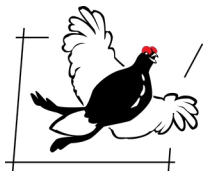




Bonnes pratiques pour le calcul scientifique et la collaboration avec Python sous Windows 10

▼ Webinaire du 22/06/2020



Tétras Libre



David Beniamine

Systèmes et performances

Docteur en Informatique



David Rouquet

Science et ingénierie des données

Docteur en Informatique



Sébastien Curt

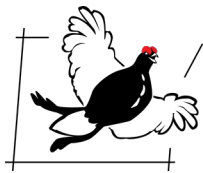
Systèmes et données

Master en Informatique



8 rue de Mayencin
38 400 Saint Martin d'Hères

Contact@tetras-libre.fr
David.Beniamine@tetras-libre.fr
David.Rouquet@tetras-libre.fr
Sebastien.Curt@tetras-libre.fr



Plan

1) Configurer son environnement Python

1) De Python à Anaconda

2) Utiliser les environnements virtuels

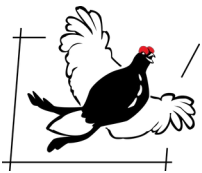
3) Exercices de prise en main

2) Développer en mode collaboratif

1) Du script partagé aux projets

2) Gestionnaire de version

3) Couche sociale



Python un langage de programmation



Python



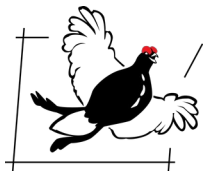
Interpréteur



Bibliothèques standards



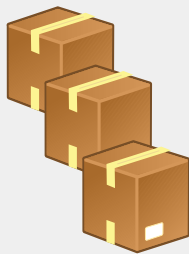
(Éditeur)



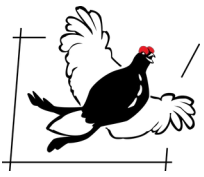
Mon pc

Systeme

Python

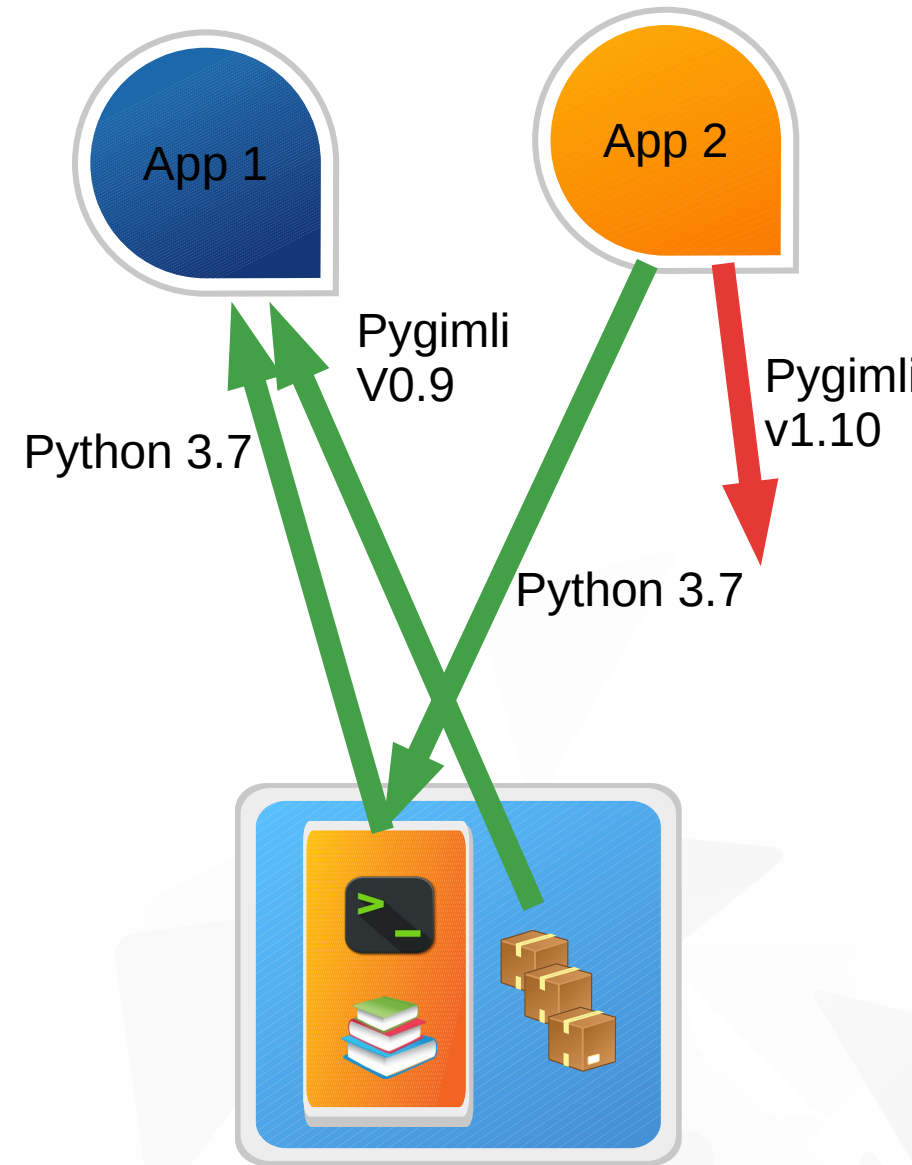


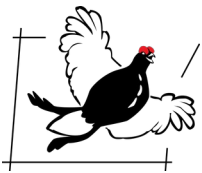
Paquets = bibliothèque non standard



Installation classique

- ▼ 1 version de python
- ▼ Paquets installés “à la main”
- ▼ Conflits de dépendances



