



**GHENT
UNIVERSITY**

PYTHON FOR DUMMIES

SSHT invites – 26th januari

Computers should...

- improve efficiency,
- reduce human error,
- automate the mundane,
- simplify the complex,
- and accelerate research.

But scientists aren't trained to use them effectively.



DISCLAIMER

- After this session you will not be able to program in Python. The only goal is to get you started as researcher in using Python
- This session does not make a difference between researchers with programming experience and researchers with no programming experience

CONTENT

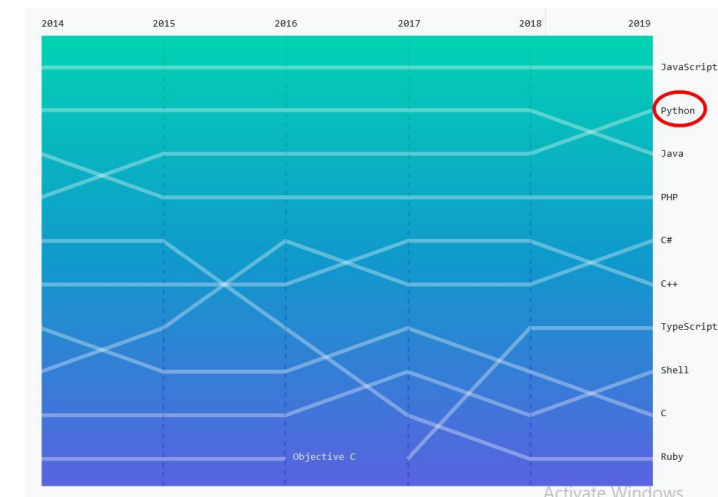
- Why Python?
- How to learn Python ?
- How to install Python?
- Python examples:
 - Using Ipython
 - Using PyCharm
 - Using Jupyter Notebooks

WHY PYTHON (IN GENERAL)?

- Easier than other programming languages
- Open source
- Highly productive, elegant, simple, yet powerful
- Vitamines included
- Rankings:
 - PopularitY of Programming Language Index: 1st
 - GitHub repositories contributions: 2nd

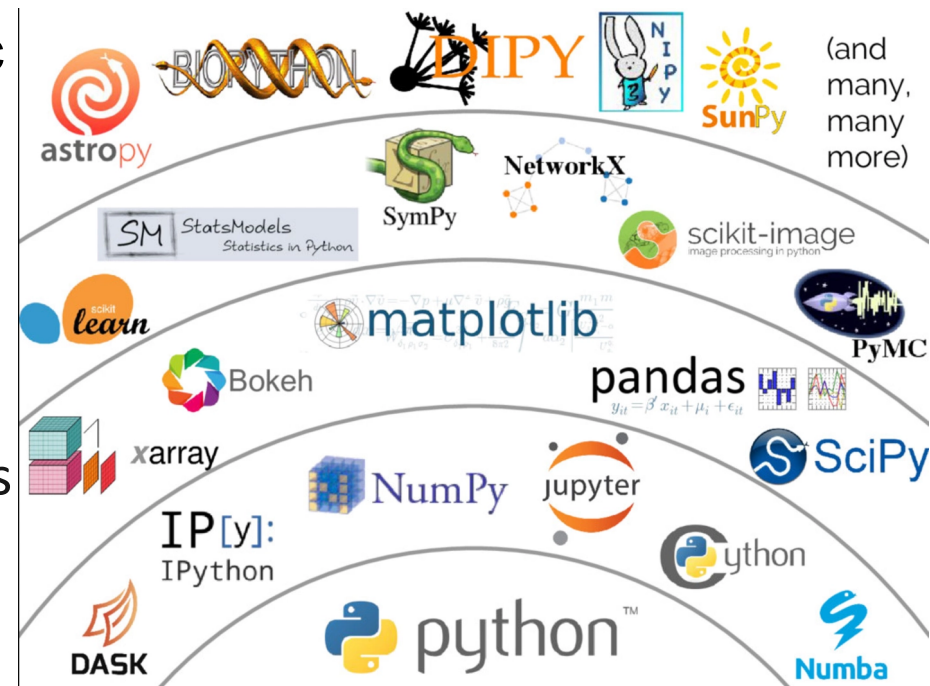
Worldwide, Jan 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	30.44 %	+1.2 %
2		Java	16.76 %	-2.0 %
3		JavaScript	8.44 %	+0.3 %
4		C#	6.53 %	-0.7 %
5	↑	C/C++	6.33 %	+0.3 %
6	↓	PHP	6.05 %	-0.2 %
7		R	3.87 %	+0.1 %
8		Objective-C	3.71 %	+1.2 %
9		Swift	2.14 %	-0.3 %
10		TypeScript	1.78 %	-0.0 %
11		Matlab	1.74 %	-0.1 %



WHY PYTHON (FOR RESEARCH)?

