

# Plan de cours

---

Programmation  
Examens  
Ressources

# Optimisation et IoT

---

Optimiser les réseaux d'objets connectés pour ...

- Minimiser la consommation d'énergie,
- Maximiser la couverture du réseau et répartir les ressources de façon équitable

# Optimisation et IoT - Exemple

---

[Hal open science](#) ; « Placement des données de l'internet des objets dans une infrastructure de fog. Mohammed Islam Naas »

# Optimisation et IoT - Exemple

---

Une ville souhaite installer un réseau de capteurs de qualité de l'air dans une ville

# Optimisation et IoT - Exemple

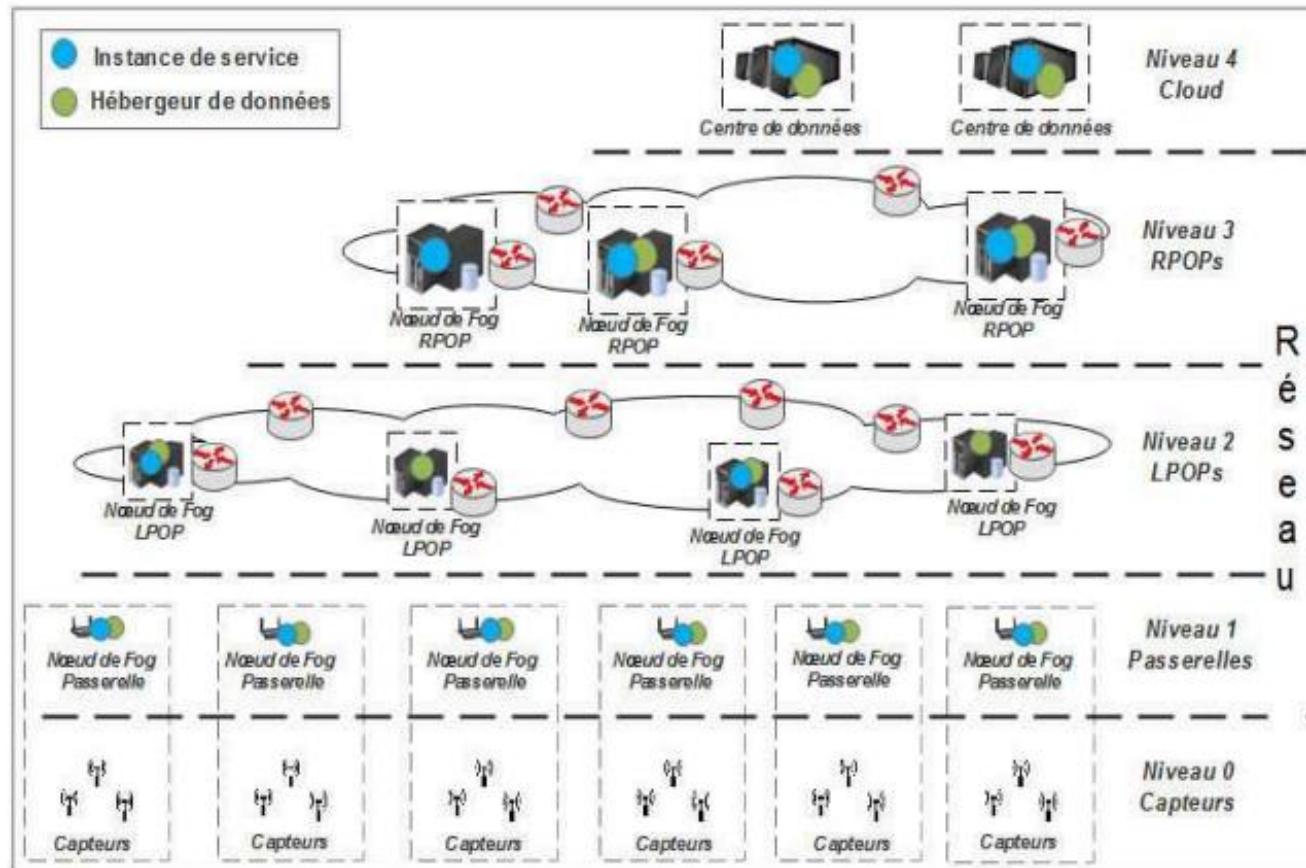


FIGURE 4 : Infrastructure du système d'un scénario d'une ville intelligente déployée dans le Fog.

# Optimisation et IoT - Exemple

---

Chaque capteur est un objet connecté qui mesure la qualité de l'air à un emplacement spécifique du réseau avec ...

