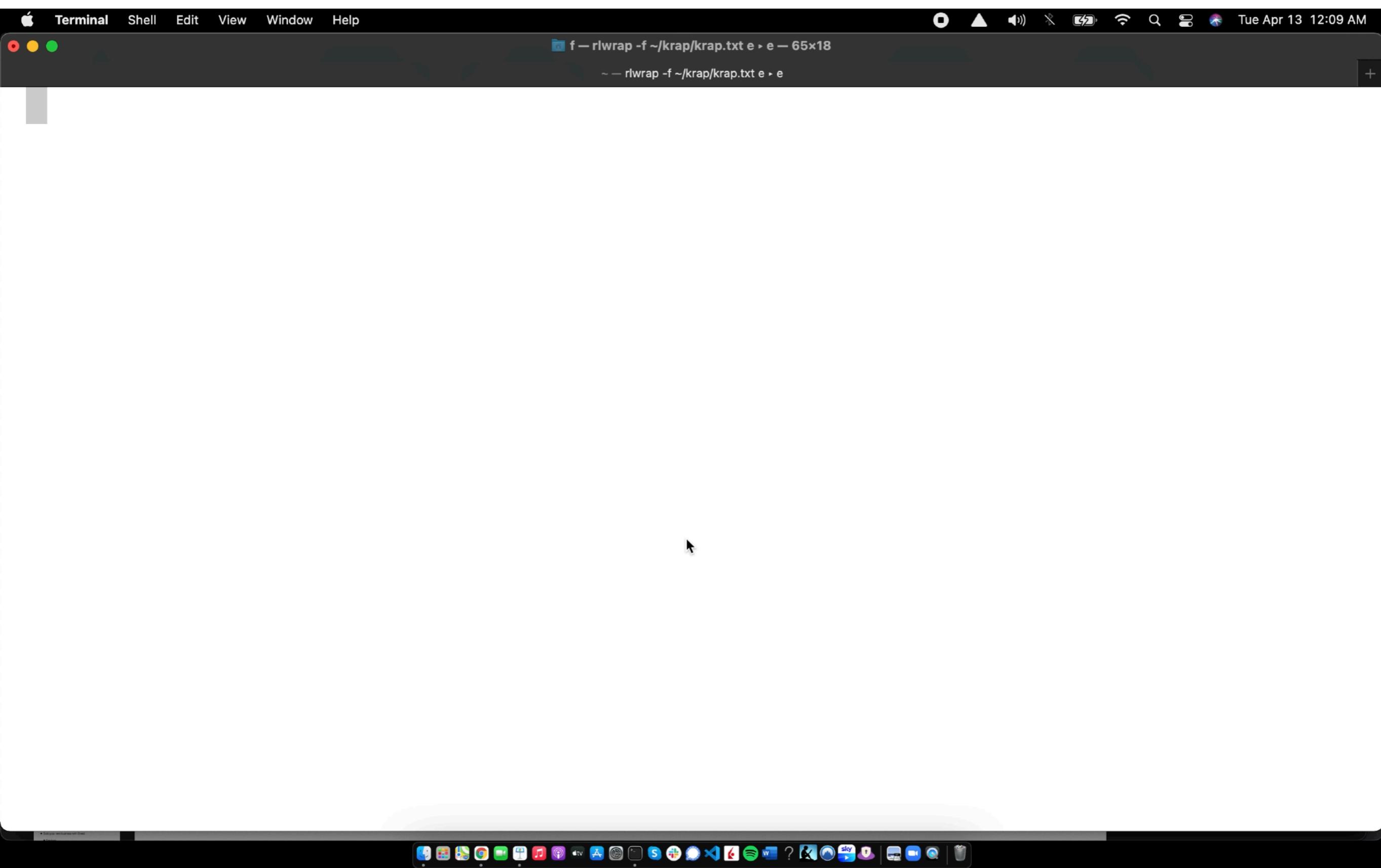




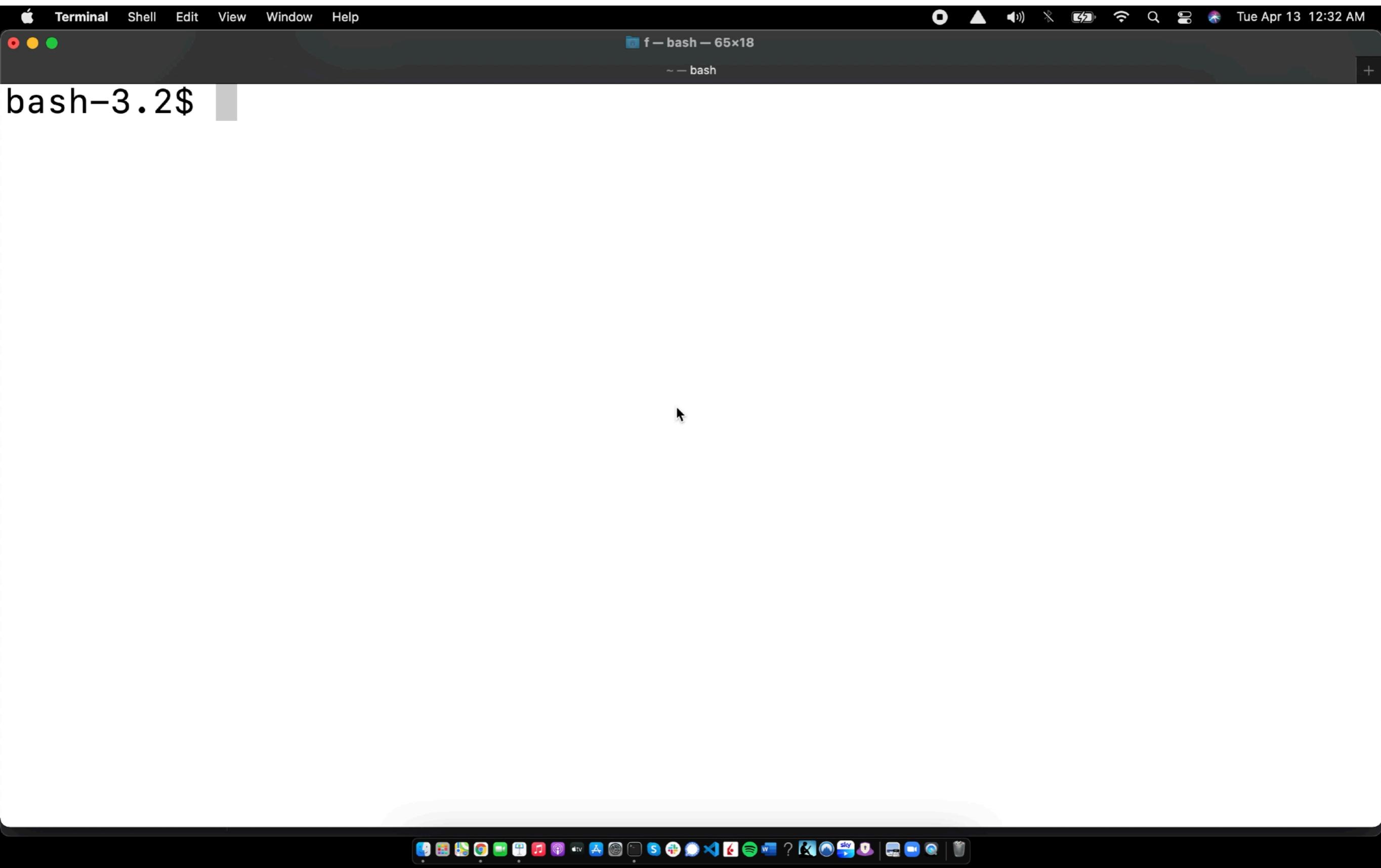
