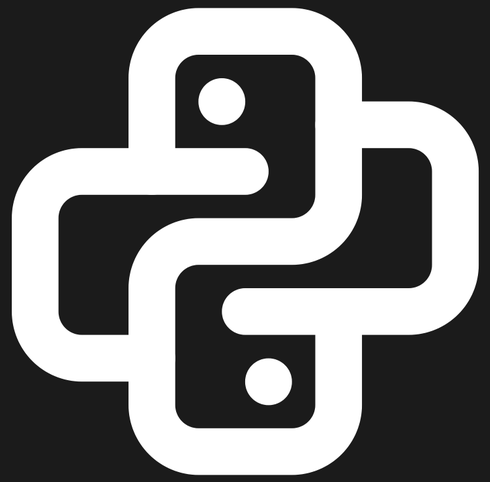
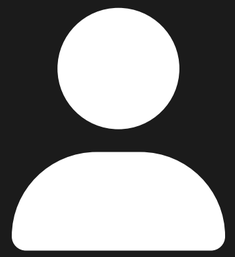


Pretty Output Python

DarkInfern010

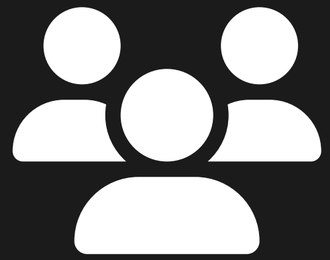
Sommaire

Pourquoi ?



Vous

Apprendre de nouvelles choses
Meilleur visualisation de son code



Equipe

Meilleur compréhension
Visualisation des idées

Autres

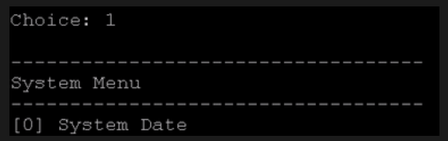
Elargissement du public
S'exprimer d'une autre façon

Comment ?


Édite des lignes de commande, mémorise des commandes Windows XP
et crée des macros.
DOSKEY [/REINSTALL] [/LISTSIZE=taille] [/MACROS[:ALL | :nom_d'exe]]
[/HISTORY] [/INSERT | /OVERSTRIKE] [/EXENAME=nom_d'exe]
[/MACROFILE=nom_de_fichier] [nom_de_macro=[texte]]
/REINSTALL Installe une nouvelle copie de Doskey.
/LISTSIZE=taille Définit la taille du tampon d'historique des commandes
/MACROS Affiche toutes les macros de Doskey.
/MACROS:ALL Affiche toutes les macros de Doskey de tous les exécutables qui en ont.
/MACROS:nom_d'exe Affiche toutes les macros de Doskey d'un exécutable.
/HISTORY
/INSERT
/OVERSTRIKE
/EXENAME=nom_d'exe Spécifie l'exécutable.
/MACROFILE=nom_de_fic. Spécifie un fichier de macros à installer.
nom_de_macro Spécifie un nom pour une macro que vous créez.
texte Spécifie des commandes que vous voulez enregistrer.
Flèches HAUT/BAS rappellent les
l'historique ; ALT+F7 efface l'hi
F9 choisit une commande par son n
; ALT+F10 efface les déf. de macros.
Les codes spéciaux suivants s'uti pour définir les macros Doskey :
$T Séparateur de commande.
$1-$9 Paramètres de batch.
TERMINAL


Interface choix prog !
Donnée arg1
Donnée arg2
Donnée arg3
Donnée arg4
INTERFACE
Choix du programme
PROG1
PROG2
PROG3
PROG4
PROG5
PROG6

Modules



PROGRESS

Permet de visualiser l'avancement d'étapes.