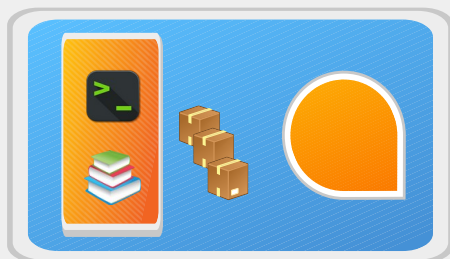


Environnements virtuels

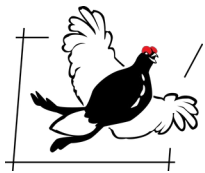
Environnement virtuels



Projet 1



Projet 2



Conda



Conda



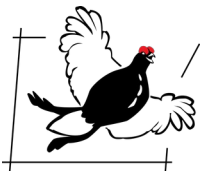
Gestionnaire d'environnements



Gestionnaire de paquet



Interface en ligne de commande



Interface en ligne de commande

▼ Création

```
(base)> conda env create toto
```

▼ Changement

```
(base)> conda env activate toto
```

▼ Installation

```
(toto)> conda install click
```

▼ Désinstallation

```
(toto)> conda uninstall click
```

▼ Liste des paquets

```
(toto)> conda list
```

▼ Liste des environments

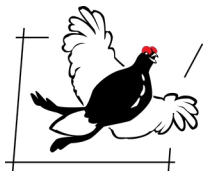
```
(base)> conda env list
```

▼ Export

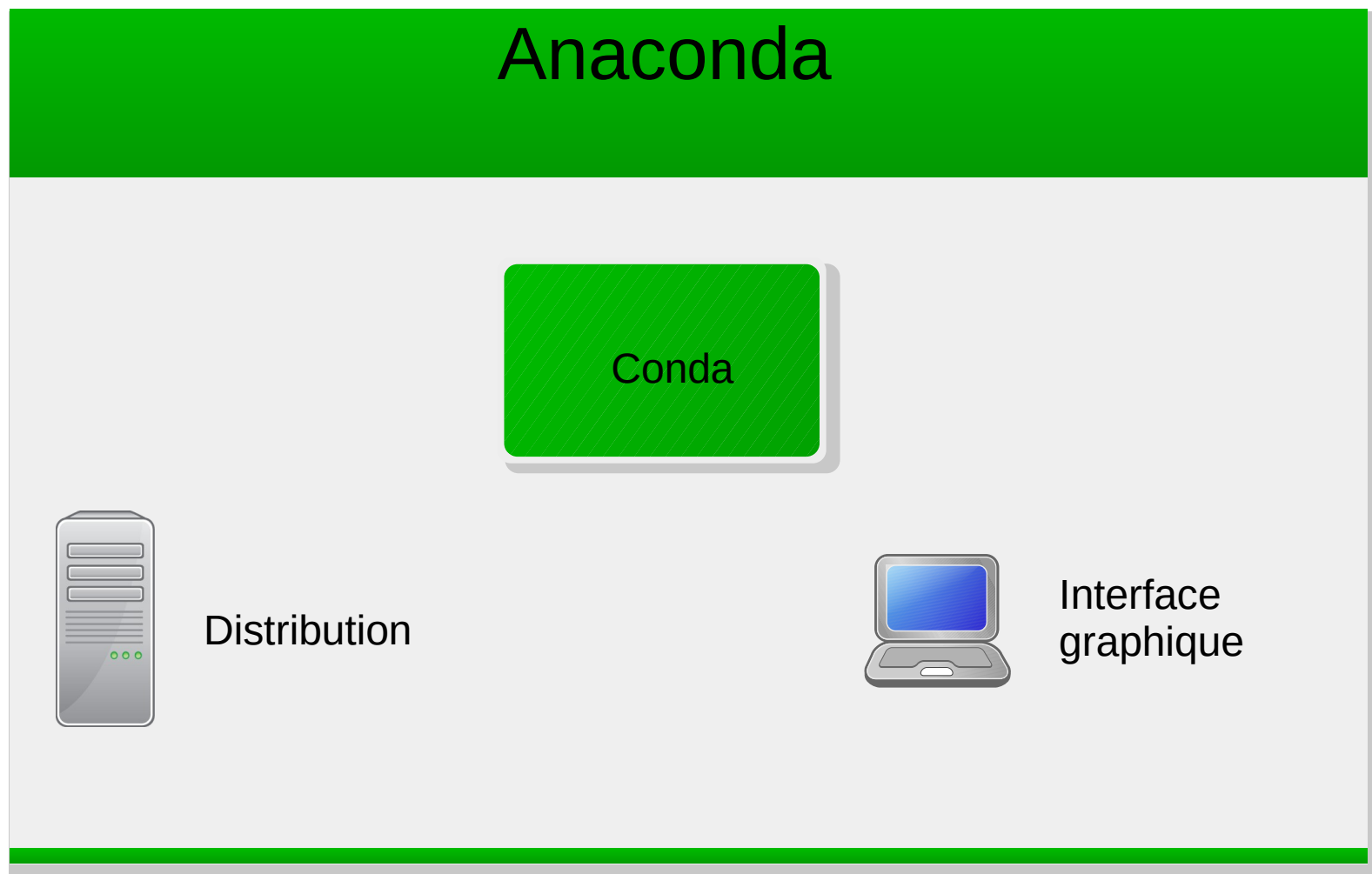
```
(toto)> conda env export >  
toto.yml
```

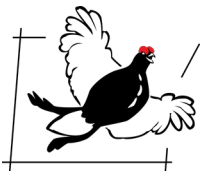
▼ Import

```
(toto)> conda create -f  
toto.yml
```



Anaconda





Plan

1) Configurer son environnement Python

1) De Python à Anaconda

2) Utiliser les environnements virtuels

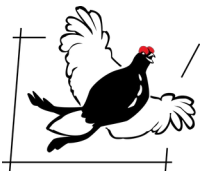
3) Exercices de prise en main

2) Développer en mode collaboratif

1) Du script partagé aux projets

2) Gestionnaire de version

3) Couche sociale




Anaconda Navigator





 ANACONDA NAVIGATOR

[Sign in to Anaconda Cloud](#)

 Home

 Environments

 Learning

 Community

[Documentation](#)

[Developer Blog](#)



Applications on pybert

Channels

[Refresh](#)



psyploit-gui
1.2.4

[Install](#)



pyboat
0.8.1.2

[Install](#)



Qt Console
4.7.4

PyQt GUI that supports inline figures,
proper multiline editing with syntax
highlighting, graphical calltips, and more.