Shell



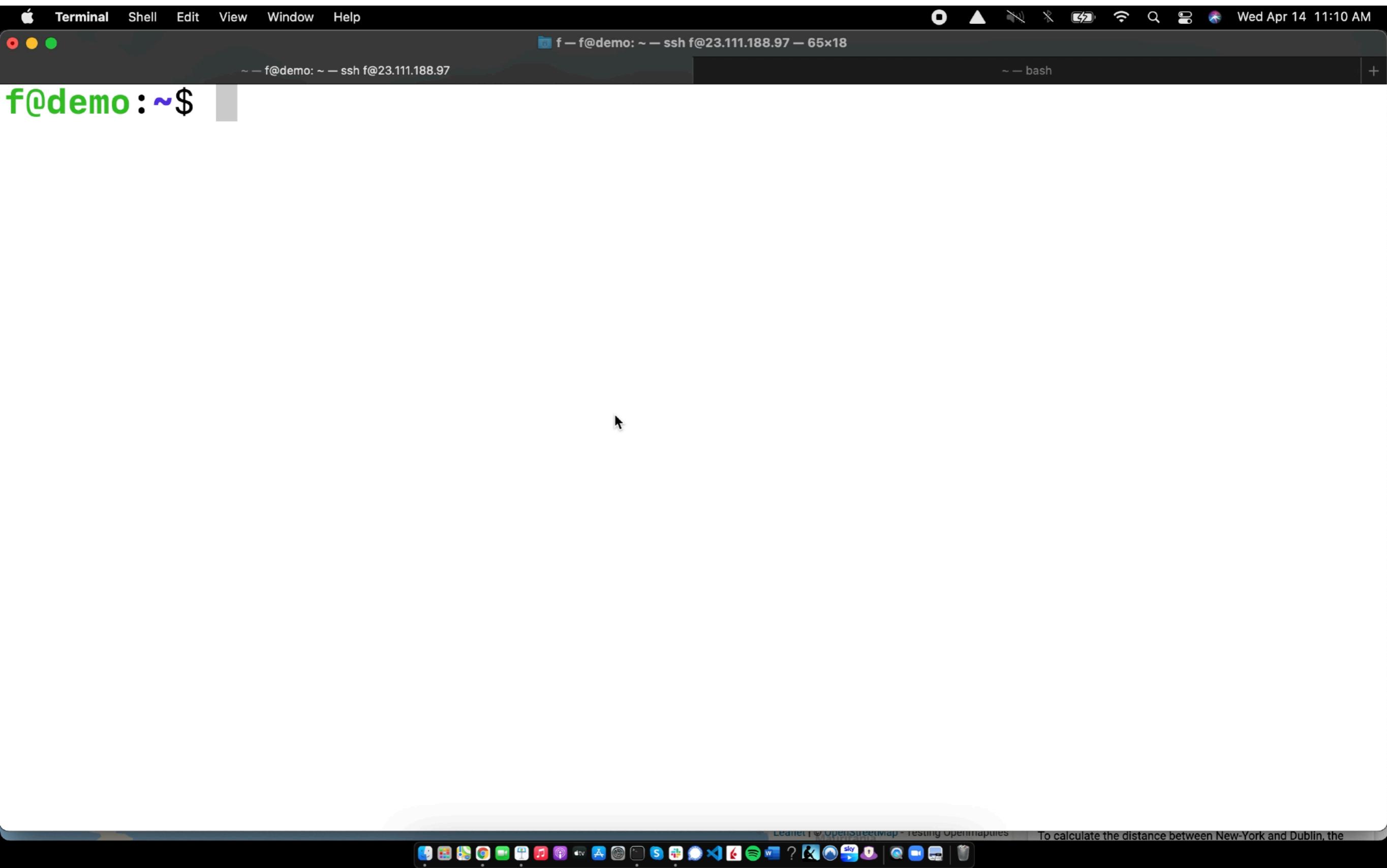
# Data Types