# Optimisation et IoT - Exemple

---

- Une certaine capacité de mesure,
- Une consommation d'énergie,
- Et une couverture spatiale, limitée.

# Optimisation et IoT - Exemple

---

- Une certaine capacité de mesure,
- Une consommation d'énergie,
- Et une couverture spatiale, limitée.



# Optimisation et IoT - Exemple

---

$$\left\{ \begin{array}{ll} \text{Minimiser} & \sum_{i \in [1..l]} \sum_{j \in [1..n]} \alpha_{i,j} \cdot x_{i,j} \\ \text{Sous contraintes.} & \\ & \sum_{i \in [1..l]} s_{d_i} \cdot x_{i,j} \leq f_{dh_j} \quad \forall j \in [1..n] \\ & \sum_{j \in [1..n]} x_{i,j} = 1 \quad \forall i \in [1..l] \\ & x_{i,j} \in \{0, 1\} \quad \forall i \in [1..l], \forall j \in [1..n] \end{array} \right.$$

# Optimisation et IoT - Exemple

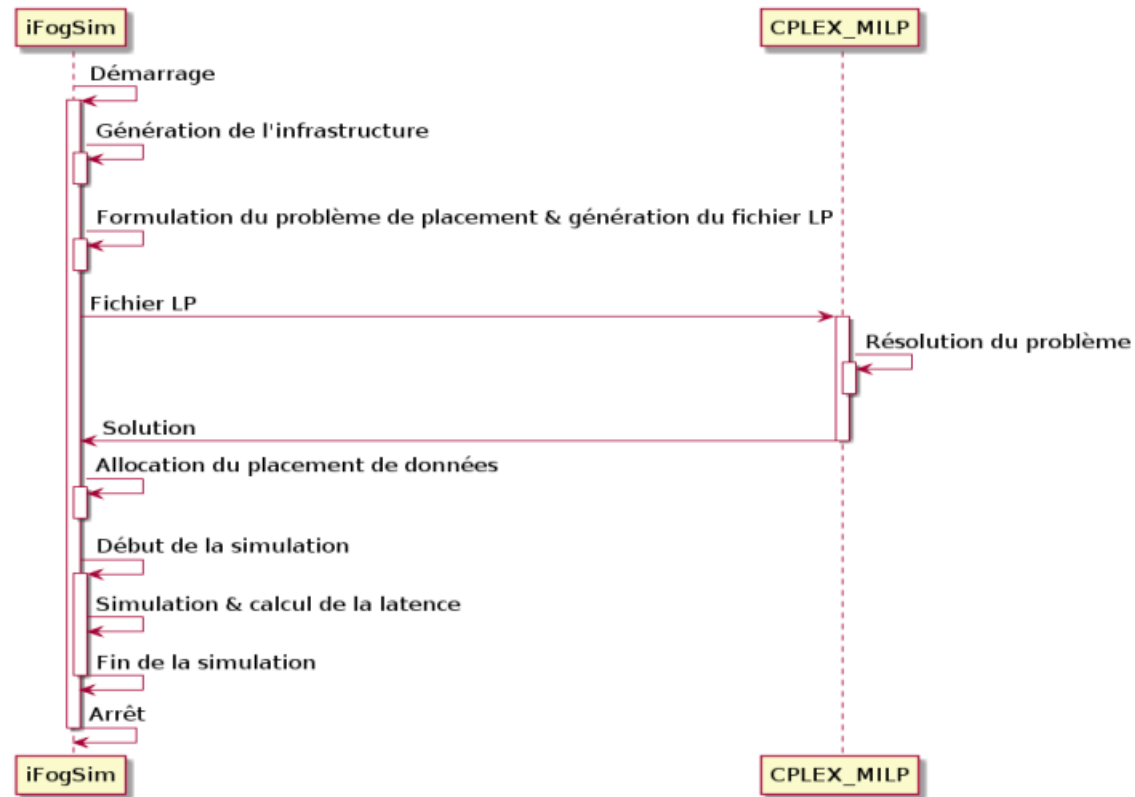


FIGURE 7 : Diagramme de séquence entre *iFogSim* et *CPLEX*.