Différents style de bar possible

Totalement personnalisable

ART

Génération automatique de text ASCII

Grand choix de font

Emoji Japonais

SIMPLE-TERM-MENU

Affichage et gestion de menu

Évite l’affichage de menu à choix

Interactif

PANDAS

Affichage de tableau

Compréhension de fichier CSV

Permet également la manipulation de données

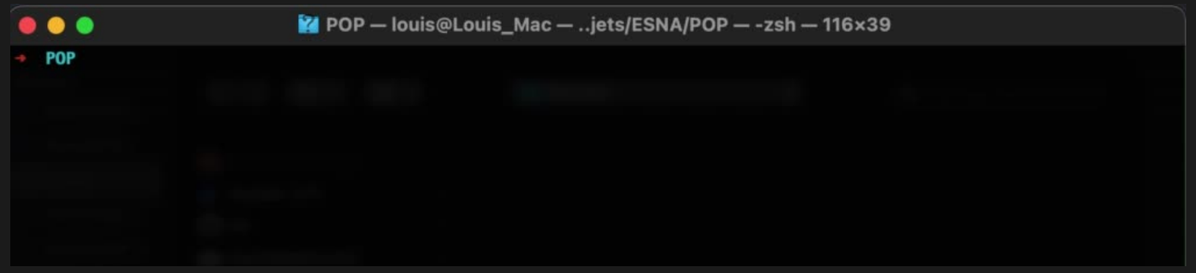
RICH

Énorme choix de visualisation

Affichage et compréhension de texte

Syntax Highlight

Progress

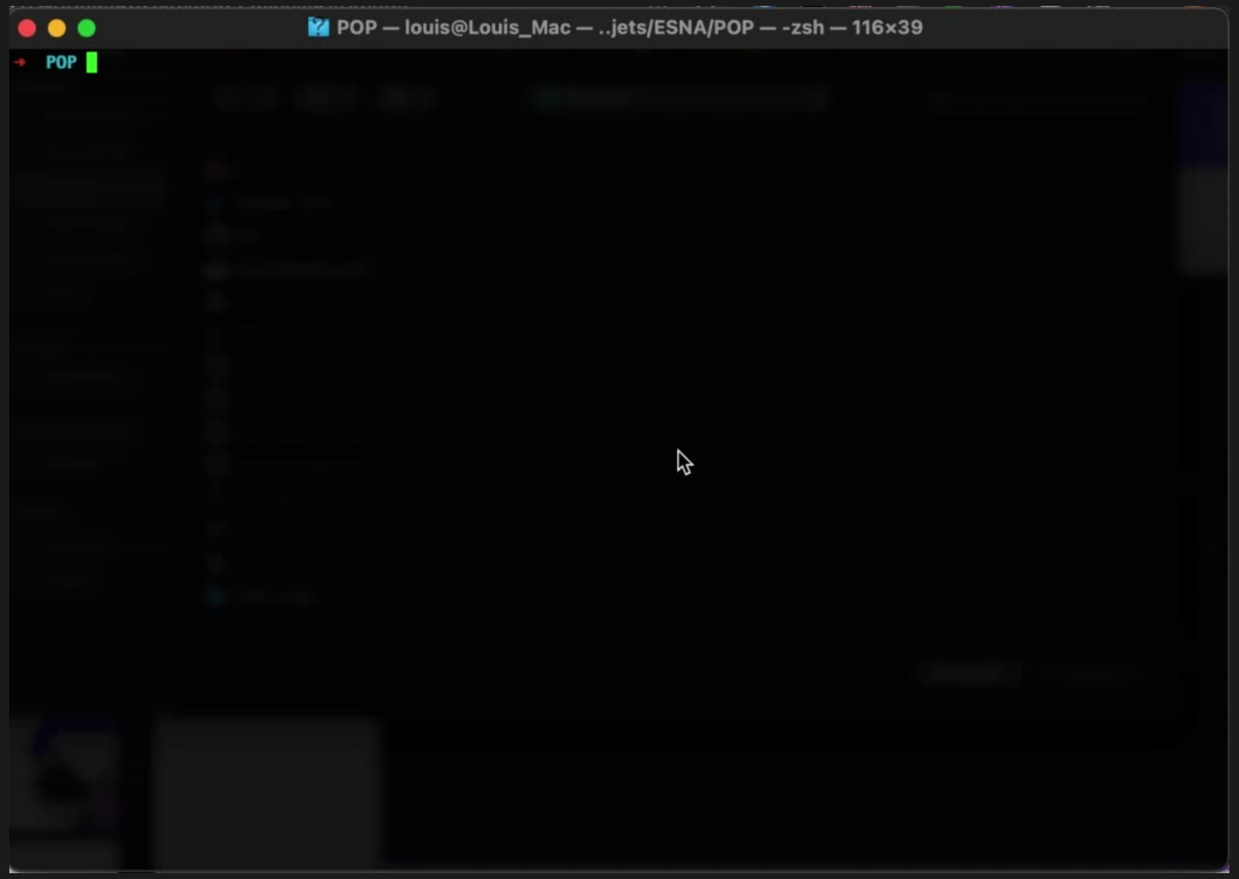


```python
from progress.bar import Bar
import time

bar = Bar('Processing', max=10)
for i in range(10):
    time.sleep(0.5)
    bar.next()
bar.finish()