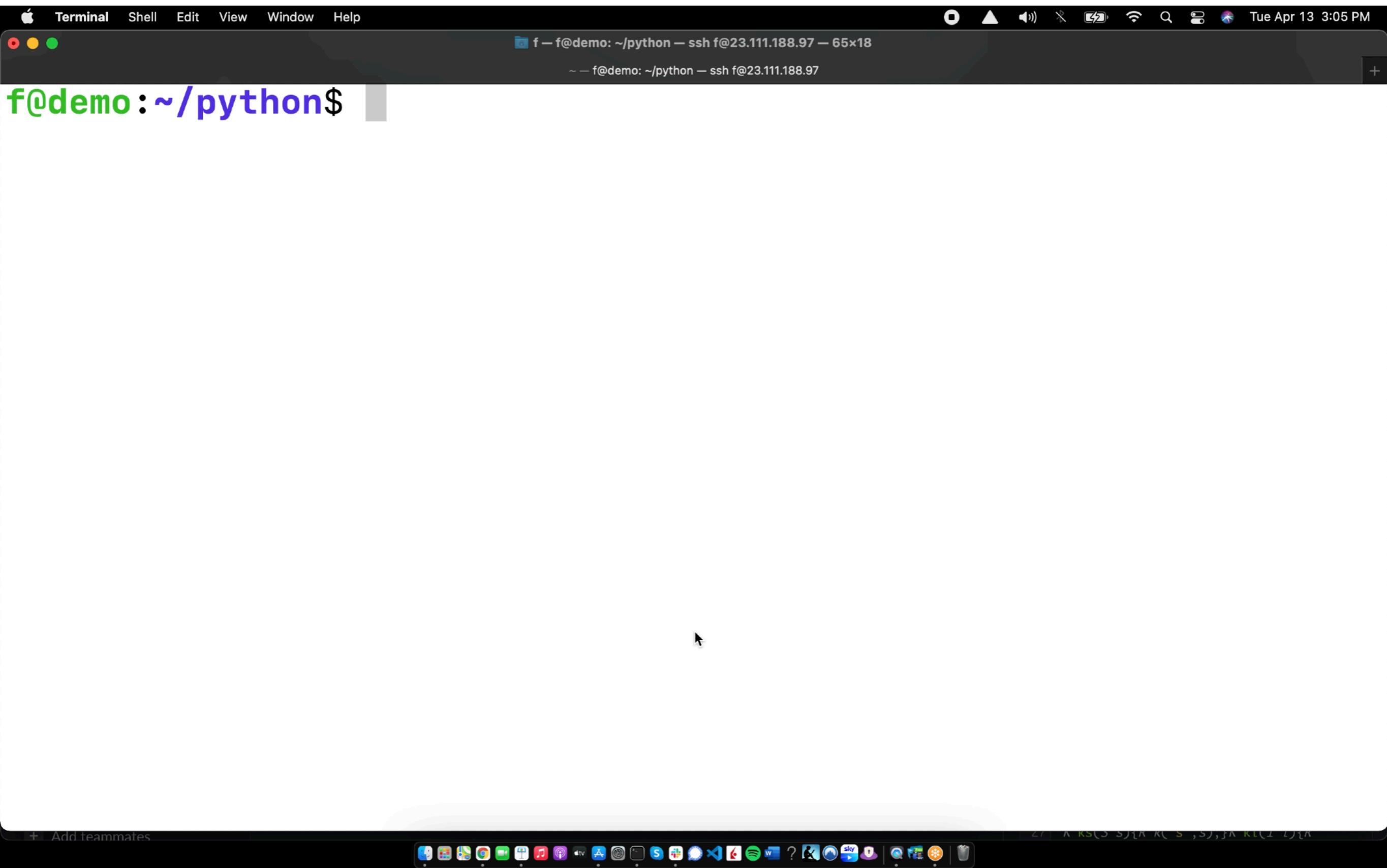
import`csv



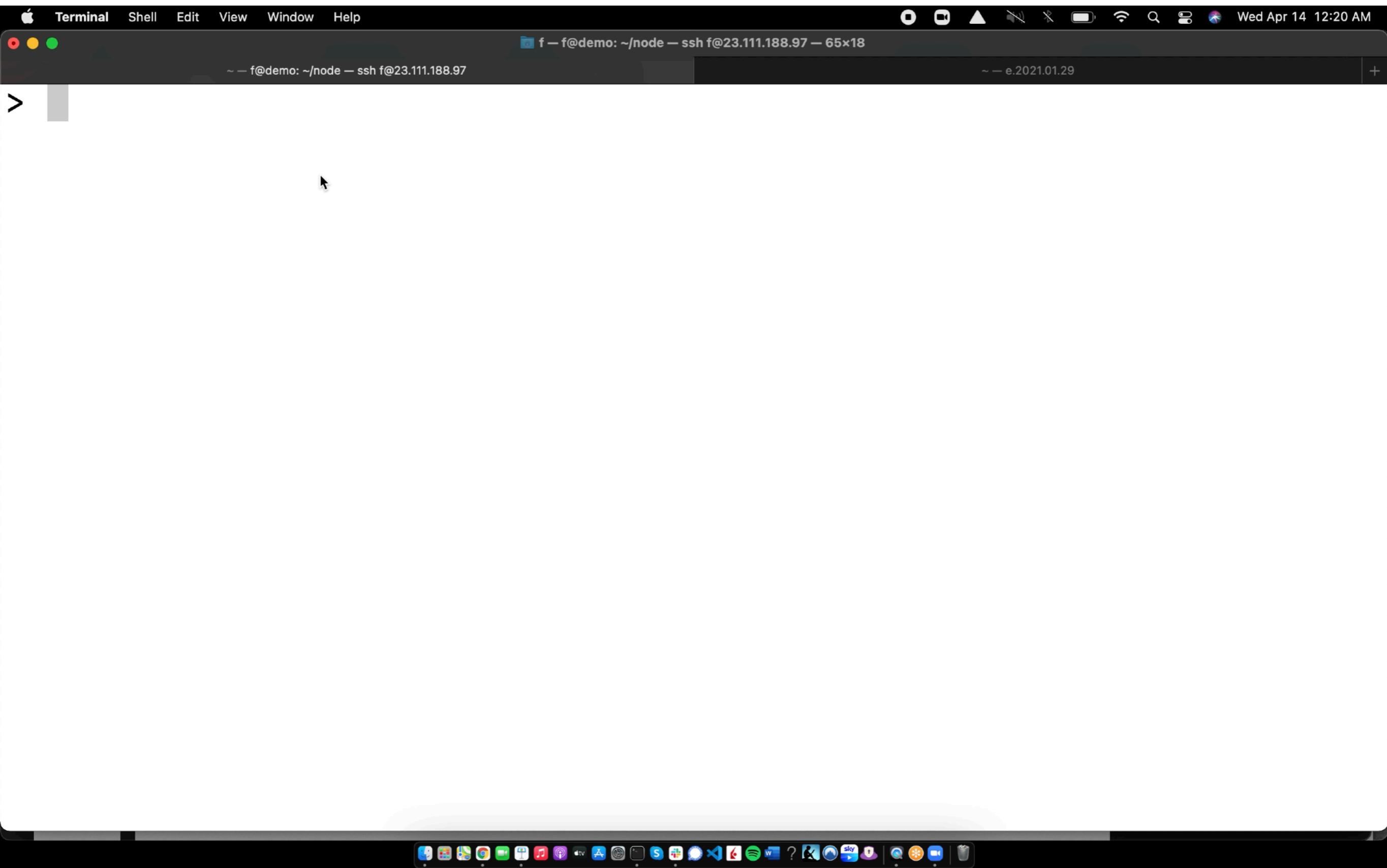