# Optimisation et IoT - Exemple

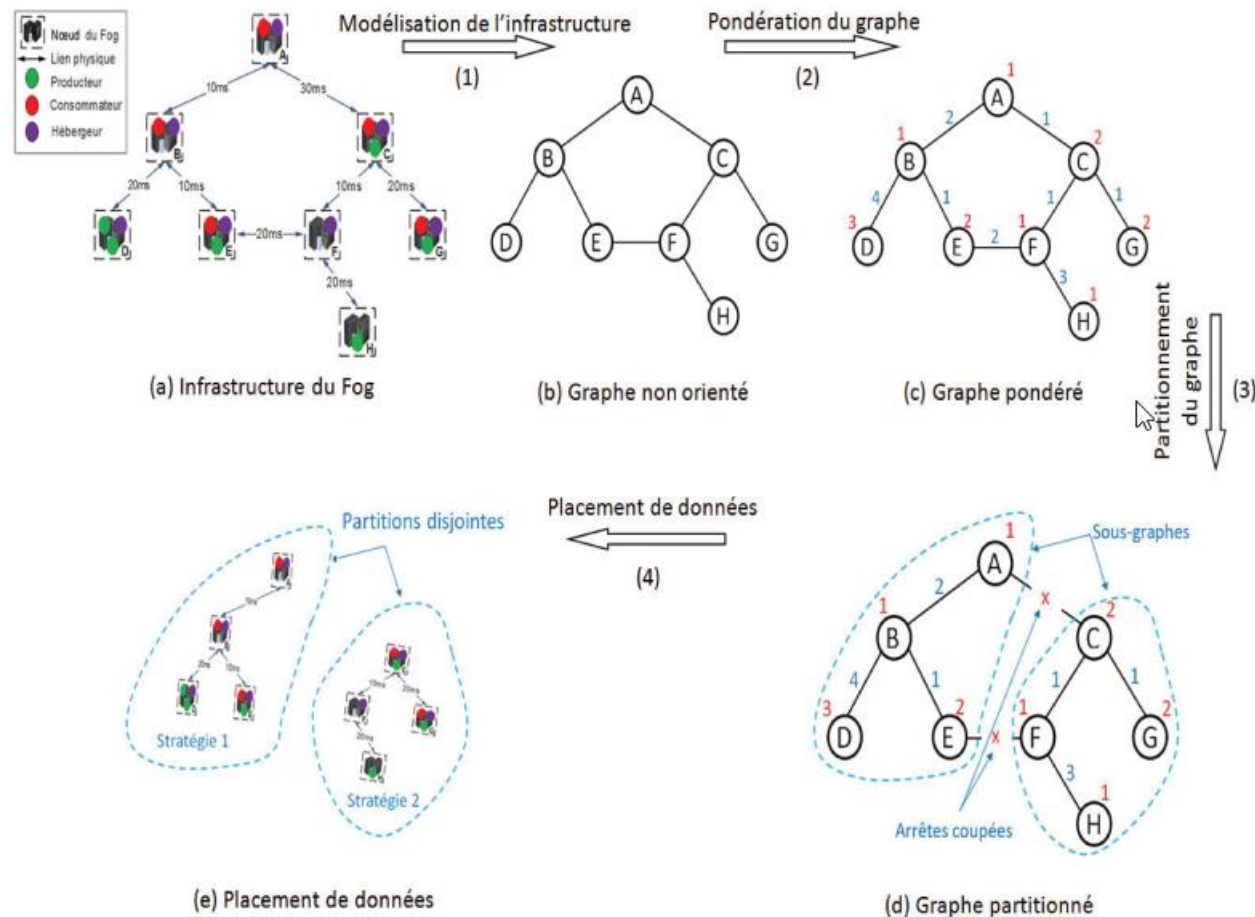


FIGURE 11 : Les différentes étapes de l'heuristique *iFogStorG*.

# Optimisation et IoT - Exemple

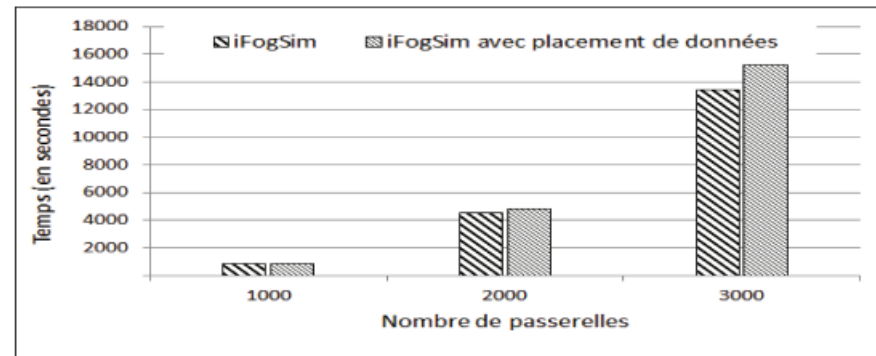


FIGURE 21 : Temps de simulation.