```

Art

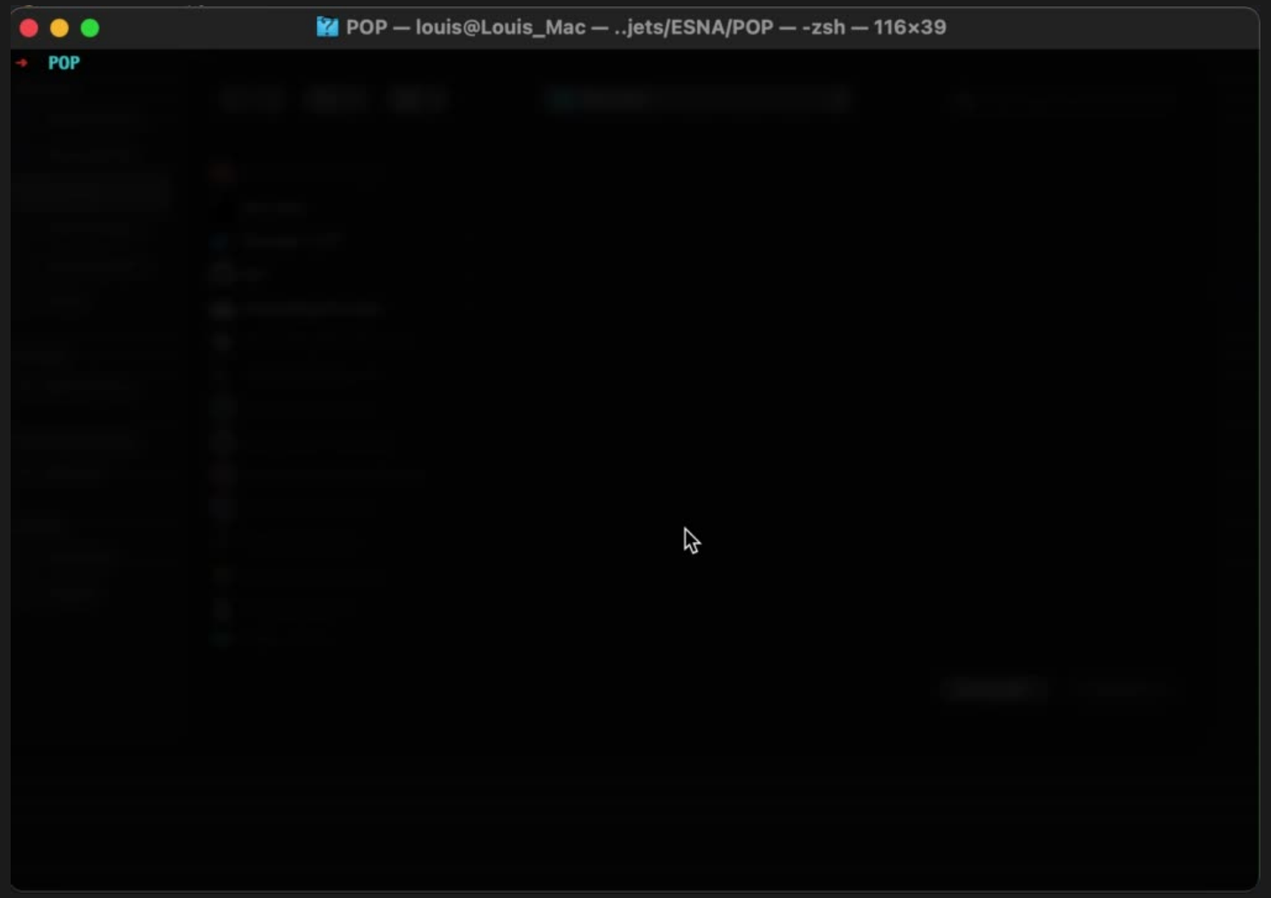



```python
from art import *

tprint("POP")
print(text2art("pop",font='block',chr_ignore=True))
print(text2art("POP", "random"))

print(art("coffee"))
print(art("happy"))
aprint("butterfly")
```