FFI



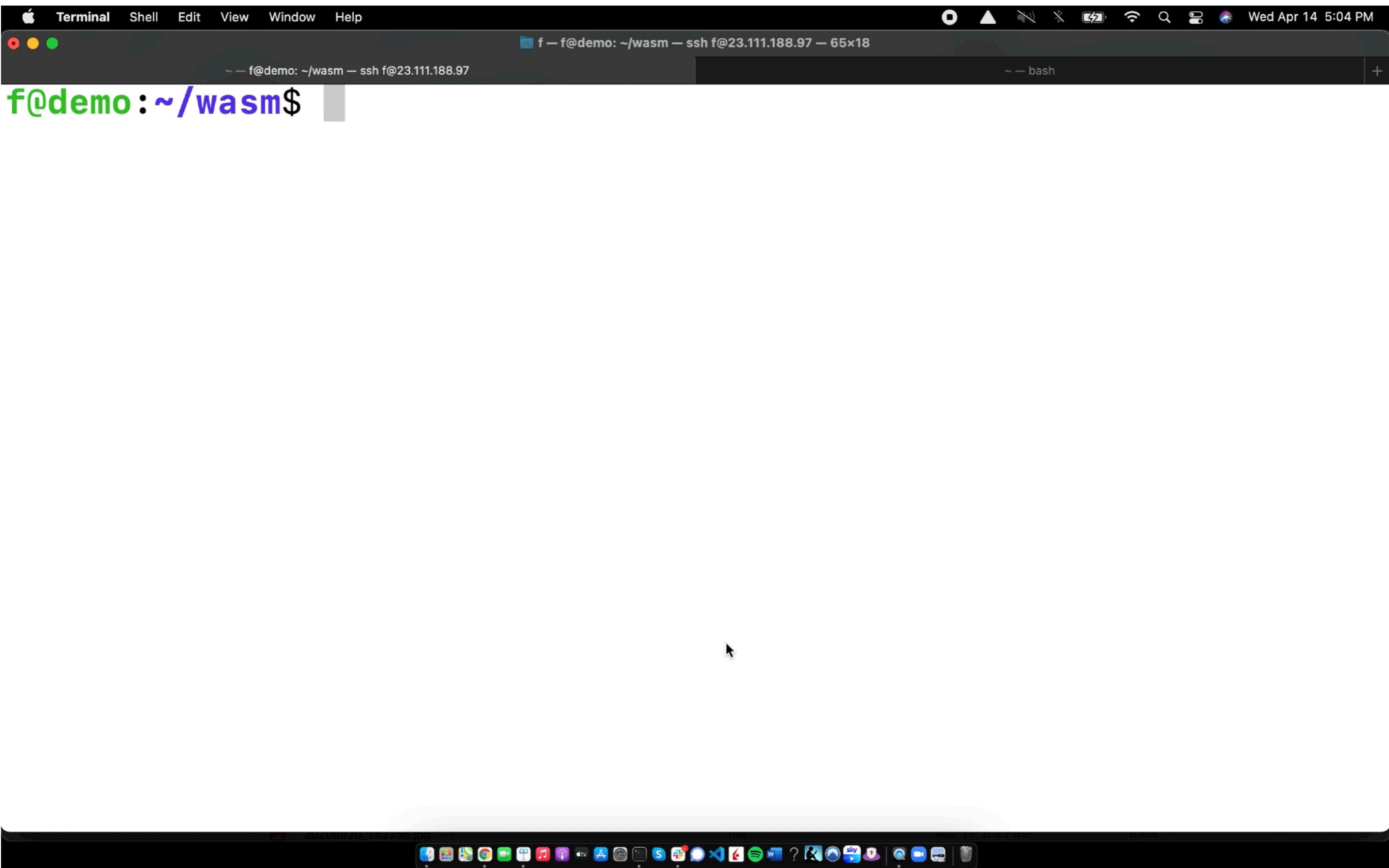