[Install](#)



RStudio
1.1.456

A set of integrated tools designed to help
you be more productive with R. Includes R
essentials and notebooks.

[Install](#)



Spyder
4.1.3

Scientific PYTHON Development
EnviRONment. Powerful Python IDE with
advanced editing, interactive testing,
debugging and introspection features

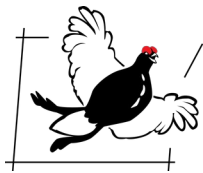
[Install](#)



stradtize
0.1.3

[Install](#)





Liste des environnements



[Home](#)
[Environments](#)
[Learning](#)
[Community](#)

[Documentation](#)
[Developer Blog](#)

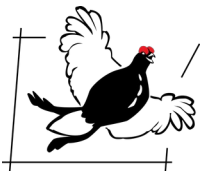
base (root) ▶

Installed Channels Update index... Search Packages

Name	T	Description	Version
✓ _ipyw_jlab_nb_ex...	🔄	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
✓ alabaster	🔄	Configurable, python 2+3 compatible sphinx theme.	0.7.12
✓ anaconda	🔄	Simplifies package management and deployment of anaconda	2020.02
✓ anaconda-client	🔄	Anaconda.org command line client library	1.7.2
✓ anaconda-project	🔄	Tool for encapsulating, running, and reproducing data science projects	0.8.4
✓ argh	🔄		0.26.2
✓ asn1crypto	🔄	Python asn.1 library with a focus on performance and a pythonic api	1.3.0
✓ astroid	🔄	A abstract syntax tree for python with inference support.	🔗 2.3.3
✓ astropy	🔄	Community-developed python library for astronomy	🔗 4.0
✓ atomicwrites	🔄	Atomic file writes.	🔗 1.3.0
✓ attrs	🔄	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	19.3.0
✓ autopep8	🔄	A tool that automatically formats python code to conform to the pep 8 style g...	🔗 1.4.4
✓ babel	🔄	Utilities to internationalize and localize python applications	2.8.0
✓ backcall	🔄	Specifications for callback functions passed in to an api	0.1.0
✓ backports	🔄		1.0
✓ backports.functoo...	🔄		1.6.1
✓ backports.functoo...	🔄	Backport of Functools.lru_cache from python 3.3 as published at activestate	1.6.1

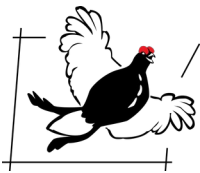
322 packages available

Create Clone Import Remove



4 bonnes pratiques

- 1) Toucher le moins possible à l'environnement “Base”
- 2) Utiliser au maximum (uniquement ?) Anaconda et conda
- 3) 1 projet = 1 environnement
- 4) Partager les environnements entre collaborateur·ice·s



IDE et environnements

▼ Spyder



- ▼ 1 Spyder par env.
- ▼ Redémarrer Spyder à chaque changement de projet
- ▼ Préférences Spyder spécifiques à chaque env.

▼ Pycharm

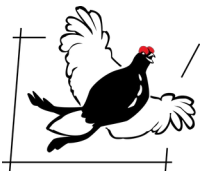


- ▼ Aucune configuration
- ▼ Possibilité de changer d'env.
- ▼ 1 installation dans “base”

▼ Jupyter lab



- ▼ Un peu de configuration
- ▼ Possibilité de changer d'env.
1 installation dans “base”



Plan

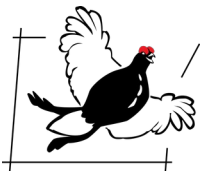
1) Configurer son environnement Python

- 1) De Python à Anaconda
- 2) Utiliser les environnements virtuels

3) Exercices de prise en main

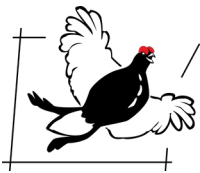
2) Développer en mode collaboratif

- 1) Du script partagé aux projets
- 2) Gestionnaire de version
- 3) Couche sociale



Mise en pratique

- ▼ Import des environnements pré configurés
- ▼ Configuration de git-bash
- ▼ Utilisation et changement d'environnement
- ▼ Utilisation des environnement dans Pycharm



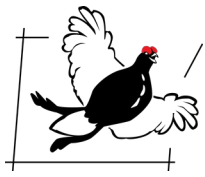
Plan

1) Configurer son environnement Python

- 1) De Python à Anaconda
- 2) Utiliser les environnements virtuels
- 3) Exercices de prise en main

