Logilab

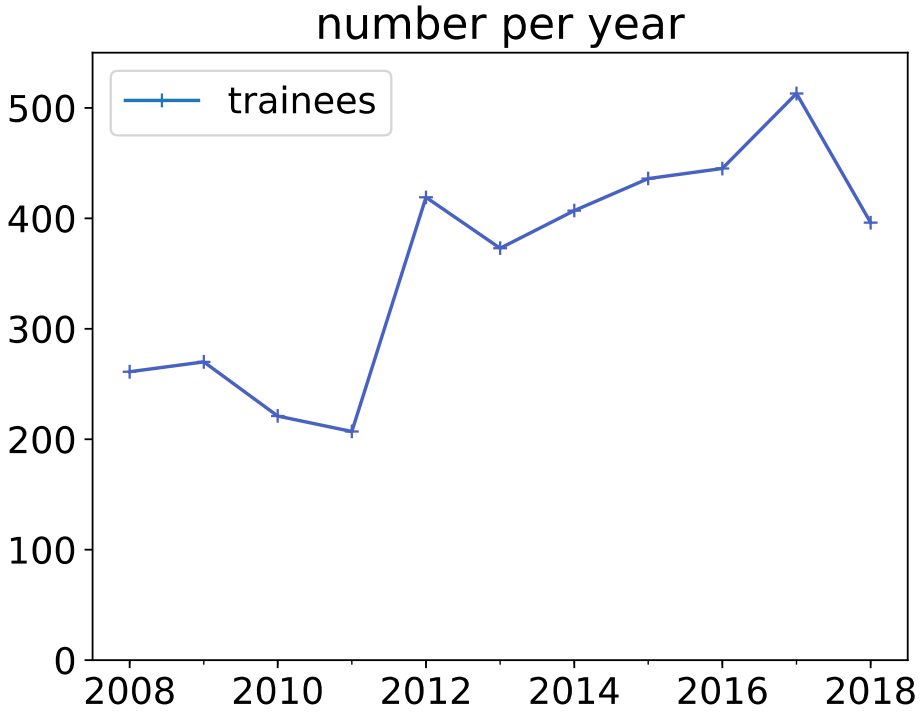
- French private SME
- Founded in 2000
- 20 people
- 2 M€ of annual turnover

Activity:

- Software development
  - Python, JavaScript, etc.
- Consulting
- Professional trainings
  - since 2000
  - 20 to 40% of annual turnover

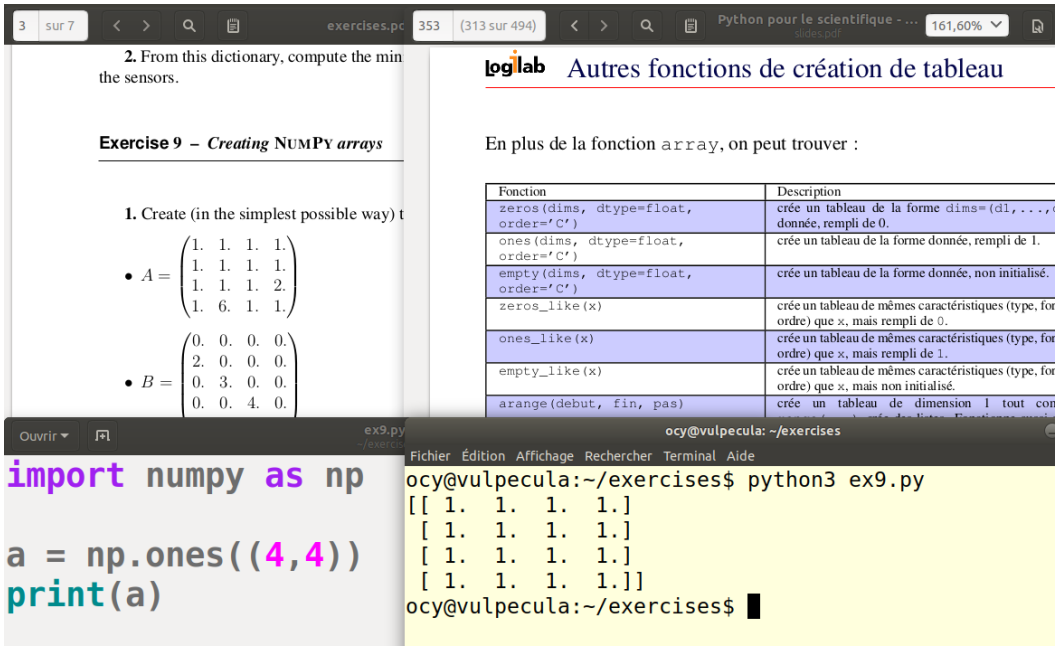
Profile of trainees:

- Engineers, scientists, etc.
- Already working in big companies or organisations
- Wanting to enhance their skills:
  - learning Python programming
  - improving their Python knowledge



At first:

- Sessions of 4 or 5 contiguous days
- PDF slides
- PDF exercises manual
- Exercises in regular Python files



Problems:

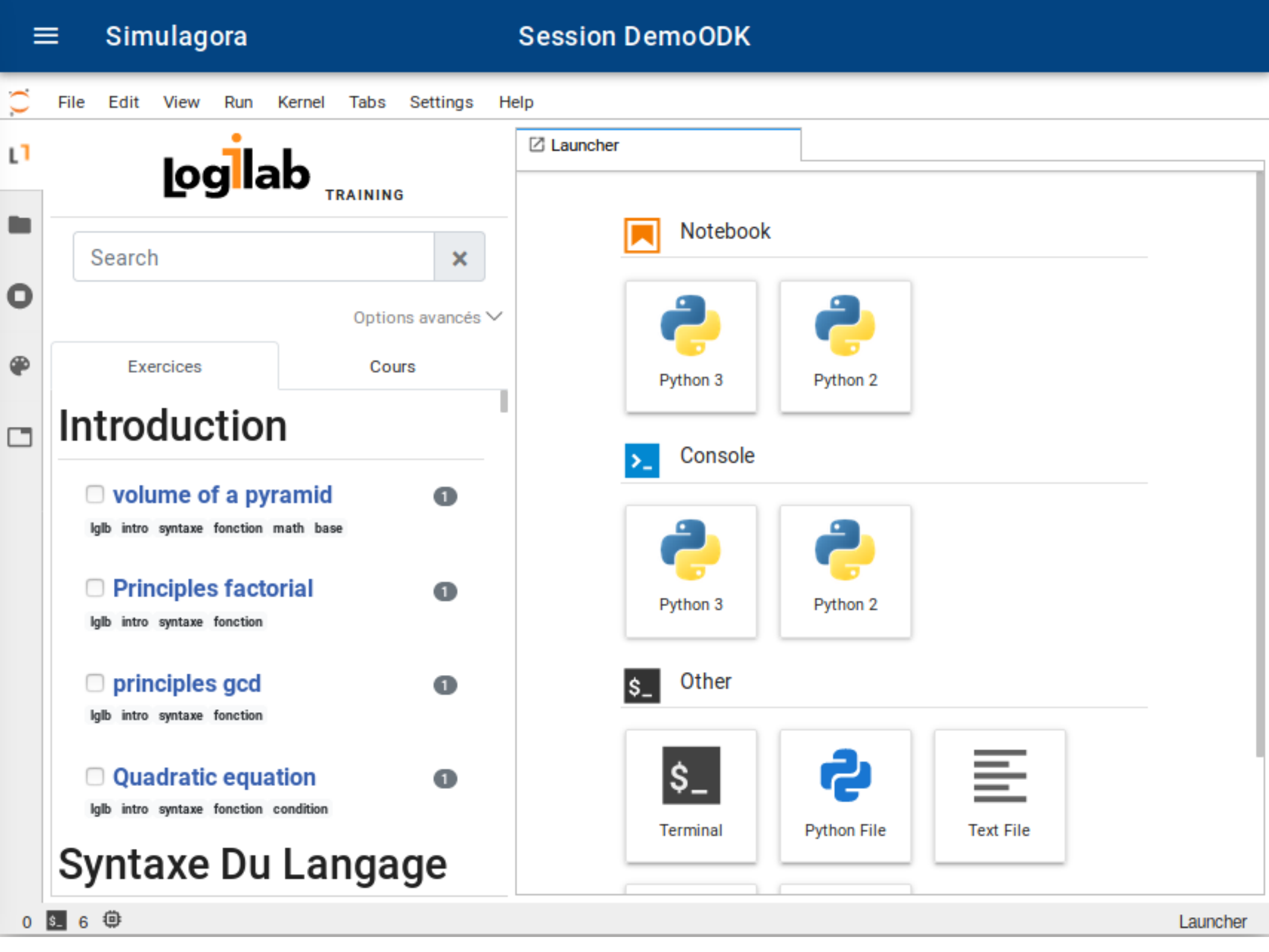
- Difficulty to set up environment for exercises
  - Python, various libraries, editors, IDEs
- Lack of interactivity

OPEN DREAMKIT

- Virtual environments
  - accessible through a simple Web browser
  - containing a set of integrated libraries
- Interactive tools (e.g. Jupyter and its satellites)
  - usable for demos, exercises, course
  - providing a better learning experience

Nowadays:

- Virtual environments dedicated to trainings
  - hosted on https://www.simulagora.com/
  - specific to each person
  - using Jupyter, JupyterHub, JupyterLab
- Specific JupyterLab component dedicated to exercises
  - https://gitlab.com/logilab/jupyterlab-training
    - free software
    - usable with any Jupyter kernel
  - list of available exercises
    - tags, categories, difficulty
  - each exercise described in a Jupyter notebook
    - description and questions
    - cell for writing the code
    - button for executing automatic tests
    - button to display the solution