# Python



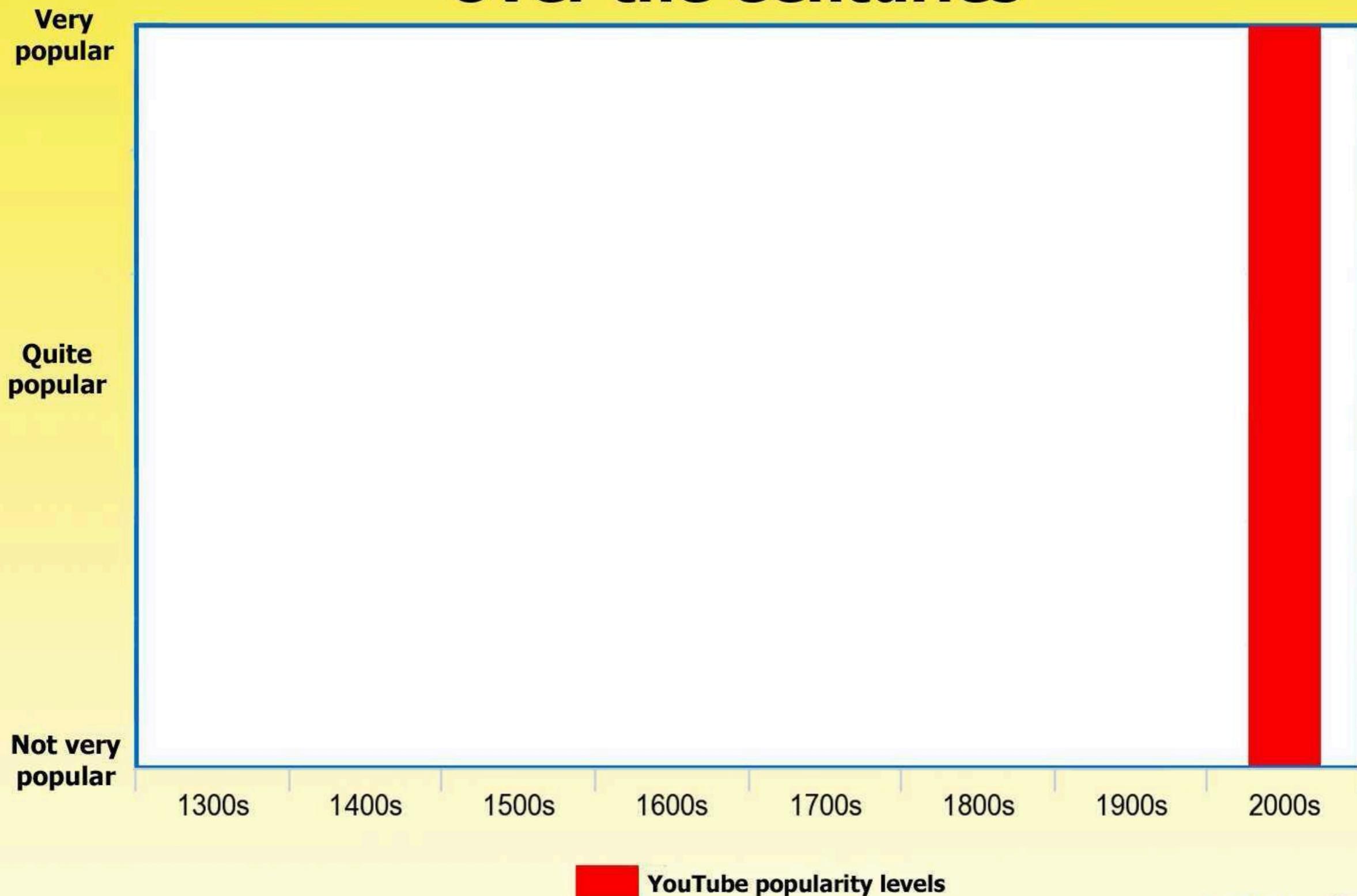
# Node.js



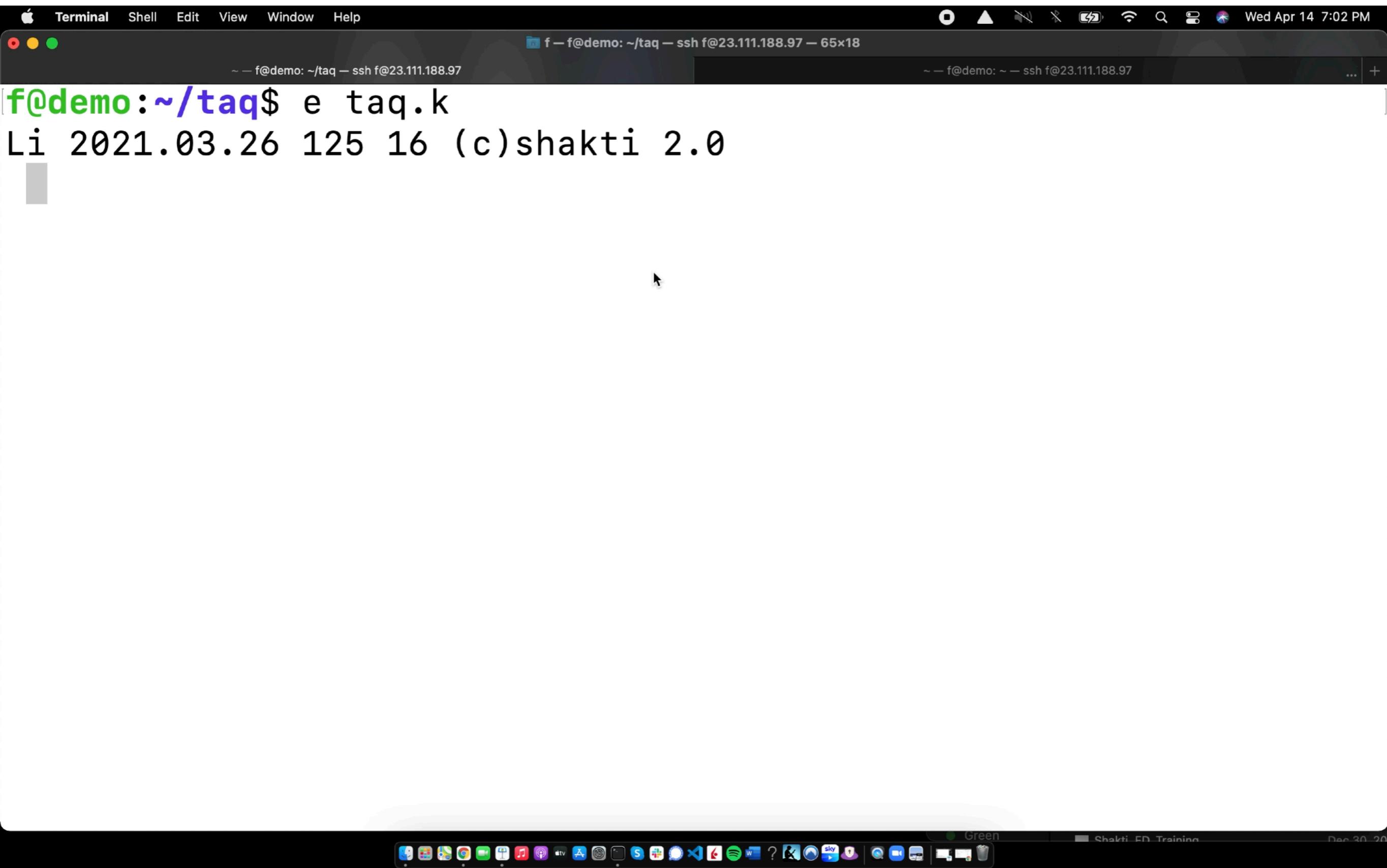
