Bytes&Bites: Puzzles!

VU VRIJE UNIVERSITEIT AMSTERDAM

LOOKING FURTHER

Introduction

Software community activity at the VU:

Bytes and Bites

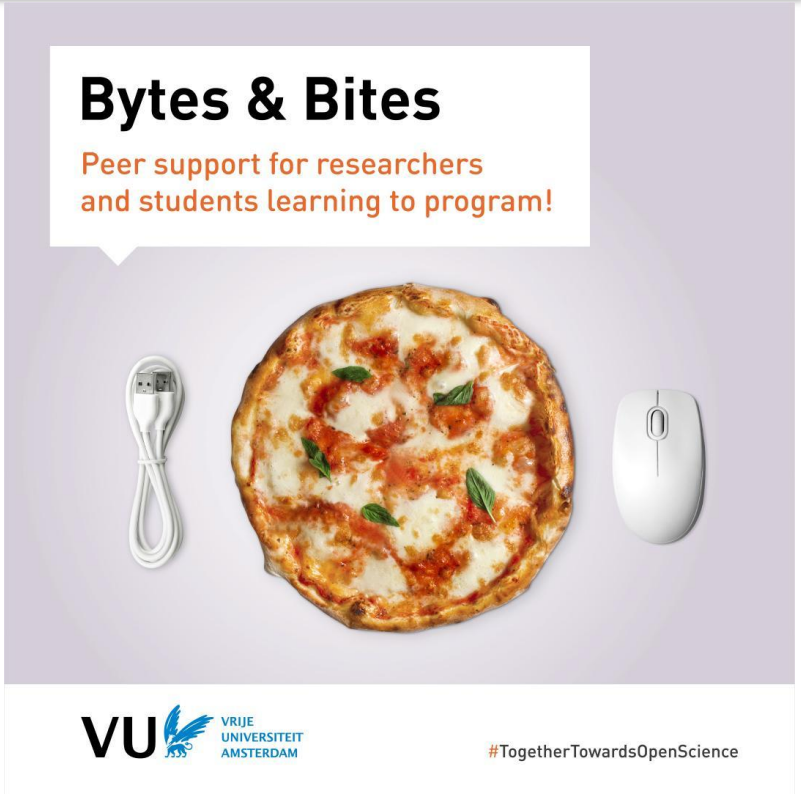
- for all programming languages
- for beginners and experts

Bytes

- presentations on interesting topics
- bring your questions and bugs
- hackathons and coding sprints

Bites

- Pizza!



Introduction

Software community at the VU:

Bytes and Bites

- for all programming languages
- for beginners and experts

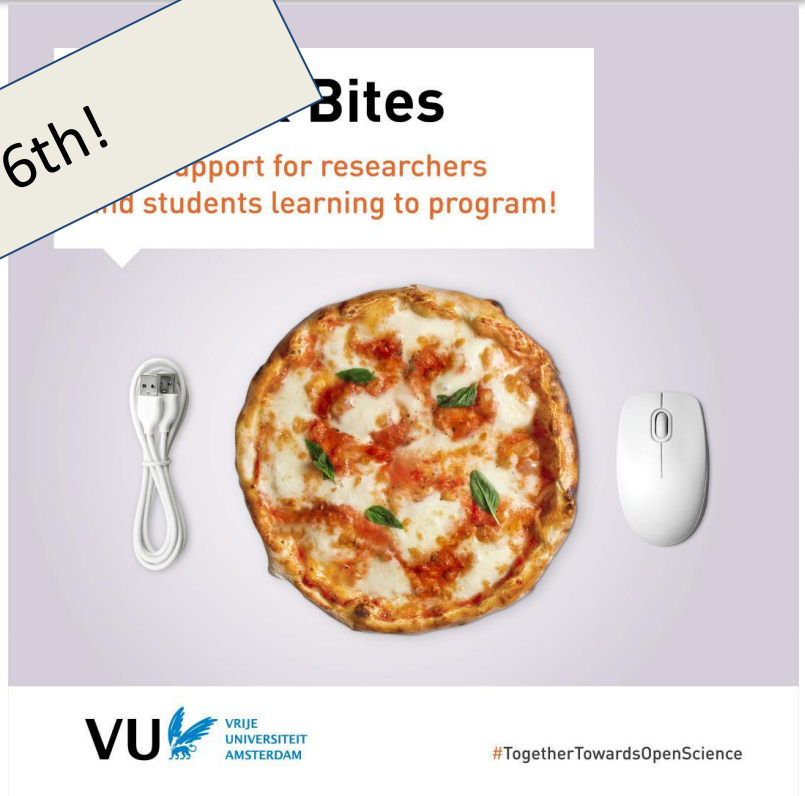
Bytes

- presentations on int
- bring your q
- hacka prints

Bites

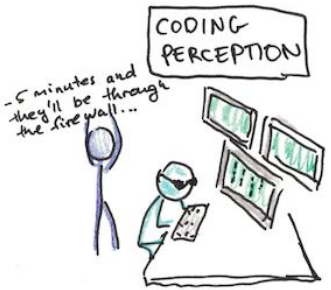
- Piz

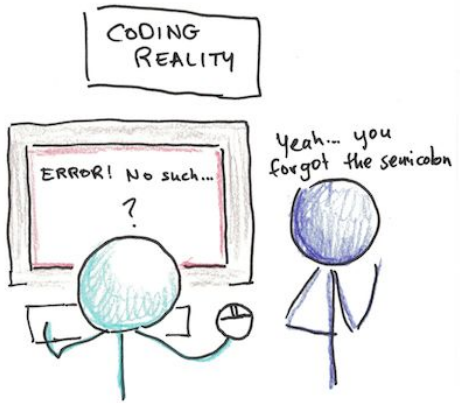
Next Bytes & Bites will be June 6th!



Puzzles in Programming

- learn by doing
- Practical logical thinking
- harder than following a tutorial
  - but easier than building your own big project
- Self paced
- Nice to experiment
- Learn about new algorithms





Bytes&Bites: Puzzles!

Bytes&Bites: Puzzles!

Advent of Code

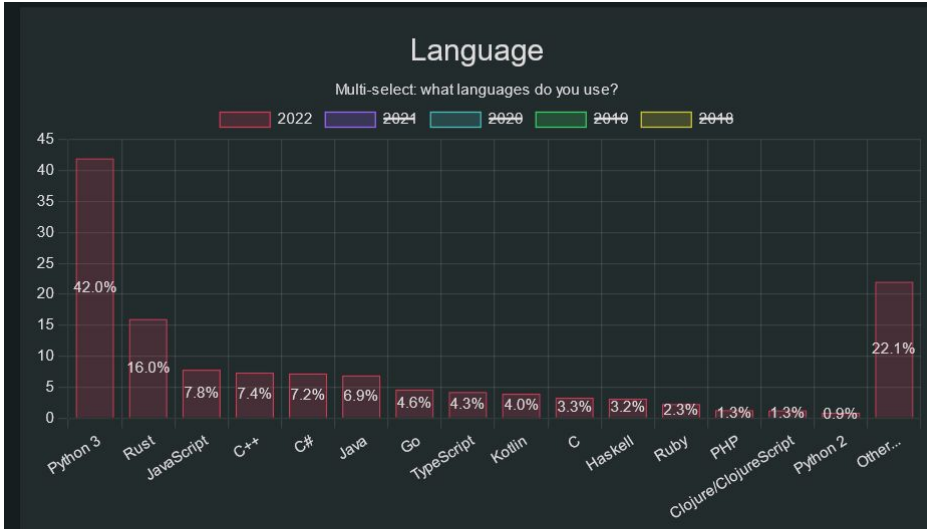
- Annual set of Christmas-themed computer programming challenges
- Started in 2015 with 81 participants and gained popularity since then (e.g. on reddit)
- In 2022 the project reached more than 1 mio. registered users
- Global and private leaderboards
- Great opportunity to learn coding or challenge yourself with a new programming language or more exotic tool (e.g. ChatGPT or minecraft)

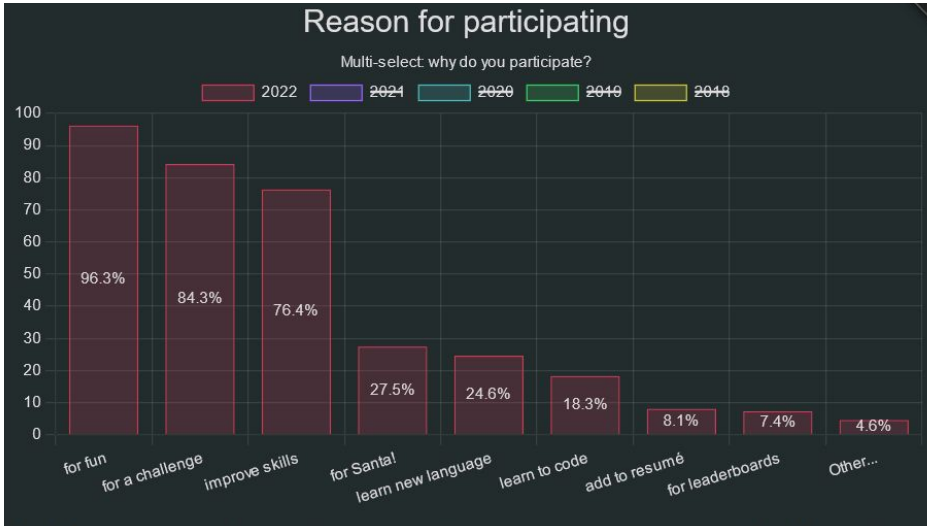
Advent of Code Puzzles

- 2 parts per puzzle
- Puzzle is the same for everyone
  - the answer is different
- Solved any language, any way, in your time
- You earn a gold star (⭐) for each part of the puzzles
- Private & global leaderboards provide a reference or comparison

