

# Sprint

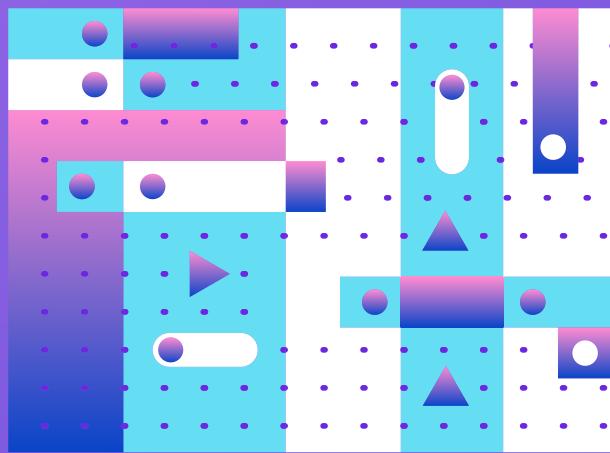
A Startup Programming Language

Rodrigo Guimarães Coelho

# 01.

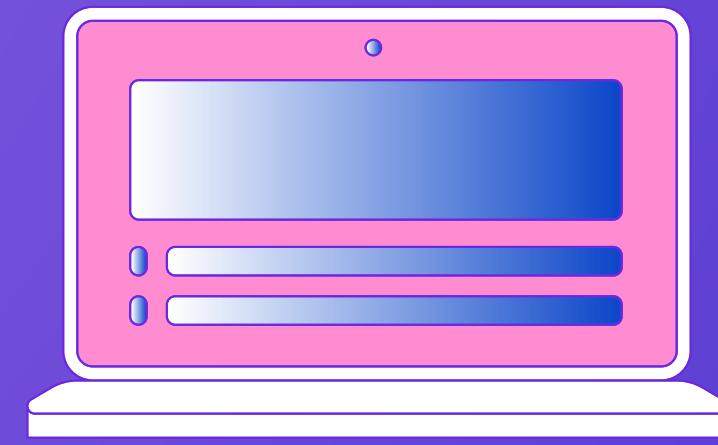
## Motivation

# Why create a language?



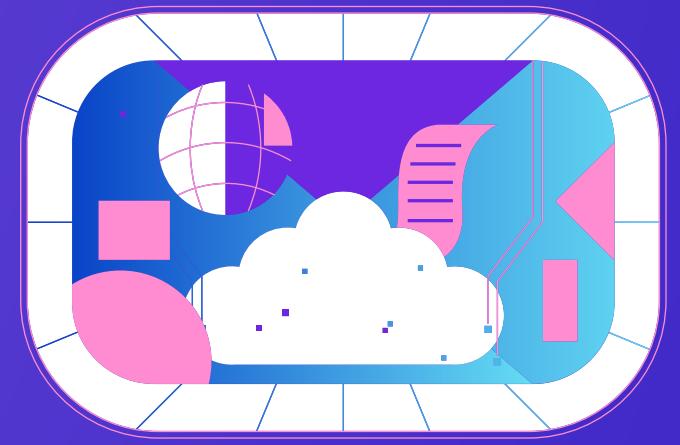
## Manage startup squads

Managing teams is frequently a huge pain for startup managers



## Code for freedom

Create a language gives those managers more freedom to organize themselves and the team how they want to