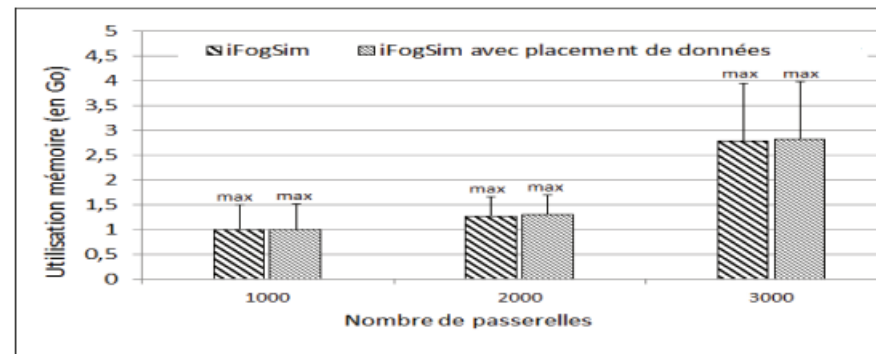


FIGURE 22 : Empreinte mémoire.



# Cours 01

---

**Mettez en place votre  
environnement Python**



# Table des matières

---



Partie 1 - Gérez des paquets Python à l'aide de pip



Partie 2 - Créez un environnement virtuel Python avec  
virtualenv



Partie 3 - Configurez un projet dans un IDE à l'aide de  
PyCharm





# Modules et paquets

---

Un module est un fichier python (\*.py), contenant du code

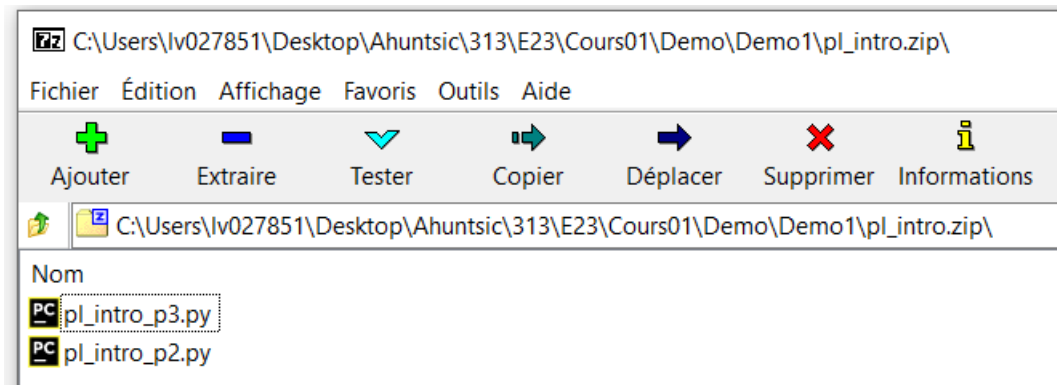
« exécutable » (python <module>) ou  
« importable » (import <module>)

Un paquet regroupe plusieurs modules  
i.e des fichiers « importables » (import  
<paquet>) ou « installables » (pip  
<paquet>)

---



# Modules et paquets



```
C:\Windows\System32\cmd.exe
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo1>python pl_intro_p2.py
File "C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo1\pl_intro_p2.py", line 93
    print "Solution:"
      ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Solution:")?

C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo1>python --version
Python 3.9.1

C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo1>python pl_intro_p3.py
Solution:
x1 = 2.0
x2 = 3.0

C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo1>
```





# import <module> | <paquet>

---

- Importer un module Python  
`import <module>`
- Importer un paquet Python  
`import <package>`
- Importer une fonctionnalité d'un paquet/module  
`from <module|package> import <function|class>`
- Importer tout le contenu d'un paquet/module  
`from <module|package> import *`
- Création d'alias de module/paquet  
`import <module|package> as <alias>`



# import <module> | <paquet>

---

```
pl_intro_p2.py - Bloc-notes
Fichier Edition Format Affichage Aide
# -*- coding: latin-1 -*-
from __future__ import division

class SimplexSolver:
    def __init__(self, objective_coeffs, constraints_coeffs,
constraints_rhs):
        self.objective_coeffs = objective_coeffs
        self.constraints_coeffs = constraints_coeffs
        self.constraints_rhs = constraints_rhs
        self.num_constraints = len(constraints_coeffs)
        self.num_variables = len(objective_coeffs)
```



# PyPi et pip

- Les paquets Python servent à stocker du code effectuant des tâches courantes, afin de pouvoir le réutiliser facilement.
- Les paquets peuvent rester privés ou être mis à disposition du public via PyPi.
- L'installation et la désinstallation des paquets Python sont contrôlées via `pip`.