Simulagora
Session DemoODK
File Edit View Run Kernel Tabs Settings Help
Logilab TRAINING
Search
Options avancés
Exercices
Cours
Introduction
volume of a pyramid
lglb intro syntaxe fonction math base
Principles factorial
lglb intro syntaxe fonction
principles gcd
lglb intro syntaxe fonction
Quadratic equation
lglb intro syntaxe fonction condition
Syntaxe Du Langage
Launcher
Notebook
Python 3
Python 2
Console
Python 3
Python 2
Other
Terminal
Python File
Text File
0 6
Launcher

Launcher | Principles factorial

Flake8 Code Python 3

Logilab TRAINING

Search

Options avancés

Exercices Cours

Introduction

- volume of a pyramid 1
  lglb intro syntaxe fonction math base
- Principles factorial 1
  lglb intro syntaxe fonction
- principles gcd 1
  lglb intro syntaxe fonction
- Quadratic equation 1
  lglb intro syntaxe fonction condition

Syntaxe Du Langage

Principles factorial

Write a `fact` function that computes the factorial $n!$ of a non-negative integer n.

Note: $n! = n \times (n-1) \times \cdots \times 1$ and $0! = 1! = 1$.

For this exercise, if n is negative, `fact(n)` will return 1.

```
[ ]: # Your code
```

show Unittests

```
(Re) Run the cell containing your function before launch
ing unit tests
```

✔ Execute tests

show Solution

File Edit View Run Kernel Tabs Settings Help

Logilab TRAINING

Search

Options avancés

Exercices Cours

Introduction

- volume of a pyramid 1
  lglb intro syntaxe fonction math base
- Principles factorial 1
  lglb intro syntaxe fonction
- principles gcd 1
  lglb intro syntaxe fonction
- Quadratic equation 1
  lglb intro syntaxe fonction condition

Syntaxe Du Langage

Launcher | Principles factorial

Flake8 Code Python 3

For this exercise, if *n* is negative, `fact(n)` will return 1.

[4]:
```
# Your code
def fact(n):
    if n == 0:
        return 1
    return n * fact(n-1)
```

show Unittests

(Re) Run the cell containing your function before launching unit tests

✔ Execute tests

1 tests failed ✖ Invalid

```
E.
======================================================================
ERROR: test_fact_negative_integer (__main__.FactTest)
Check that factorial of a negative integer is -1.
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-3-8f18b3c802ee>", line 20, in test_negative_integer
    self.assertEqual(fact(-1), 1)
  File "<ipython-input-4-9e1216538ebd>", line 5, in fact
    return n * fact(n-1)
  File "<ipython-input-4-9e1216538ebd>", line 5, in fact
```

0 3 Python 3 | Idle Mode: Command Ln 1, Col 1 README.en.ipynb

Logilab TRAINING

Search

Options avancés

Exercices Cours

Introduction

- volume of a pyramid 1
  lglb intro syntaxe fonction math base
- Principles factorial 1
  lglb intro syntaxe fonction
- principles gcd 1
  lglb intro syntaxe fonction
- Quadratic equation 1
  lglb intro syntaxe fonction condition

Syntaxe Du Langage

Launcher | Principles factorial

Flake8 Code Python 3

For this exercise, if *n* is negative, `fact(n)` will return 1.

[5]:
```
# Your code
def fact(n):
    if n <= 0:
        return 1
    return n * fact(n-1)
```

show Unittests

(Re) Run the cell containing your function before launching unit tests

✔ Execute tests

Congratulations! ✔

show Solution

Logilab TRAINING

Search

Options avancés

Exercices Cours

Introduction

- volume of a pyramid 1
  lglb intro syntaxe fonction math base
- Principles factorial 1
  lglb intro syntaxe fonction
- principles gcd 1
  lglb intro syntaxe fonction
- Quadratic equation 1
  lglb intro syntaxe fonction condition

Syntaxe Du Langage

Launcher | Principles factorial

Flake8 Code Python 3

For this exercise, if *n* is negative, `fact(n)` will return 1.

[5]:
```
# Your code
def fact(n):
    if n <= 0:
        return 1
    return n * fact(n-1)
```

show Unittests

(Re) Run the cell containing your function before launching unit tests

✔ Execute tests

Congratulations! ✔

hide Solution

[ ]:
```
# formation_solution


def fact(n):
    """return fact(n), computed recursivly"""
    if not isinstance(n, int):
        raise TypeError("n is not an integer")
    elif int(n) < 0:
        return 1
    elif n == 0:
```

Thanks to OpenDreamKit:

- A **new way to conduct training sessions**
  - each trainee can go at his own pace
    - focusing on the parts he wants to work on

- A **better learning experience**
  - interactive, integrated environment
  - accessible through a simple Web browser

- A **free sofware component**
  - that can be used to learn any language available in Jupyter
  - that gathers exercises defined in source repositories

- A **differentiating factor for Logilab**
  - trainees love it and continue to use it after the training
    - we have to cut off access 1 month after the session