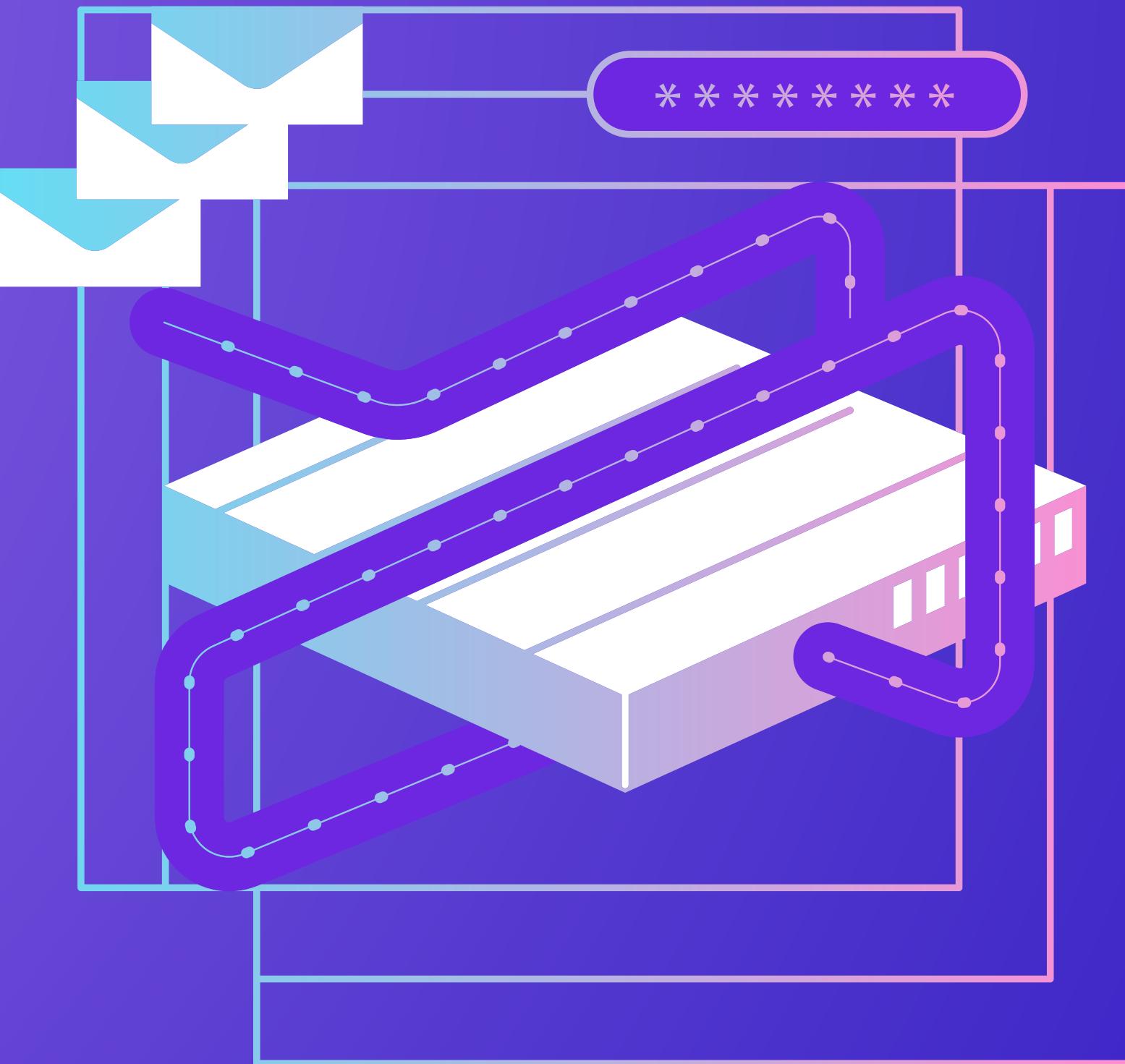


## Trigger tasks creation

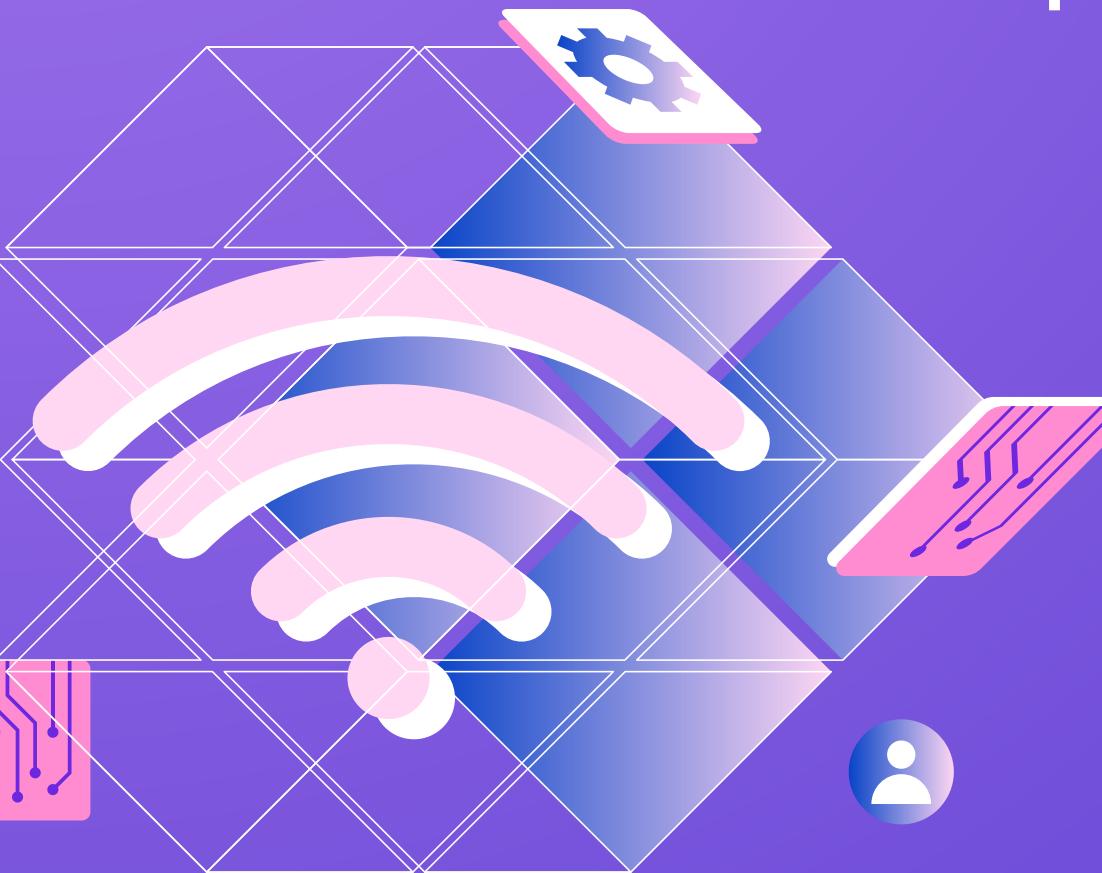
Besides freedom, managers need tools that make their life easier by automating repetitive processes

02.

## Curiosities



# More about the managing world



*“The average manager spends 1.83 hours/day on productive task work, or 9.2 hours a week. On average, managers waste 1.82 hours/day on unproductive task work like checking emails, Slack, or sifting through their task list.”*