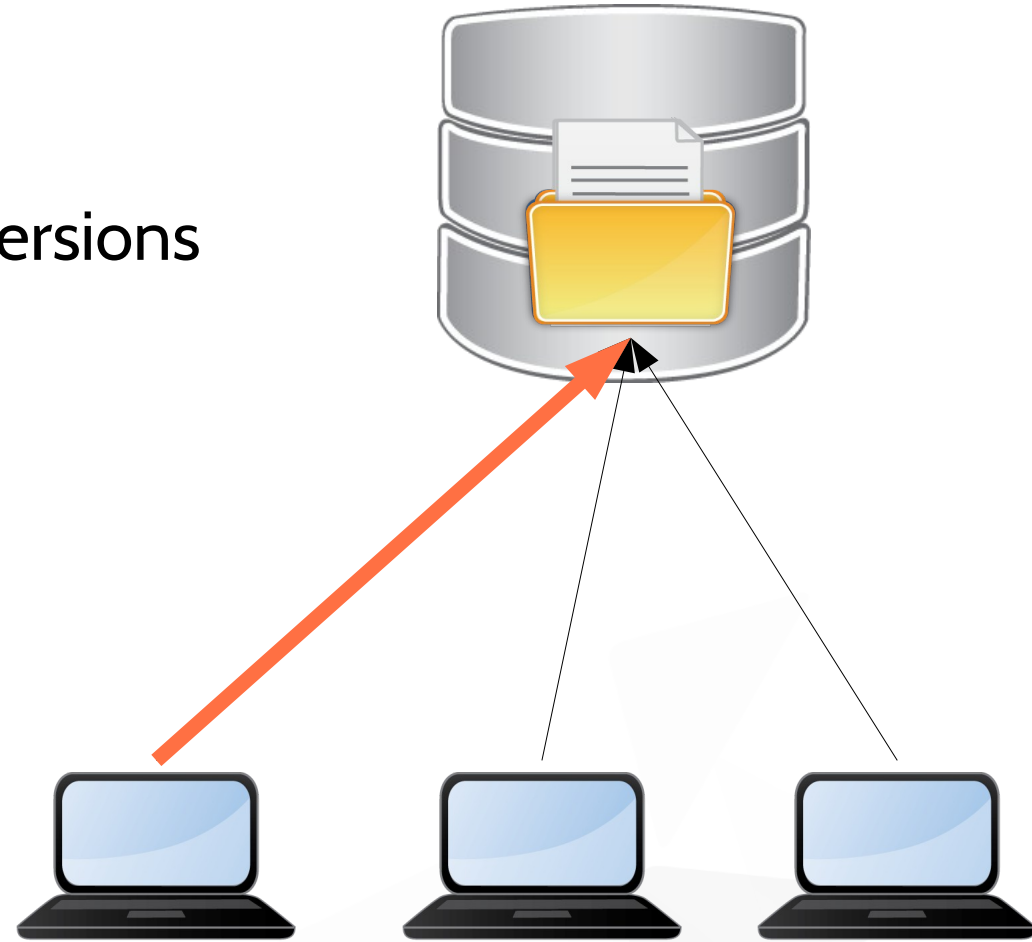
2) Développer en mode collaboratif

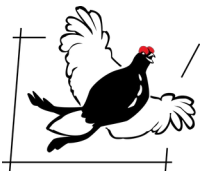
- 1) Du script partagé aux projets
- 2) Gestionnaire de version
- 3) Couche sociale



Scripts partagés

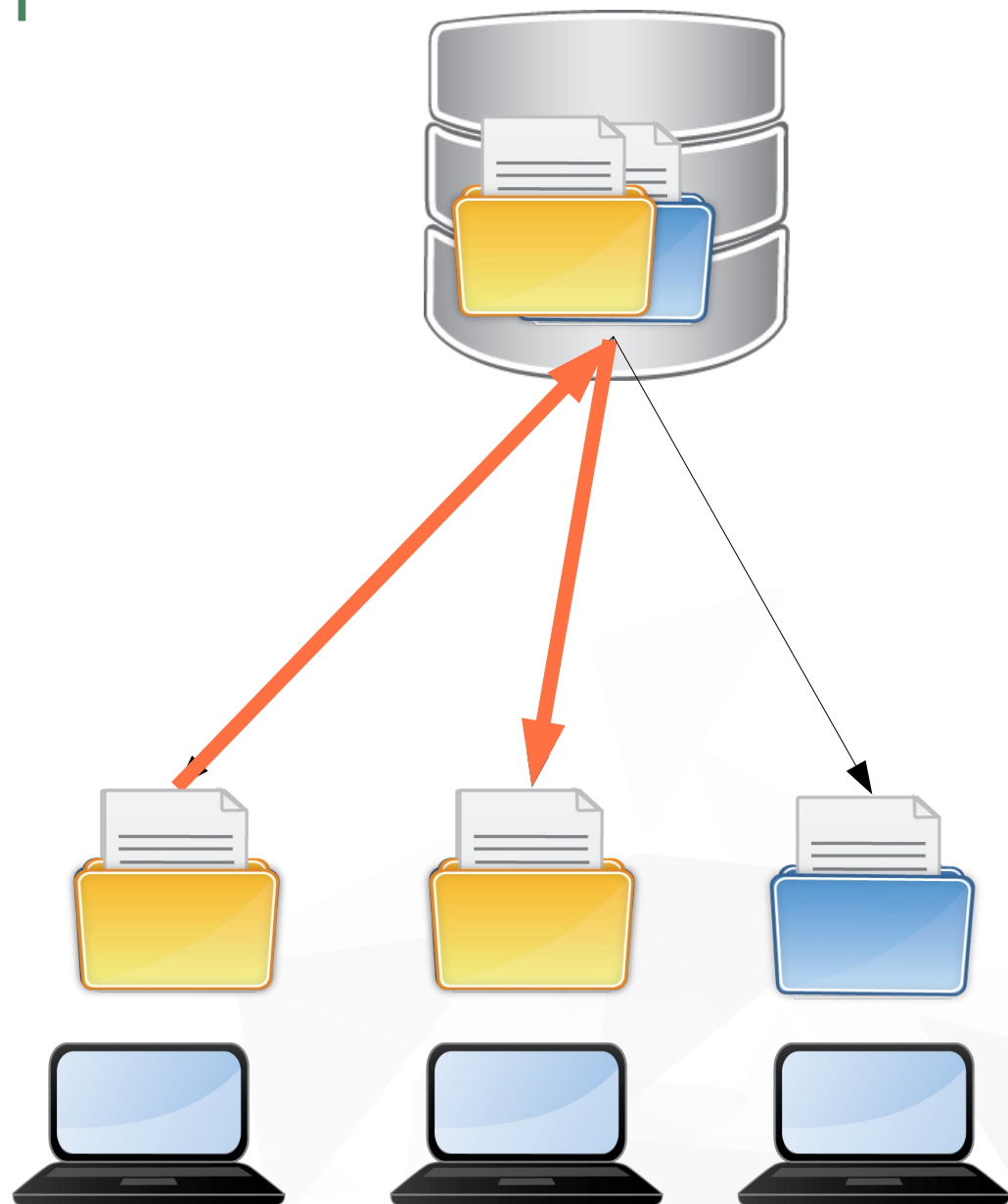
- ▼ Comment gérer les différentes versions
- ▼ Comment rattraper les erreurs
- ▼ Problèmes de path
- ▼ Travail en direct sur le serveur

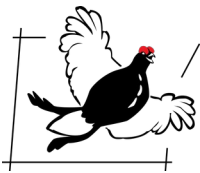




Gestionnaire de version

- ▼ 1 projet = n scripts cohérents
- ▼ Code de base partagé
- ▼ Chacun travaille sur une copie
- ▼ Les modifications sont rendues publiques quand elles sont prêtes