- Python has built-in support for scientific computing
 - SciPy
 - NumPy
 - Matplotlib
- Python has bridges to MATLAB
- Python has very good input/output options
- Python has strong support for task automation






HOW TO LEARN PYTHON

- UGent resources:
 - University wide Course Programming:
<https://studiegids.ugent.be/2020/EN/studiefiches/C003080.pdf>
 - Dodona platform: <https://dodona.ugent.be>
 - Online Dodona Course: De Programmeursleerling
- Other resources:
 - Datacamp
 - MOOCs:
 - Harvard course on EDx: <https://www.edx.org/course/using-python-for-research>
 - MIT cours on Edx: <https://www.edx.org/course/introduction-to-computer-science-and-programming-7>
 - Pearson Textbook: The practice of computing using Python
 - Online Textbooks
 - Podcasts




DEMONSTRATION DODONA

☰ Dodona 🎓 Frederik ▾ 🌐 en ▾

Algoritmen en Datast...   
2020–2021
Frederik Gailly · Universiteit Gent

44 445 Submissions 248 Users

📖 [GO TO THIS COURSE](#)

Objectgeoriënteerd P...   
2020–2021
Frederik Gailly · Universiteit Gent

67 638 Submissions 259 Users

📖 [GO TO THIS COURSE](#)

Frederik Gailly



511 Submissions 47 Correct exercises

RECENT EXERCISES

- ✓ **Parental control deel 2: SportActiviteit klasse** >
Objectgeoriënteerd Programmeren - 2de bach HIR
- ✗ **Parental control deel 1: Activiteit klasse** >
Objectgeoriënteerd Programmeren - 2de bach HIR
- ✓ **Woorden Trainer** >
Objectgeoriënteerd Programmeren - 2de bach HIR
- ✓ **Ontwikkelingsindex** >
Objectgeoriënteerd Programmeren - 2de bach HIR
- ✓ **BMR** >
Objectgeoriënteerd Programmeren - 2de bach HIR




[MORE COURSES ...](#)

2019–2020

Inleiding tot Programme...  
2019–2020
Frederik Gailly · Universiteit Gent

355 Submissions 20 Users

📖 [GO TO THIS COURSE](#)

Test Python   
2019–2020
Universiteit Gent

0 Submissions 1 Users

📖 [GO TO THIS COURSE](#)

DEMONSTRATION DATACAMP

The screenshot displays the DataCamp interface for an exercise. On the left, the exercise title is "The Python Interface". Below the title, there is a paragraph explaining that Python code is typed into a script (script.py) and executed in the IPython Shell. Further instructions mention hitting "Run Code" and "Submit Answer" to check submissions. Below this, there is a section titled "Instructions" worth 100 XP, which lists three steps: 1. Experiment in the IPython Shell; type `5 / 8`, for example. 2. Add another line of code to the Python script on the top-right (not in the Shell): `print(7 + 10)`. 3. Hit *Submit Answer* to execute the Python script and receive feedback.

The right side of the interface shows a code editor for `script.py` with the following content:

```
1 # Example, do not modify!
2 print(5 / 8)
3
4 # Put code below here
5 print(7+10)
```

Below the code editor are three buttons: a refresh button, a "Run Code" button, and a "Submit Answer" button. Below the code editor is an IPython Shell with a "Slides" tab and a dropdown arrow. The shell shows "In [1]:" followed by a blank line.