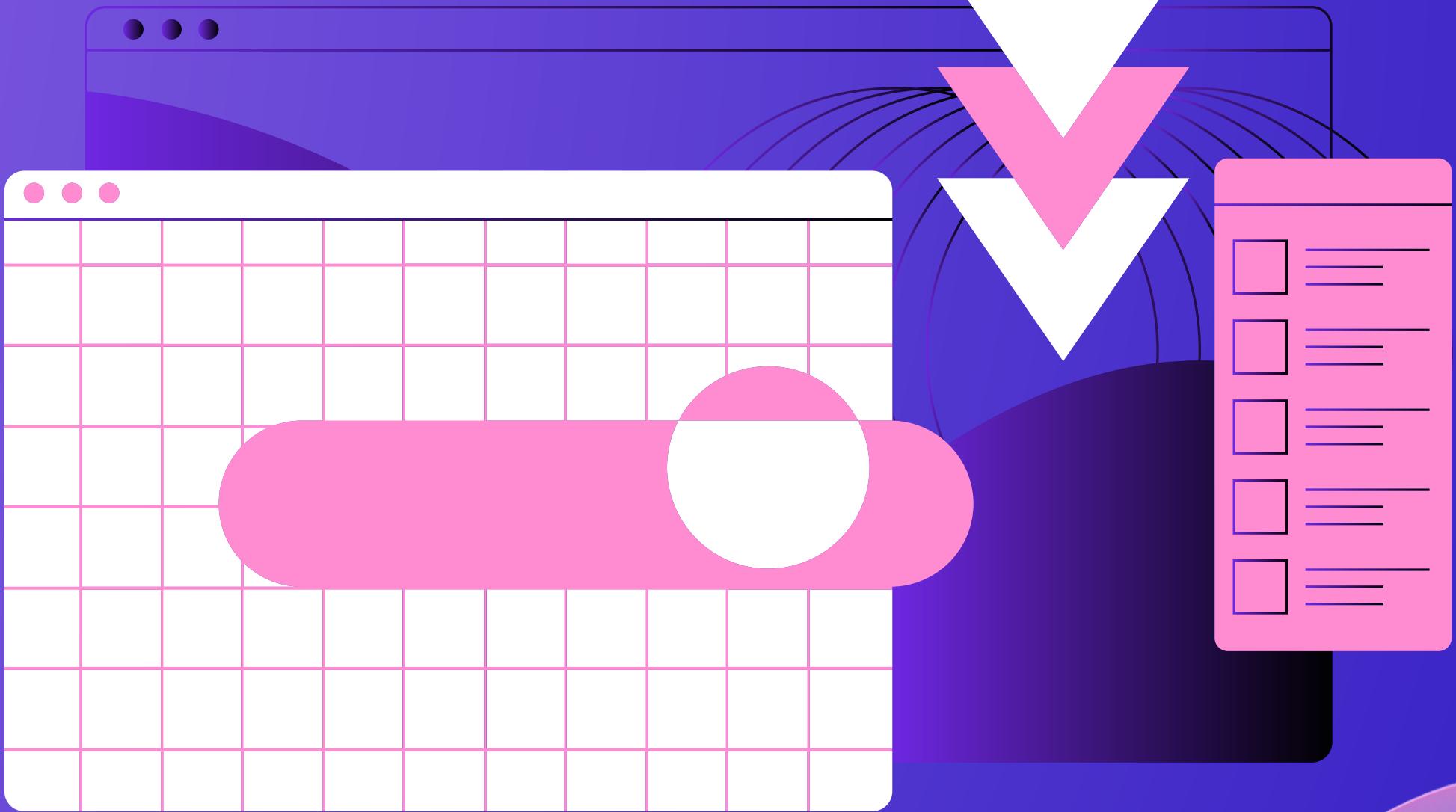
Reclaim.AI

*“14.6% of workers report that their manager is ineffective at managing their time, making it the third most common skill managers lack behind team building and providing feedback.”*

The Predictive Index

# 03.

## Characteristics



# What you can do with **Sprint**?

01.

Create squads

02.

Create employees

03.

Connect squads  
and employees

04.

Create tasks

05.

Update task  
props

06.

Iterate over tasks

07.

Check tasks  
status

08.

Trigger new tasks  
programmatically

04.