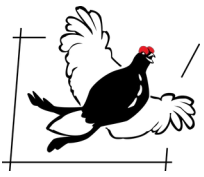
Plan

1) Configurer son environnement Python

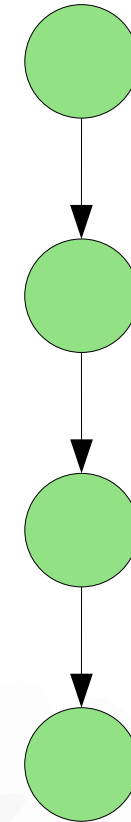
- 1) De Python à Anaconda
- 2) Utiliser les environnements virtuels
- 3) Exercices de prise en main

2) **Développer en mode collaboratif**

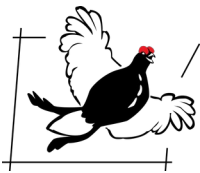
- 1) Du script partagé aux projets
- 2) **Gestionnaire de version**
- 3) Couche sociale



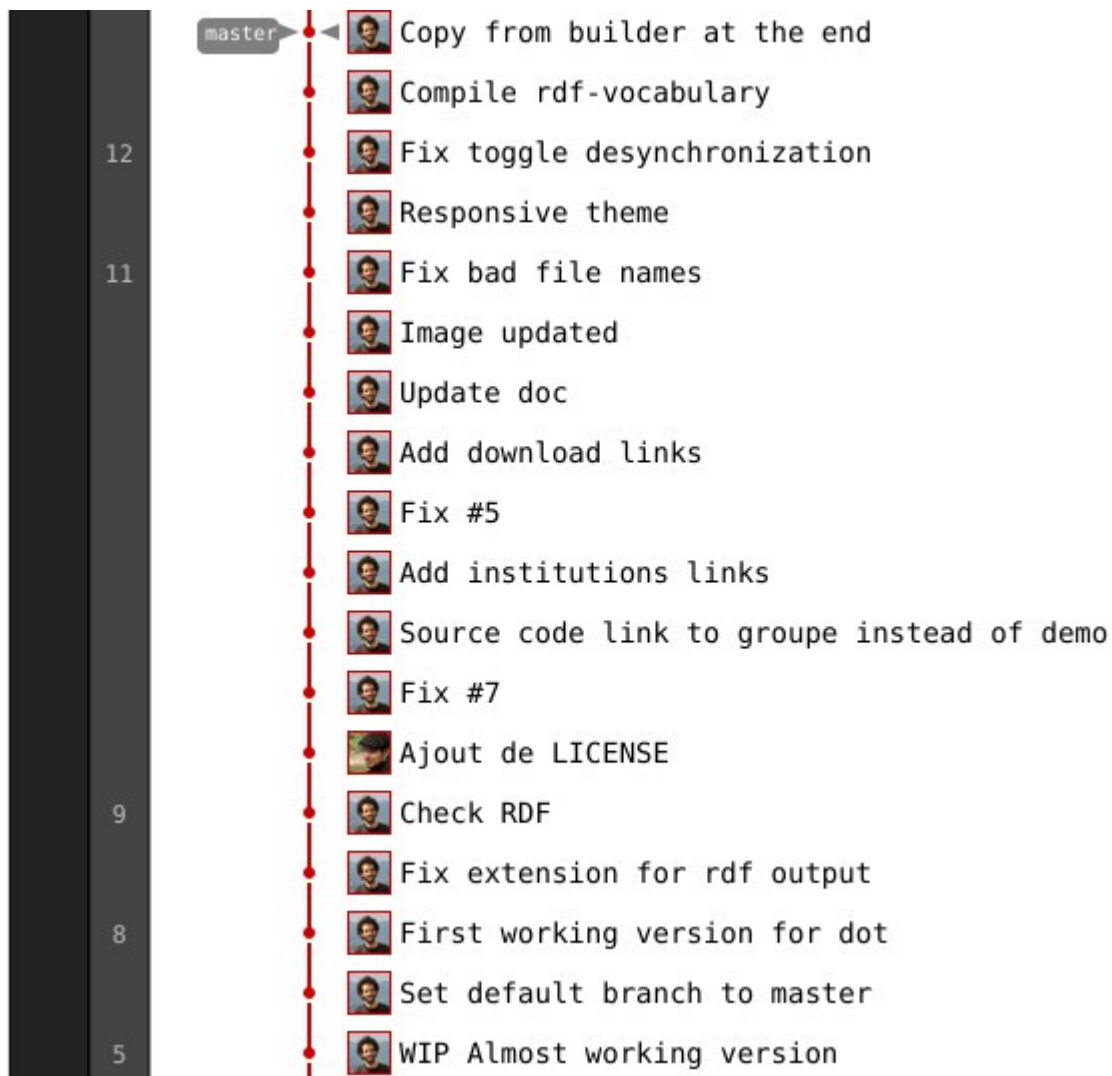
- ▼ Commit :
 - ▼ Enregistrement de modifications
 - ▼ Sur n'importe quel type de fichier
- ▼ Git maintient un « arbre » de commits
- ▼ Utilisation d'un serveur « remote »