The screenshot shows the PyPi project page for the 'future' package. At the top, there's a search bar and navigation links. The package name 'future 0.18.3' is prominently displayed, along with a 'pip install future' button and a 'Dernière version' badge. Below this, the package description is shown: 'Clean single-source support for Python 3 and 2'. A sidebar on the left contains navigation links like 'Description du projet', 'Historique des versions', and 'Téléchargement des fichiers'. The main content area provides a detailed description of the package as a compatibility layer between Python 2 and 3, followed by a code snippet showing how to use the package.

Rechercher des projets

Aide Sponsors Se connecter S'inscrire

## future 0.18.3

✓ Dernière version

`pip install future`

Dernière version : 12 janv. 2023

Clean single-source support for Python 3 and 2

### Navigation

- Description du projet
- Historique des versions
- Téléchargement des fichiers

### Liens du projet

- Homepage

### Statistiques

### Description du projet

`future` is the missing compatibility layer between Python 2 and Python 3. It allows you to use a single, clean Python 3.x-compatible codebase to support both Python 2 and Python 3 with minimal overhead.

It is designed to be used as follows:

```
from __future__ import (absolute_import, division,
                        print_function, unicode_literals)

from builtins import (
    bytes, dict, int, list, object, range, str,
    ascii, chr, hex, input, next, oct, open,
    pow, round, super,
    filter, map, zip)
```



# pip install | pip freeze

---

- `!pip install <package>!` permet d'installer des paquets Python à partir de la ligne de commande.
- Une fois que vous avez installé un paquet Python, vous pouvez utiliser la commande `!import!` pour l'importer dans un script Python et utiliser ses fonctionnalités.
- `!pip!` dispose de différentes commandes qui vous permettent de voir les paquets actuellement installés.



# Versions de paquets

---

- Les paquets Python sont versionnés pour faciliter le suivi des versions de chaque code.
- Les paquets Python reçoivent un nouveau numéro de version chaque fois qu'une modification leur est apportée.
- Vous pouvez utiliser `!pip!` pour installer la version souhaitée d'un paquet Python.



# Pourquoi un environnement virtuel?

Isoler la configuration d'un projet utilisant

- Des versions différentes de Python
- Et/ou des versions différentes de paquets



# Gérer un environnement virtuel

---

Il y a plusieurs outils permettant de créer/utiliser un environnement virtuel;

1. Le paquet **venv** (python -m venv) présenté dans la [seconde section du cours OC](#)
2. Le paquet **virtualenv** (démon prof)
3. Un « ide » comme **Pycharm** (exercice01)



# Demo Astar virtualenv

---


- [Télécharger une version portable de Python2](#)
- Installer virtualenv avec python2.7.exe  
python -m pip install virtualenv
- Créer un environnement virtuel
- Activer l'environnement virtuel
- Installer les paquets manquants
- Tester le code
- Désactiver l'environnement virtuel





# Version portable de Python 2

Home / Browse Open Source / Software Development / Portable Python / Files



## Portable Python Files

Brought to you by: [thewatcher2018](#)


Summary

Files


Reviews

Support










Tickets

 **Download Latest Version**  
Portable Python-3.10.5 x64.exe (40.8 MB)

Get Updates



[Home](#) / [Portable Python 2.7](#)

Name	Modified	Size	Downloads / Week
 <a href="#">Parent folder</a>			
<a href="#">Readme.txt</a>	2019-11-06	770 Bytes	 
<a href="#">Portable Python-2.7.17 x64.exe</a>	2019-11-06	30.0 MB	39  
<a href="#">Portable Python-2.7.17.exe</a>	2019-11-06	28.7 MB	16  
<a href="#">Portable Python-2.7.16 x64.exe</a>	2019-05-22	29.6 MB	1  



# Vérifier les versions de python et virtualenv

---

C:\Windows\System32\cmd.exe

Microsoft Windows [version 10.0.19041.1415]  
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2>python --version  
Python 3.9.1

C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2>virtualenv --version  
virtualenv 20.14.1 from c:\users\lv027851\appdata\local\programs\python\python39\lib\site-packages\virtualenv\\_\_init\_\_.py



# Installer virtualenv (Python2)

---

```
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Partage\Demo2\Portable Python-2.7.17 x64\AppData\Python>python -m pip install virtualenv
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't be maintained after that date. A future version of pip will drop support for Python 2.7. More details about Python 2 support in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support
Collecting virtualenv
  Downloading https://files.pythonhosted.org/packages/6f/43/df7c7b1b7a5ac4e41fac24c3682c1cc32f2c1d683d308bba2500338d1e3e/virtualenv-20.15.1-py2.py3-none-any.whl (10.1MB)
    ##### | 10.1MB 3.4MB/s
Collecting importlib-metadata>=0.12; python_version < "3.8" (from virtualenv)
```

