

The Right Tool for the Job

Patrick Pfeifer

2023-03-03

advent of code

<https://adventofcode.com/>

<https://github.com/Pfiver/advent-of-code-2021>

Advent of Code 2021

<https://adventofcode.com/2021/>

Skip the libraries!

```
/usr/bin/env java --source 17 /Users/patrick/Code/advent-of-code-2021/runit

day1.SonarSweep#1      : 1228      (computed in 9 milliseconds)
day1.SonarSweep#2      : 1257      (computed in 24 milliseconds)
day2.Dive#1            : 2070300   (computed in 4 milliseconds)
day2.Dive#2            : 2078985210 (computed in 7 milliseconds)
day3.BinaryDiagnostic#1 : 3309596   (computed in 2 milliseconds)
day3.BinaryDiagnostic#2 : 2981085   (computed in 8 milliseconds)
day4.GiantSquid#1     : 69579    (computed in 11 milliseconds)
day4.GiantSquid#2     : 14877    (computed in 18 milliseconds)
day5.HydrothermalVenture#1 : 6397    (computed in 43 milliseconds)
day5.HydrothermalVenture#2 : 22335   (computed in 133 milliseconds)
day6.Lanternfish#1    : 376194   (computed in 1 milliseconds)
day6.Lanternfish#2    : 1693022481538 (computed in 2 milliseconds)
day7.TheTreacheryofWhales#1 : 356922   (computed in 16 milliseconds)
day7.TheTreacheryofWhales#2 : 100347031 (computed in 37 milliseconds)
```

Advent of Code [About] [Events] [Shop] [Log In]
<y>2021</y> [Calendar] [AoC++] [Sponsors] [Leaderboard] [Stats]

Here are the current completion statistics for each day. Gold indicates users that have completed both parts of a puzzle, while silver indicates users that have completed only the first half. Each * or * star represents up to 6211 users.

25	11632	6143	***
24	13461	134	****
23	13921	2183	****
22	16403	6093	****
21	20600	6554	*****
20	23609	382	*****
19	17552	259	****
18	26139	184	*****
17	34092	1511	*****
16	34089	1734	*****
15	40041	3634	*****
14	47683	7213	*****
13	53692	935	*****
12	51584	2134	*****
11	61084	345	*****
10	68053	1915	*****
9	66763	10821	*****
8	71349	12628	*****
7	91619	2516	*****
6	92953	7228	*****
5	94636	4867	*****
4	107569	6629	*****
3	138842	39004	*****
2	196070	8070	*****
1	219658	28760	*****

stock portfolio

```
def print_statement(tx, v):
    pf = {}
    for amount, company, date in tx:
        total = amount + (pf[company][1] if company in pf else 0)
        pf[company] = [company, total, v[company], total * v[company],
                       f"{'bought' if amount > 0 else 'sold'} {abs(amount)} on {date}"]
    print("\n".join(
        ("company | shares | current price | current value | last operation",) +
        tuple(f"{p[0]} | {p[1]} | ${p[2]:,.2f} | ${p[3]:,.2f} | {p[4]}" for p in pf.values())))
    )
```

stock portfolio test

```
import stock_portfolio

tx = ((1000, "Old School Waterfall Software LTD", "14/02/1990"),
      (400, "Crafter Masters Limited", "09/06/2016"),
      (700, "XP Practitioners Incorporated", "10/12/2018"),
      (-500, "Old School Waterfall Software LTD", "11/12/2018"))

v = {"Old School Waterfall Software LTD": 5.75,
     "Crafter Masters Limited": 17.25,
     "XP Practitioners Incorporated": 25.55}

stock_portfolio.print_statement(tx, v)
```

data centricism

Ralph Westphal (.net guy)

- IOSP Integration Operation Segregation Principle

-> <https://www.youtube.com/watch?v=MoBNdW2DhQo>

-> <https://ralfwestphal.substack.com/p/integration-operation-segregation>

- Lambda Calculus (Alonso Church, Princeton, 1930s)

(OOP: Alan Kay, MIT, late 1950s)

managed languages for modern cpus

Martin Thompson

- Event Log Architectures | DDD Europe 2020

-> <https://www.youtube.com/watch?v=RlwO6CJbJjQ>

- High Performance Managed Languages

-> <https://www.youtube.com/watch?v=Pz-4co8lal8>

data analysis and workflow automation tools

Integration

- bash
- zsh

Text processing

- sed
- find
- grep

Data retrieval and processing

- curl
- jq
- xmlstarlet

(serious) Data processing

- awk
- perl (and the famous PCRE)
- python

```
#!/bin/sh

# Alt: https://shop.savognin.ch/Savognin/ukv/house/TDS00020010019636954
# Neu: https://savognin.graubuenden.ch/de/node/20583#/unterkuenfte/CH1/c0195db2-2db6-47d5-9453-414a27e7d694/tga-christoffel

api_base=https://webapi.deskline.net/savognin/de/accommodations/CH1/products/bb3b5913-07a0-4893-88c5-6a1d52278ea5

from_date=$(perl -e 'use Time::Piece; use Time::Local; $t=localtime timelocal(0, 0, 0, 1, localtime->_mon, localtime->year); print $t->ymd;')
to_date=$(perl -e 'use Time::Piece; use Time::Seconds; use Time::Local; $t=localtime timelocal(0, 0, 0, 1, localtime->_mon, localtime->year+1); $t-=ONE_DAY; print $t->ymd;')

#api_params="fields=availabilityCalendar(fromDate:\"$from_date\",toDate:\"$to_date\"){date,available,canArrive,canDepart,minStay,maxStay,stayInterval}"
api_params="fields=availabilityCalendar(fromDate:\"$from_date\",toDate:\"$to_date\"){date,available}"

api_params="$api_params&spId=c0195db2-2db6-47d5-9453-414a27e7d694"

curl -gs "$api_base?$api_params" |
python -c 'import sys,json; data=json.load(sys.stdin); json.dump(data, sys.stdout) if data["availabilityCalendar"][0]["available"] in [0,1] else sys.stderr.write("something went wrong\n")'
```


Questions / Discussion

Thank you!

patrick@patrickpfeifer.net

PFiver@mastodon.social

GitHub, LinkedIn: PFiver