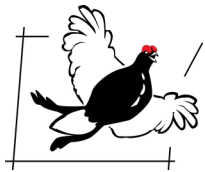


Master

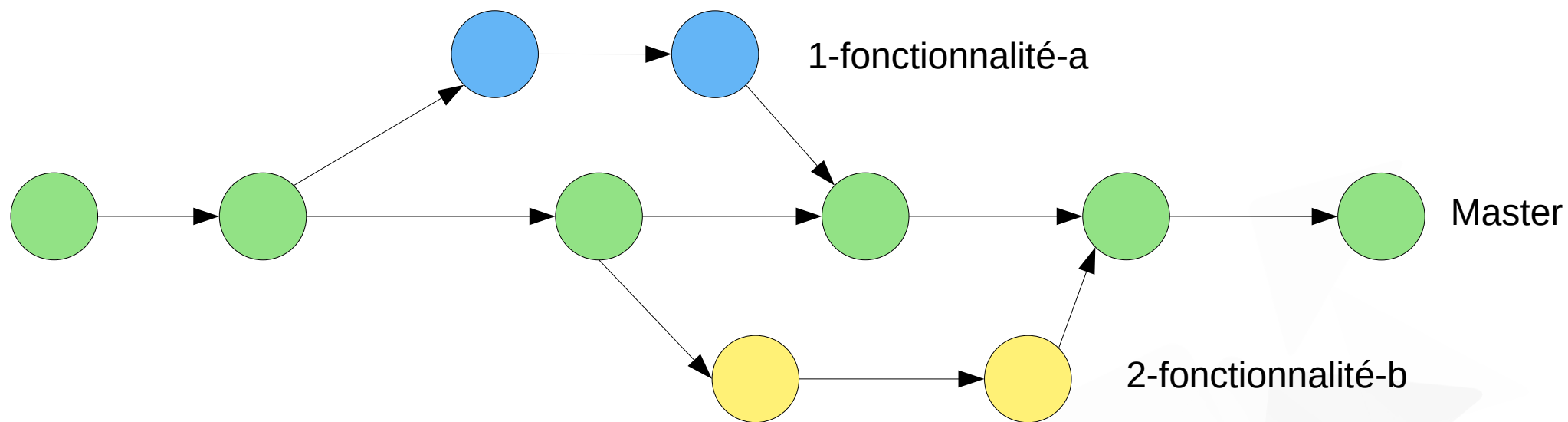


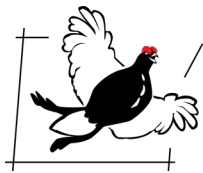
Développement linéaire



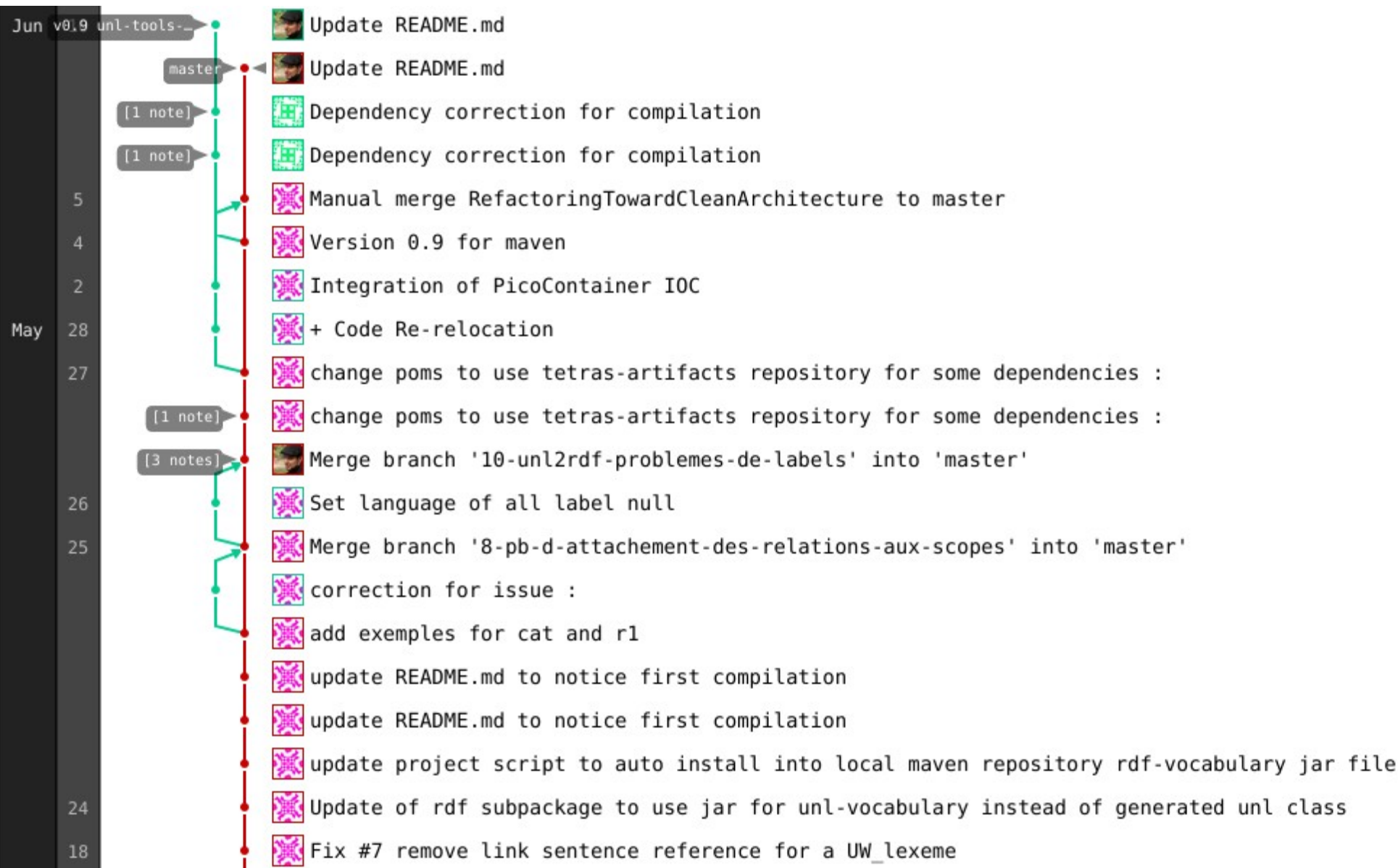


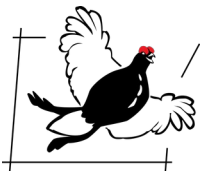
Utilisation des branches



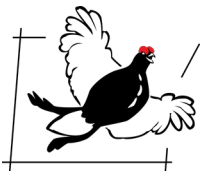


Un exemple





- ▼ Interface en ligne de commande
- ▼ Coût d'apprentissage
- ▼ Demande une vision globale du code comme un projet



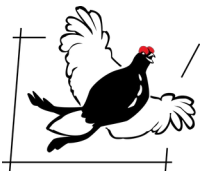
Plan

1) Configurer son environnement Python

- 1) De Python à Anaconda
- 2) Utiliser les environnements virtuels
- 3) Exercices de prise en main

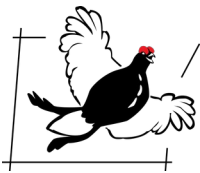
2) **Développer en mode collaboratif**

- 1) Du script partagé aux projets
- 2) Gestionnaire de version
- 3) **Couche sociale**



Gitlab / Github : une couche sociale

- ▼ Suivi des “Issues”
 - ▼ Discussion
 - ▼ Vote
 - ▼ Assignment
- ▼ Vision du projet
 - ▼ Documentation (Wiki)
 - ▼ Jalon
 - ▼ Tableaux de board (Kanban)
 - ▼ Revue de code (demande de fusion)
- ▼ Visualisation des branches et contributions
- ▼ Lien avec d'autres outils (Mattermost / Slack, ...)