C:\Windows\System32\cmd.exe

```
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\Portable Python-2.7.17 x64\AppData\Python>pip freeze
absl-py==0.14.0
altgraph==0.17
anyio==3.5.0
appdirs==1.4.4
argon2-cffi==21.1.0
asgiref==3.4.1
```

```
ctqum==4.65.0
traitlets==5.0.5
typing-extensions==4.6.2
urllib3==1.25.11
virtualenv==20.14.1
wcwidth==0.2.5
webencodings==0.5.1
```



# Créer un env. virtuel (Python2)

---

```
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\Portable Python-2.7.17 x64\AppData\Python>Python -m virtualenv "C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\venvp2"
created virtual environment CPython2.7.17.final.0-64 in 908ms
  creator CPython2Windows(dest=C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\venvp2, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, wheel=bundle, setuptools=bundle, via=copy, app_data_dir=C:\Users\lv027851\AppData\Local\pypa\virtualenv)
    added seed packages: pip==20.3.4, setuptools==44.1.1, wheel==0.37.1
  activators NushellActivator,PythonActivator,FishActivator,BatchActivator,PowerShellActivator,BashActivator
```



# Activer l'environnement virtuel

---

```
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Partage\Demo2\venv2>cd ..  
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Partage\Demo2>cd venv2  
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Partage\Demo2\venv2>cd Scripts  
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Partage\Demo2\venv2\Scripts>activate  
(venv2) C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Partage\Demo2\venv2\Scripts>pip2 --version  
pip 20.3.4 from C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Partage\Demo2\venv2\lib\site-packages\pip (python 2.7)  
(venv2) C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Partage\Demo2\venv2\Scripts>python --version  
Python 2.7.17  
(venv2) C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Partage\Demo2\venv2\Scripts>_
```



# Tester

---

```
C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19041.1415]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\venvp2\Scripts>activate

(venvp2) C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\venvp2\Scripts>cd ..

(venvp2) C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\venvp2>cd ..

(venvp2) C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2>python --version
Python 2.7.17

(venvp2) C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2>python pl_intro_p2.py
Solution:
x2 = 3.0
x1 = 2.0

(venvp2) C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2>python pl_intro_p3.py
Solution:
x2 = 3
x1 = 2

(venvp2) C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2>_
```



# Désactiver l'environnement virtuel

---

```
C:\Windows\System32\cmd.exe
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2>cd venvp2
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\venvp2>cd Scripts
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\venvp2\Scripts>deactivate.bat
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\venvp2\Scripts>cd ..
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\venvp2>cd ..
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2>python --version
Python 3.9.1
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2>
```




# Create New Project - Location


Create Project

Location: C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo3

▼ Python Interpreter: New Virtualenv environment

☒ New environment using  Virtualenv ▼

Location: C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo3\venvp2

Base interpreter:  C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\Portab ▼ ...

☐ Inherit global site-packages

☐ Make available to all projects

☐ Previously configured interpreter

Interpreter: <No interpreter> ▼ ...

Create





# Virtualenv Location

Create Project

Location: C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo3

Python Interpreter: New Virtualenv environment

☒ New environment using Virtualenv

Location: C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo3\venvp2

Base interpreter: C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\Portab ...

☐ Inherit global site-packages

☐ Make available to all projects

☐ Previously configured interpreter

Interpreter: <No interpreter> ...

Create




# Erreur si env. location empty


Create Project

Location: C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo3

▼ Python Interpreter: New Virtualenv environment

☒ New environment using  Virtualenv ▼

Location: C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo3

Base interpreter:  C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\Portab ...

☐ Inherit global site-packages

☐ Make available to all projects

☐ Previously configured interpreter

Interpreter: <No interpreter> ...

⚡ Environment location directory is not empty

Create



# Base interpreter

Enregistrement automatique Cours 01 - Mise en place de l'environnement Python.pptx • Enregistré dans ce PC Rechercher Vincent Leduc

Fichier Accueil Insertion Con

Annuler Presse-papiers Nouve diaposi

32 Create New Project - Location

33 Virtualenv Location

**Create Project**

Location: C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo3

▼ Python Interpreter: New Virtualenv environment

☒ New environment usi... Virtualenv

Location: C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo3\

Base interpreter: C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\Portab ...