# WebAssembly



# How YouTube Has Increased In Popularity Over the Centuries

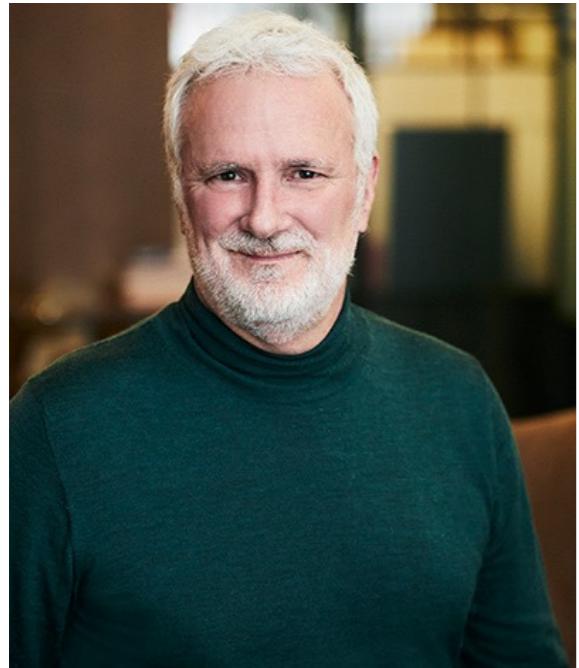


TAQ



Who?