Issues

UNL > web-service-demo > Issues > #8

Open Opened 3 weeks ago by David Rouquet

Close issue

New issue

UNL avec deux phrases, le SVG téléchargé est en fait deux SVGs concaténés



Pour une UNL avec deux phrases (l'exemple prérempli), le SVG téléchargé est en fait composé deux SVGs concaténés. Le XML n'est donc pas valide.

De là, selon l'outil utilisé pour afficher le SVG :

1. Soit les deux SVG sont affichés (par exemple dans un navigateur Web)
2. soit un seul SVG est affiché (par exemple dans Inkscape)
3. Soit une erreur est retournée (par exemple dans un script Python qui parse le DOM pour exploiter le XML)

TODO : REgarder le DOT généré et voir s'il y a des options dans Graphviz pour obtenir un SVG correct

Edited 2 days ago by David Rouquet



0



0



Show all activity

Create merge request



David Rouquet @daxid changed title from UNL avec plusieurs phrases, dans le SVG téléchargé, seul que le graphe de la première phrase est visible to UNL avec deux phrases, le SVG téléchargé est en fait deux SVGs concaténés 2 days ago



David Rouquet @daxid changed the description 2 days ago



David Rouquet @daxid · 2 days ago

Owner



Un exemple de dot généré à partir de deux phrase (ce sont deux digraph concaténés) :

```
digraph G {
graph [fontname="courier", compound="true"];
1 [
label="provide(cob>thing,icl>equip(icl>do),obj>thing)\n.@entry.@present.@promise"
shape="box"
fontname="courb"
1.
```



To Do

Add a To Do



Assignee

Edit



David Beniamine
@dbeniamine

Milestone

Edit

None

Time tracking



No estimate or time spent

Due date

Edit

None

Labels

Edit

None

Confidentiality

Edit

Not confidential

Lock issue

Edit

Unlocked

2 participants



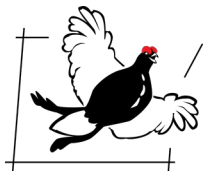
Notifications



Reference: unl/web-service-dem...



Move issue



Issue (board)



Urgent3+

options d'impression limitée selon type de fichier
InformationRequired
#23 ⌚ 4h

Logiciels allophones
#58

Sauvegarde automatique libreoffice fermeture session
bug
#68

InformationRequired1+

options d'impression limitée selon type de fichier
Urgent
#23 ⌚ 4h

toMerge0

Closed63

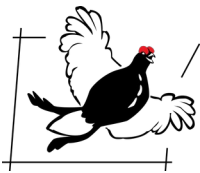
Set default passwords
#1 📅 Sep 30, 2018

Ansible
#2 📅 Sep 17, 2018


Send @MAC / hostname to partage
#3 📅 Sep 17, 2018

Impossible d'installer java oracle
#4

Add packages to auto install
#5



UNL > unlTools > Snippets > \$7

Author 4 hours ago by  David Beniamine

Appeler unlTools en python

Embed <script src="https://gitlab.tetras-libre.fr/unl/unlTools/snippets/7.js"></script>



 unltools.py 2.17 KB 



```
1 def convertUnl(unl, outputs, path):
2     ...
3     Converts a unl string to dot and / or svg, returns a dictionary containing results
4     Parameters:
```



0



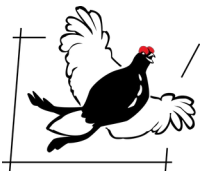
0



Write Preview

B *I* ” </> 🔗 ☰ ☷ ☸ ☹ ↗

Write a comment or drag your files here...



Rendre un snippet “utilisable”

▼ Utilisation de la bibliothèque [click](#)

▼ Exemple

▼ [Code](#)

```
$ ./unlizeXml.py --help
Usage: unlizeXml.py [OPTIONS] INPUT OUTPUT

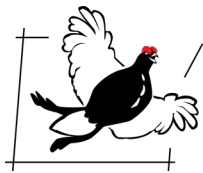
Options:
  --lang [en|ru]
  --dry-run / --no-dry-run  if true do not send request to unl.ru
  --svg / --no-svg         Add svg node representing unl graph
  --unltools-path FILE     Path of the unltools jar
  --help                   Show this message and exit.
```

▼ Création d'interface web via [flask](#)