Advent of Code Puzzles

- You’re not alone!





https://jeroenheijmans.github.io/advent-of-code-surveys/

Advent of Code Puzzles

1. Log in on the Advent of Code website.
2. Read the puzzle text
   - find the puzzle and example
3. Download your personalized input for the puzzle
4. Code up your solution.
5. Enter your answer
6. Repeat steps 2 and 4 for part two of the puzzle
7. Enter your second answer

Solving a puzzle

- Day 1, 2020 puzzle
- you need *find the two entries that sum to 2020* and then multiply those two numbers together.

  For example, suppose your expense report contained the following:

  1721
  979
  366
  299
  675
  1456

  In this list, the two entries that sum to 2020 are 1721 and 299. Multiplying them together produces 1721 * 299 = 514579, so the correct answer is *514579*.

Solving a puzzle

- Day 1, 2020 puzzle

```
SET array numbers to expense numbers
FOR each number1 in numbers
    FOR each number2 in numbers
        IF number1 + number2 == 2020
         THEN
            PRINT number1 * number2
        END IF
    END FOR
END FOR
```

For example, suppose your expense report contained the following:

```
1721
979
366
299
675
1456
```

In this list, the two entries that sum to 2020 are 1721 and 299. Multiplying them together produces 1721 * 299 = 514579, so the correct answer is *514579*.

Solving a puzzle

- Day 1, 2020 puzzle

```
SET array numbers to expense numbers
FOR each number1 in numbers
    FOR each number2 in numbers
        IF number1 + number2 == 2020
        and number1 < number2 THEN
            PRINT number1 * number2
        END IF
    END FOR
END FOR
```

```
For example, suppose your expense report contained the
following:

1721
979
366
299
675
1456


In this list, the two entries that sum to 2020 are 1721
and 299. Multiplying them together produces 1721 * 299
= 514579, so the correct answer is 514579.
```

Solving a puzzle - modular coding

```
FUNCTION parse
  Pass In: puzzle input
  Parse into a usable data structure
  Pass Out: numbers
END FUNCTION

FUNCTION part1
  Pass In: numbers
  FOR each number1 in numbers
    FOR each number2 in numbers
        IF number1 + number2 == 2020
        and number1 < number2 THEN
          Pass Out: number1 *
        number2
        END IF
    END FOR
  END FOR
END FUNCTION
```

```
FUNCTION main
  Pass In: filenames
  FOR each file in filenames
    READ text files to puzzleinput
    CALL parse
    CALL part1
  END FOR
  Pass Out: answer
END FUNCTION
```

Solving a puzzle - modular coding template

```
FUNCTION parse
  Pass In: puzzle input
  Parse into a usable data structure
  Pass Out: numbers
END FUNCTION

FUNCTION part1
  Pass In: data
  Pass Out: solution1
END FUNCTION

FUNCTION part2
  Pass In: data
  Pass Out: solution1
END FUNCTION
```

```
FUNCTION solve
  Pass In: puzzle input
  CALL parse
  CALL part1
  CALL part2
  Pass Out: solution1 and solution2

FUNCTION main
  Pass In: filenames
  FOR each file in filenames
    READ text files to puzzleinput
    CALL solve
  END FOR
  Pass Out: solutions
END FUNCTION
```

Solving a puzzle - modular coding template

```
FUNCTION parse
  Pass In: puzzle input
  Parse into a usable data structure
  Pass Out: numbers
END FUNCTION

FUNCTION part1
  Pass In: data
  Pass Out: soluti
END FUNCTION

FUNCTI
  Pass
  Pass      olution1
END FUNCTION
```

```
FUNCTION sol
  Pass I                t
  CA
            t2
       Out: solution1 and solution2

FUNCTION main
  Pass In: filenames
  FOR each file in filenames
    READ text files to puzzleinput
    CALL solve
  END FOR
  Pass Out: solutions
END FUNCTION
```

You can also write automated tests!

Let’s start puzzling

- Read very carefully
- Use the example data first
- Break it up in smaller steps
- Work alone or in teams
- Our suggestions to get started:
  a. AoC 2020 day 1
  b. AoC 2019 day 1
  c. AoC 2020 day 5
  d. AoC 2021 day 5