C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo2\Portable Python-2.7.17 x64\AppData\Python\python.exe

- Python 3.9 (Brian) C:\Users\lv027851\AppData\Local\Programs\Python\Python39\python.exe
- Python 3.9 C:\Users\lv027851\AppData\Local\Programs\Python\Python39\python.exe
- [invalid] Python 3.9 (Labo) C:\Users\lv027851\Desktop\Ahuntsic\313\Cours10\Labo\env\Scripts\python.exe
- [invalid] Python 3.9 (V3) C:\Users\lv027851\Desktop\Ahuntsic\316\Projet\CSP\V3\env\Scripts\python.exe
- [invalid] Python 3.9 (Pokemon) C:\Users\lv027851\Desktop\Ahuntsic\AEC iot\IA vulgarisée\Pokemon\venv\Scripts\python.exe
- [invalid] Python 3.9 (Devoir) (2) C:\Users\lv027851\Desktop\Ahuntsic\316\Cours 02\Devoir\env\Scripts\python.exe
- [invalid] Python 3.9 (Depot) C:\Users\lv027851\Desktop\Ahuntsic\318\Cours 07\Depot\env\Scripts\python.exe



# Create From Existing SOurce

---



Directory Is Not Empty



The directory 'C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo3' is not empty. Do you want to create project from existing sources?

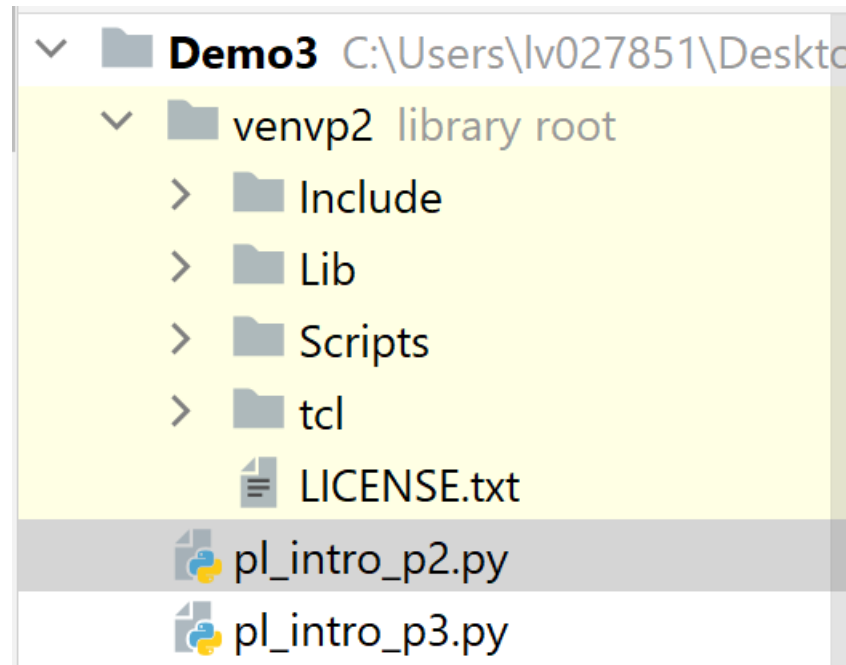
Create from Existing Sources

Open Project



# Vérifier l'environnement virtuel

---





# Tester le code

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The title bar indicates the current file is 'pl\_intro\_p2.py' in the 'Demo3' project.

The left sidebar shows the project structure:

- Project: Demo3 (C:\Users\lv027851\Desktop\...)
- venvp2 library root
  - Include
  - Lib
  - Scripts
  - tcl
  - LICENSE.txt
- pl\_intro\_p2.py
- pl\_intro\_p3.py

The main editor displays the code for 'pl\_intro\_p2.py':1 # -\*- coding: latin-1 -\*-
2 from \_\_future\_\_ import division
3
4 class SimplexSolver:
5 def \_\_init\_\_(self, objective\_coeffs, constraints\_coeffs, constraints\_rhs):
6 self.objective\_coeffs = objective\_coeffs
7 self.constraints\_coeffs = constraints\_coeffs
8 self.constraints\_rhs = constraints\_rhs
9 self.num\_constraints = len(constraints\_coeffs)
10 self.num\_variables = len(objective\_coeffs)
11

The bottom panel shows the 'Run' output for 'pl\_intro\_p2.py':

```
C:\Users\lv027851\Desktop\Ahuntsic\313\E23\Cours01\Demo\Demo3\venvp2\Scripts\python.exe
C:/Users/lv027851/Desktop/Ahuntsic/313/E23/Cours01/Demo/Demo3/pl_intro_p2.py
Solution:
x2 = 3.0
x1 = 2.0
Process finished with exit code 0
```