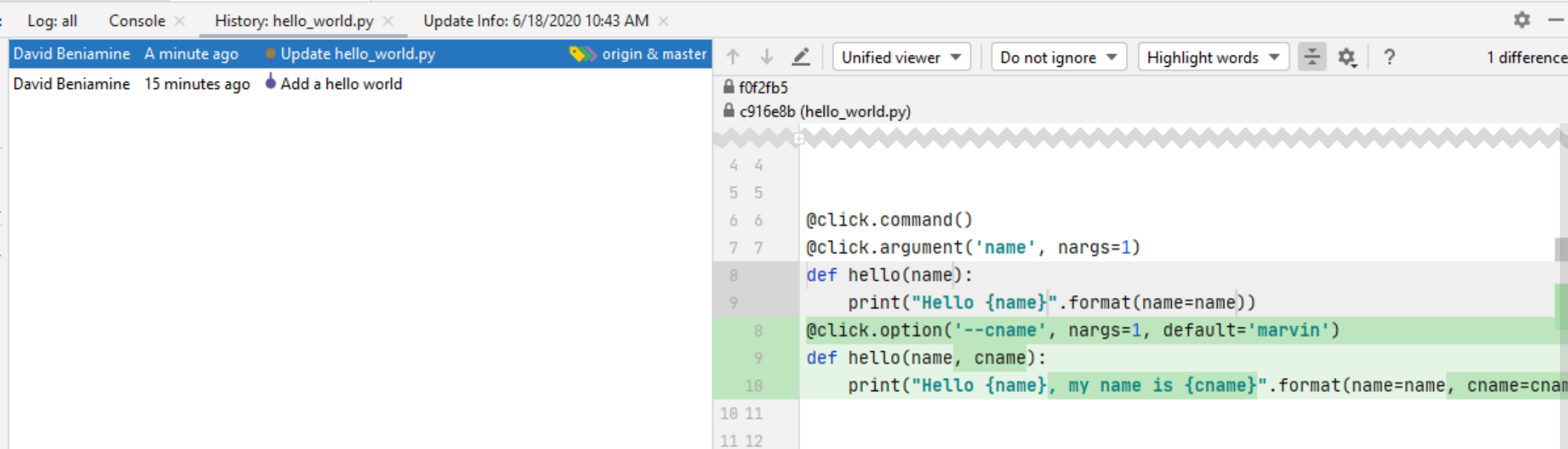
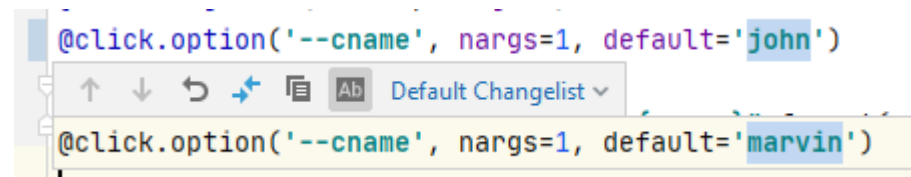
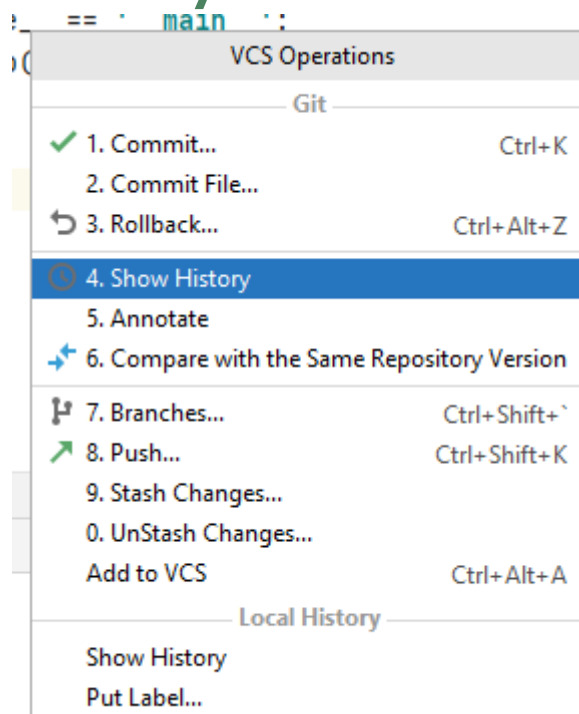
▼ Exemple

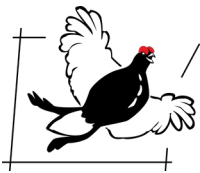
▼ [Code](#)

▼ [Demo](#)



Git dans Pycharm





Mise en pratique

- ▼ Historique Gitlab

<https://gitlab.tetras-libre.fr/unl/unlTools/-/branches>



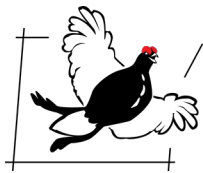
- ▼ Liste des branches

- ▼ Comparaison / demande de fusion

- ▼ Graph

- ▼ Utilisation de git dans Pycharm





Mieux communiquer pour mieux collaborer

- ▼ Utiliser et partager les environnements virtuels



- ▼ Gérer les environnements avec (Ana)conda

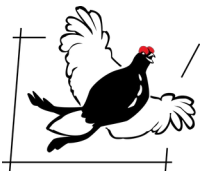


- ▼ Suivi des versions avec Git



- ▼ Organiser les projets avec Gitlab et PyCharm





Notre objectif : faciliter votre travail

- ▼ Rédaction de documentation
- ▼ Dépannage
- ▼ Développement / packaging de code
- ▼ Analyse performances / parallélisme

La ligne est ouverte, n'hésitez pas !



8 rue de Mayencin
38 400 Saint Martin d'Hères

Contact@tetras-libre.fr
David.Beniamine@tetras-libre.fr
David.Rouquet@tetras-libre.fr
Sebastien.Curt@tetras-libre.fr

Ce document est propriété de la société *Tetras Libre* (www.tetras-libre.fr)

La mise en page est basée sur le modèle Libre Office (CC BY-SA) :

<https://extensions.libreoffice.org/templates/libreoffice-presentation-templates>

Contact : contact@tetras-libre.fr

Licence :

Ce document et son contenu, sauf mention contraire explicite, est placé sous licence [Creative Commons BY-SA](#).
Vous êtes donc libres de copier, de modifier et de distribuer ce travail, selon les conditions suivantes :

› Attribution : Vous devez citer les auteurs de sa conception, sans pour autant suggérer qu'ils approuvent votre utilisation. La citation de ce contenu doit mentionner : www.tetras-libre.fr

› Partage à l'identique : Si vous reproduisez, diffusez, modifiez cette publication, vous le ferez sous les mêmes conditions ou alors vous demanderez l'autorisation préalable des auteurs.

Mentions complémentaires :

Tous les icônes autres que que des logo d'outils proviennent des bibliothèques standard libreoffice

Exception faite du logo « teminal »



ce logo provient de la banque iconarchive.com et est produit par [alecive](#)

Il est publié sous licence CC-BY-SA 4.0

Tous les logos d'outils appartiennent à leur propriétaire respectifs