Example

# What it does?

```
START_SPRINT

// Entities
squad Website
squad AI

employee Rodrigo
employee Joao
employee Marcos
employee Pedro

// Link employee to squad
Website add Rodrigo
Website add Joao

AI add Rodrigo
AI add Marcos
AI add Pedro
AI remove Rodrigo

// Tasks
task "Start front-page" to Rodrigo on Website is "TODO"
task "Start backend" to Joao on Website is "TODO"

task "Create model" to Marcos on AI is "TODO"
task "Clean data" to Pedro on AI is "TODO"

// Status
set "Start front-page" from Rodrigo on Website to "DONE"
set "Start backend" from Joao on Website to "DOING"

// Loop
for t in tasks from Rodrigo {
  if (t.name == "Develop front-page") {
    if (t.status == "DONE") {
      print(t.owner + " ended '" + t.name + "'")
      task "Deploy website" to Rodrigo on Website is "TODO"
    }
  }
}
END_SPRINT
```

For this example we are creating **2 squads**, allocating **4 employees** and adding a **single task for each** employee. We assign each employee for a squad, and then, after the setup phase, we change the status of 2 tasks. Finally we **iterate over all tasks from a specific employee** to check if he has ended or not its task, if yes, we then **add another task**.



# How to use it?

```
START_SPRINT

// Entities
squad Website
squad AI

employee Rodrigo
employee Joao
employee Marcos
employee Pedro

// Link employee to squad
Website add Rodrigo
Website add Joao

AI add Rodrigo
AI add Marcos
AI add Pedro
AI remove Rodrigo

// Tasks
task "Start front-page" to Rodrigo on Website is "TODO"
task "Start backend" to Joao on Website is "TODO"

task "Create model" to Marcos on AI is "TODO"
task "Clean data" to Pedro on AI is "TODO"

// Status
set "Start front-page" from Rodrigo on Website to "DONE"
set "Start backend" from Joao on Website to "DOING"

// Loop
for t in tasks from Rodrigo {
  if (t.name == "Develop front-page") {
    if (t.status == "DONE") {
      print(t.owner + " ended '" + t.name + "'")
      task "Deploy website" to Rodrigo on Website is "TODO"
    }
  }
}
END_SPRINT
```