INSTALLING A PYTHON ENVIRONMENT

- <https://www.python.org/downloads/>
- Python interpreter: iPython
- Python editors:
 - Spyder
 - PyCharm (Academic Version)
 - Visual Studio Code
- Anaconda:
 - Distribution Python and R for scientific computing
 - Simplifies package management and deployment (conda)
 - Includes Anaconda Navigator
 - Allows users to launch applications and manage conda packages
 - Default applications: JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glue, Orange, RStudio and Visual Code studio

PYTHON EXAMPLE USING IPYTHON

```
fgaily — IPython: Users/fgaily — ipython — 80x24
[(base) fgaily@MacBook-Pro-F-Gaily ~ % ipython
Python 3.8.5 (default, Sep  4 2020, 02:22:02)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.19.0 -- An enhanced Interactive Python. Type '?' for help.

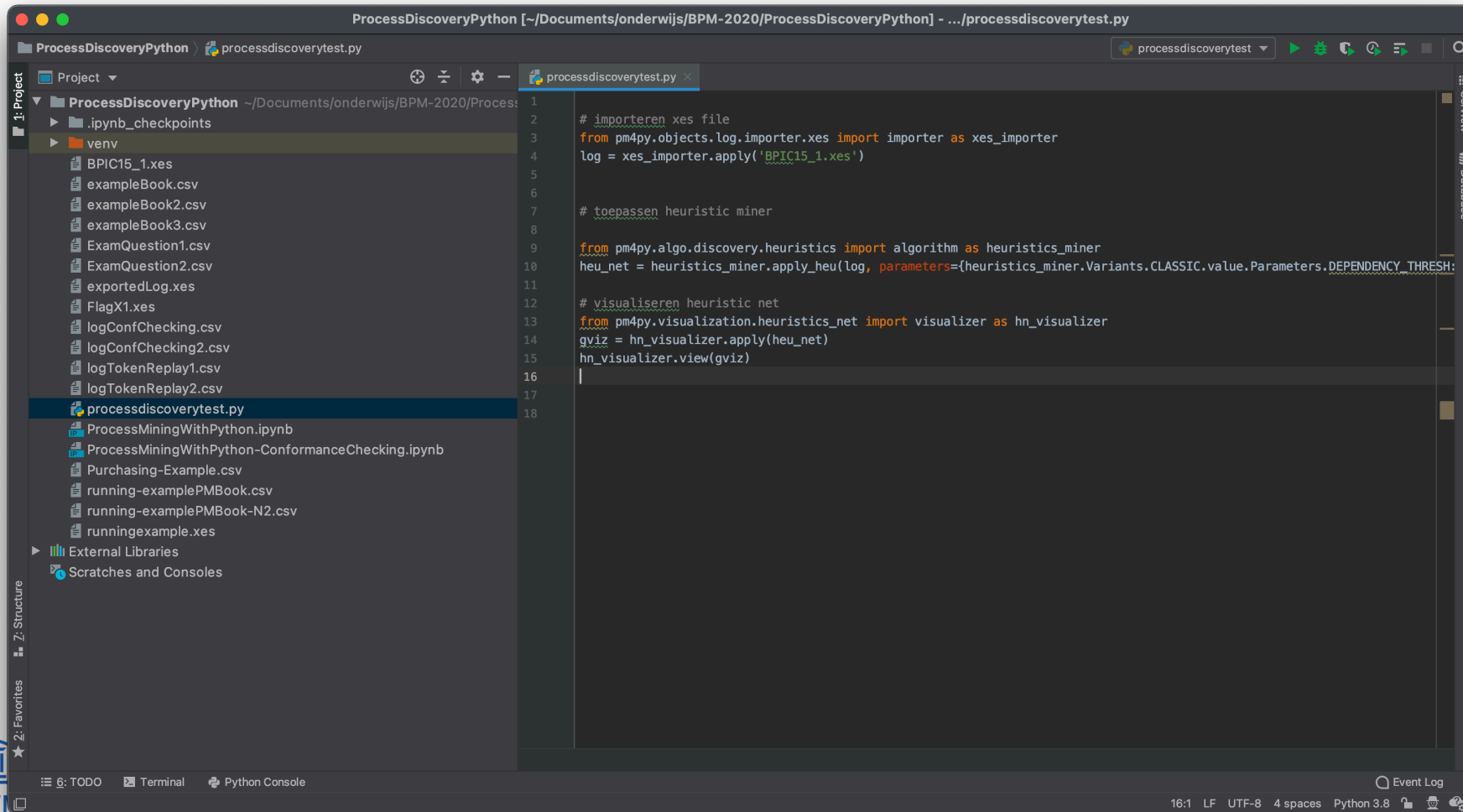
[In [1]: import numpy as np

[In [2]: i = np.random.random((2,2))

[In [3]: i
Out[3]:
array([[0.74915339, 0.37157556],
       [0.53335703, 0.46988596]])

In [4]: █
```