The bottom status bar shows: Packages installed successfully: Installed packages: 'future' (moments ago) 6:33 CRLF ISO-8859-1 4 spaces Python 2.7 (Demo3) (3) 1 Event Log



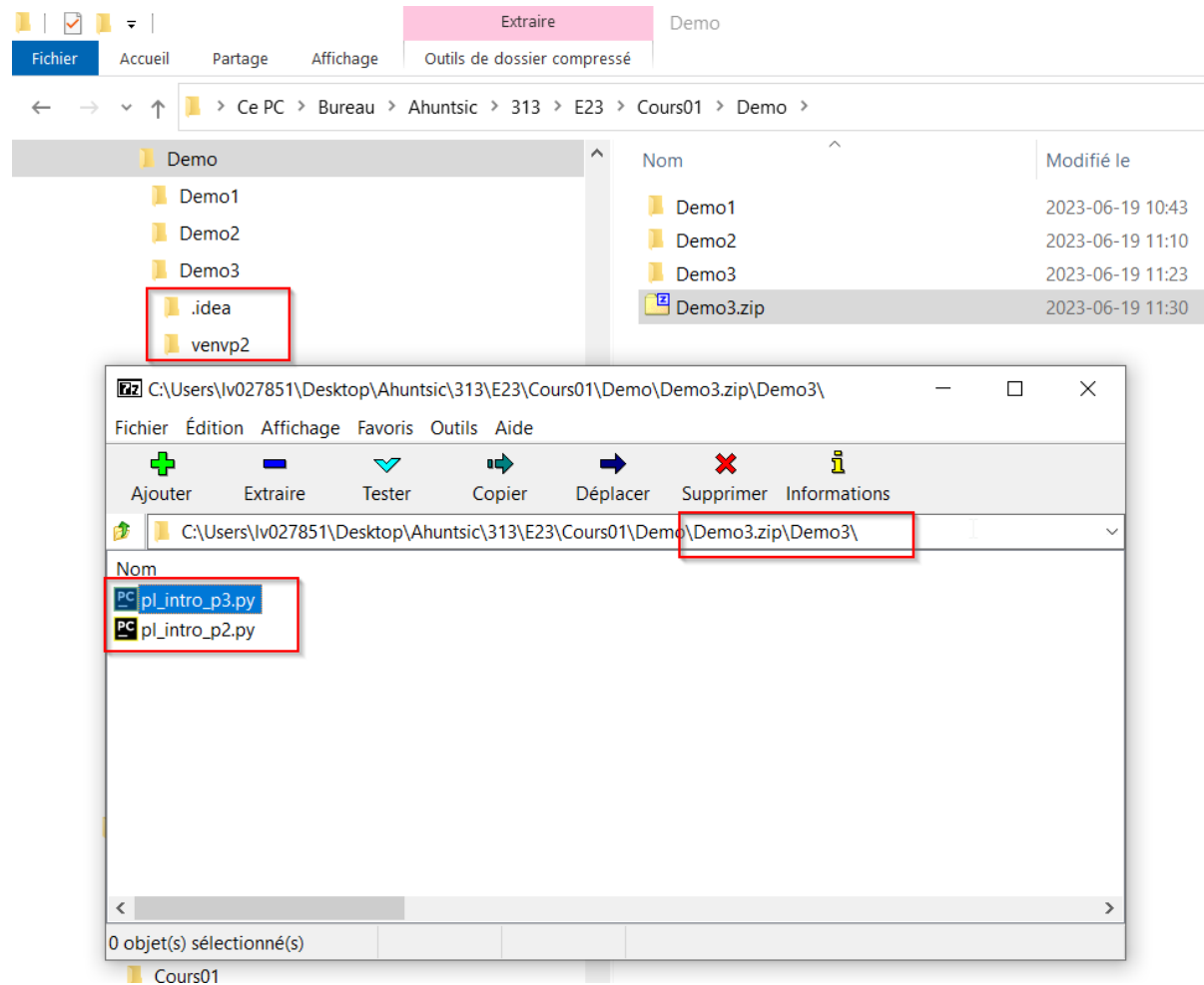
# Partager/déposer un projet PyCharm

---

1. « Zipper » le dossier du projet
2. Supprimer le code généré i.e. les 2 sous-dossiers suivants;
  - a) .idea
  - b) l'environnement virtuel (pas portable)
3. Partager/Déposer le fichier zip



# Zipper le projet







# Recréer un projet Pycharm

---

1. « Dézipper » le projet dans un autre dossier
2. Créer un nouveau projet Pycharm à partir du code source en recréant l'environnement virtuel
3. Tester le code