## Create a squad

squad <squad\_name>

## Create an employee

employee <employee\_name>

## Link squad-employee

<squad\_name> add <employee\_name>

## Unlink squad-employee

<squad\_name> remove <employee\_name>

## Create task

task "<task\_name>" to <employee\_name> on <squad\_name> is <task\_status>

## Update task

set "<task\_name>" from <employee\_name> on <squad\_name> to <task\_status>

## Iterate over tasks

for <var> in tasks from <employee\_name> {

## Access task props

<var>.name -> task name

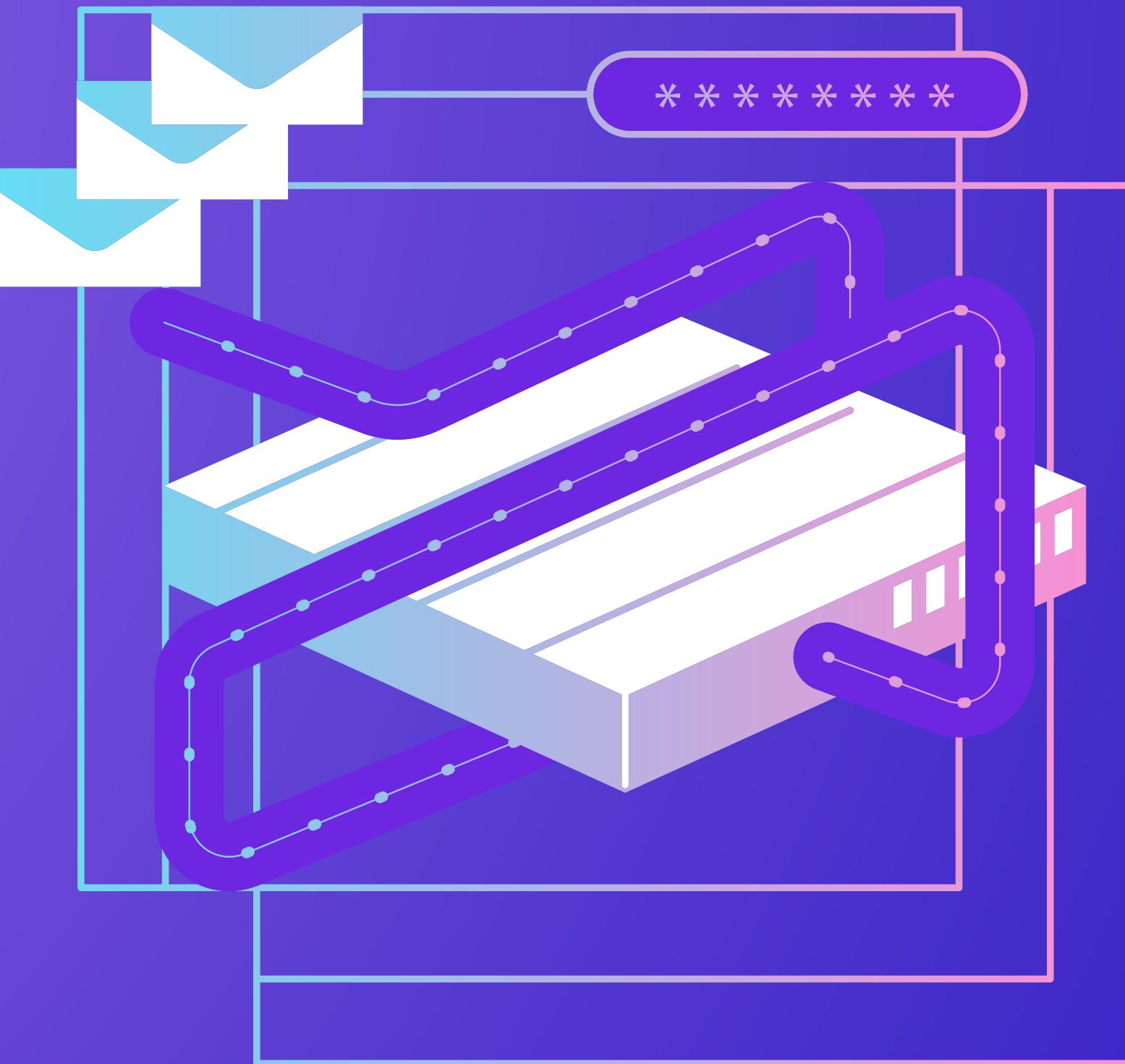
<var>.status -> task status

<var>.owner -> task owner



# 05.

## Results



# What you get as result?



This would be the only output if -v wasn't passed as argument

```
python main.py ./day1.sprint -v
```

```
Sprint started

Created squad 'Website'
Created squad 'AI'
Created employee 'Rodrigo'
Created employee 'Joao'
Created employee 'Marcos'
Created employee 'Pedro'
Added employee 'Rodrigo' to squad 'Website'
Added employee 'Joao' to squad 'Website'
Added employee 'Rodrigo' to squad 'AI'
Added employee 'Marcos' to squad 'AI'
Added employee 'Pedro' to squad 'AI'
Removed employee 'Rodrigo' from squad 'AI'
Created task 'Start front-page' for employee 'Rodrigo' in squad 'Website' with status 'TODO'
Created task 'Start backend' for employee 'Joao' in squad 'Website' with status 'TODO'
Created task 'Create model' for employee 'Marcos' in squad 'AI' with status 'TODO'
Created task 'Clean data' for employee 'Pedro' in squad 'AI' with status 'TODO'
Updated task 'Start front-page' from 'Rodrigo' in squad 'Website' to status 'DONE'
Updated task 'Start backend' from 'Joao' in squad 'Website' to status 'DOING'
Rodrigo ended 'Start front-page'
Created task 'Deploy website' for employee 'Rodrigo' in squad 'Website' with status 'TODO'

Sprint ended
```

Flag -v makes the output more verbose

# Thank you!

Hope you sky rocket your startup using **Sprint**