# Team Shakti



Arthur Whitney  
CEO



Janet Lustgarten  
President



Abby Gruen  
COO



Fintan Quill  
Sales

# AquaQ

**Big data firm AquaQ Analytics signs  
customer support partnership with Shakti**



# Partner Program

- Build your new business with Shakti
  - Services
  - Software
- Contact [abby@shakti.com](mailto:abby@shakti.com)

Where?

# Shakti

- [shakti.com](http://shakti.com)

# Community

- [tech@shakti.com](mailto:tech@shakti.com)
- [Shakti Google Group](#)
- [Shakti Meetup Group](#)
- Shakti Partner Program
- Twitter/LinkedIn: @shaktidb

# Documentation

## k9 manual

### Shakti (k9) tutorial

Next: [Introduction](#)

---

## k9: Manual

This manual is for Shakti (k9) build 2021.03.12.

The Shakti programming language is the work of Arthur Whitney and the team at Shakti. The language comes from a lineage of similar languages going back to APL. It's also known as k6, k7 and the fact that Arthur also wrote those. k9 is still under development so expect a bit of syntax change and likely even more improvements.

Learning k9 means code like `{x@(!#x)+\!#y}` is clear and actually preferred to something much longer and verbose. You'll have to go a bit down the rabbit hole before it starts to click. You probably be happy you gave it a chance and end up learning a few things. You'll understand how fast k9 can be in processing and analyzing data and how terse code is less likely to be content using more bloated and slow frameworks.

Alas this manual is not as elegant as the k9 language in that the text version of this document is almost as big as the binary of the language.

The built-in reference screen (\) containing all the k9 commands.

```
python:from k import k;k('+',2,3);nodejs:k=require('k').k;k('+',2,3)
*ffi:"./a.so"5:`a!"fi" //double a(int x){return 2.3+x;}

$ k [-p 1024] a.k
verb          adverb          noun
: x      y      f' each      char " ab"
+ flip   plus  [x]f/ over    c/ join   name ``ab
- minus  minus [x]f\ scan    c\ splt  int 2 3
* first   times [y]f':eachprior   f/:eachright g/:over
% divide divide   f/:eachright g/:over
& where   min/and f\:eachleft g\:scan
| reverse max/or
< asc     less    i/o (*enterprise)
> desc   more    0: r/w line (N;C)0:
= group   equal   1: r/w char
~ not     match   *2: r/w data
! key     key    *3: k-ipc set
, enlist  cat     *4: https get
                                         class
                                         list (2;3.4;`c)
                                         dict [n:`b;i:2]
                                         func {[a;b]a+b}
                                         expr :a+b
                                         \f
                                         \fl x
                                         \fc x
                                         \fs x
                                         \cd [d]
```



# Documentation

## kparc crash course

☰ README.md

### k crash course

a gentle introduction to k9 language is elsewhere.

#### genesis

- [k → who | why | wha | how](#)

#### exodus

- [get | run](#)
- [style → cmt | sep | tab | ids | space | bad](#)
- [parlance → xyz | rank | proj | +: | scope | nouns](#)

#### numbers

- vector math → [v≠a | v+v | v+a | v x | shp](#)
- [type system → num | char | name | date | dict | tab | λ | Ø,∞ | mix | cast](#)
- order of eval → [rtl | \(1\)2](#)
- [no stinking loops → over | scan | e⇒ | e➡ | e↔](#)

#### proverbs

- reading code → [how to solve it](#)
- writing code → [three triangles](#)
- metrics → [apples and oranges](#)
- learning plan → [gladly beyond](#)

#### Releases

No releases published

#### Packages

No packages published

#### Contributors 5



When?

# Now

- Free Version
  - [shakti.com](http://shakti.com)
- Enterprise Version
  - Contact [sales@shakti.com](mailto:sales@shakti.com)
  - Proof Of Concept

# Licensing

- Subscription based pricing
  - small, medium & large server
  - 1, 3 & 5 year
- Run Anywhere
- Early Adopter Program



[sales@shakti.com](mailto:sales@shakti.com)

 SHAKTI