PYTHON EXAMPLE USING PYCHARM



The screenshot displays the PyCharm IDE interface. The top toolbar includes icons for running, debugging, and other IDE functions. The left sidebar shows a project tree for 'ProcessDiscoveryPython' with a 'venv' folder and various files like 'BPIC15_1.xes' and 'exampleBook.csv'. The main editor window shows the code for 'processdiscoverytest.py'.

```
1  
2 # importeren xes file  
3 from pm4py.objects.log.importer.xes import importer as xes_importer  
4 log = xes_importer.apply('BPIC15_1.xes')  
5  
6  
7 # toepassen heuristic miner  
8  
9 from pm4py.algo.discovery.heuristics import algorithm as heuristics_miner  
10 heu_net = heuristics_miner.apply_heu(log, parameters={heuristics_miner.Variants.CLASSIC.value.Parameters.DEPENDENCY_THRESH:  
11  
12 # visualiseren heuristic net  
13 from pm4py.visualization.heuristics_net import visualizer as hn_visualizer  
14 gviz = hn_visualizer.apply(heu_net)  
15 hn_visualizer.view(gviz)  
16  
17  
18
```

The bottom status bar shows '16:1 LF UTF-8 4 spaces Python 3.8' and an 'Event Log' icon.

JUPYTER NOTEBOOKS

- Free, open-source, interactive web tool
- Computational notebook
- Combines in one document
 - Software code
 - Computational output
 - Explanatory tekst
 - Multimedia resources
- Support different languages: JUlia, PYThon, R
- Vision: Move computer to the data instead of data to the computer

PYTHON EXAMPLE USING JUPYTER NOTEBOOKS

Jupyter ProcessMiningWithPython Laatste checkpoint: 04-06-2020 (automatisch opgeslagen) ✓



Logout

Bestand Bewerken Bekijken Invoegen Cel Kernel Widgets Help

Vertrouwd

Python 3

Uitvoeren

Markdown

Process Discovery with pm4py

Import Example book

```
In [13]: 1 import pandas as pd
2 from pm4py.objects.log.util import dataframe_utils
3 from pm4py.objects.conversion.log import converter as log_converter
4
5 from pm4py.objects.log.util import sorting
6
7 log_dataframe = pd.read_csv("examquestion2.csv", sep=';')
8 log_dataframe = dataframe_utils.convert_timestamp_columns_in_df(log_dataframe)
9
10
11
12
13 log_dataframe = log_dataframe.rename(columns={"caseID": "case:concept:name",
14                                             "timestamp": "time:timestamp",
15                                             "activiteitShort": "concept:name"})
16 print(log_dataframe)
17
18 log_dataframe = log_dataframe.sort_values("time:timestamp")
19 event_log = log_converter.apply(log_dataframe)
20 print(event_log)
```

	case:concept:name	time:timestamp	concept:name
0	1	30-12-2020:11.02	a
1	1	30-12-2020:11.03	c
2	1	30-12-2020:11.04	d
3	2	30-12-2020:11.02	b

QUESTIONS ???

RECIPE FOR PROGRAMMING SUCCESS

BY PETER NORVIG (DIRECTOR OF RESEARCH AT GOOGLE)

- Get interested in programming
- Program. The best kind of learning is learning by doing
- Talk with other programmers, read other programs
- Optional: Go to College
- Work on projects with other programmers
- Work on projects after other programmers
- Learn at least a half dozen programming languages
- Remember that there is a “computer” in computer science
- Get involved in language standardization
- Have the good sense to get off the language standardization effort as quickly as possible.

NEXT STEPS

- Version control using git/github
- Machine learning
- Advanced visualisations

Frederik Gailly

BUSINESS INFORMATICS
RESEARCH GROUP

E Frederik.Gaillys@ugent.be

www.ugent.be

 Ghent University
 @ugent
 Ghent University