Menu



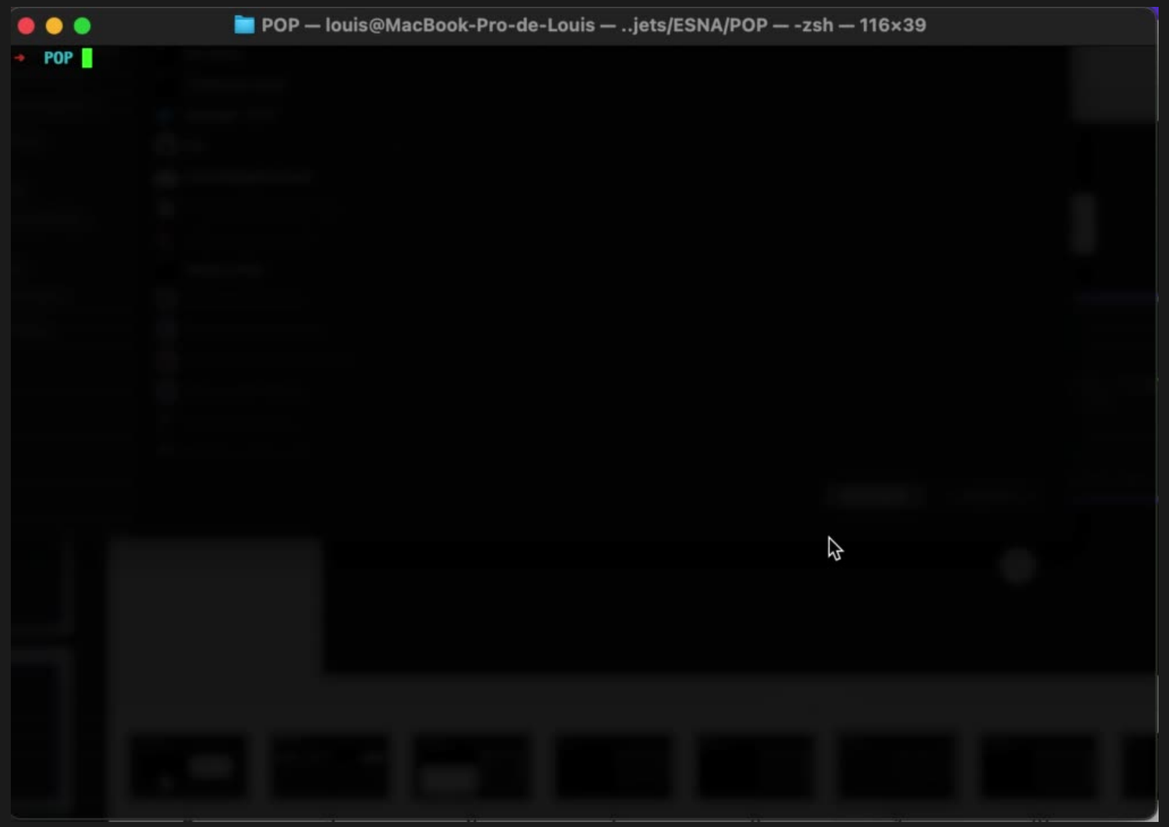

```python
from simple_term_menu import TerminalMenu

def helloworld(type):
    print("hello world".upper() if type == "upper"
          else "hello world")
#endef

options = ["hello world", "HELLO WORLD"]
menu = TerminalMenu(options)
index = menu.show()

if index == 0:
    helloworld("normal")
elif index == 1:
    helloworld("upper")
```

Panda



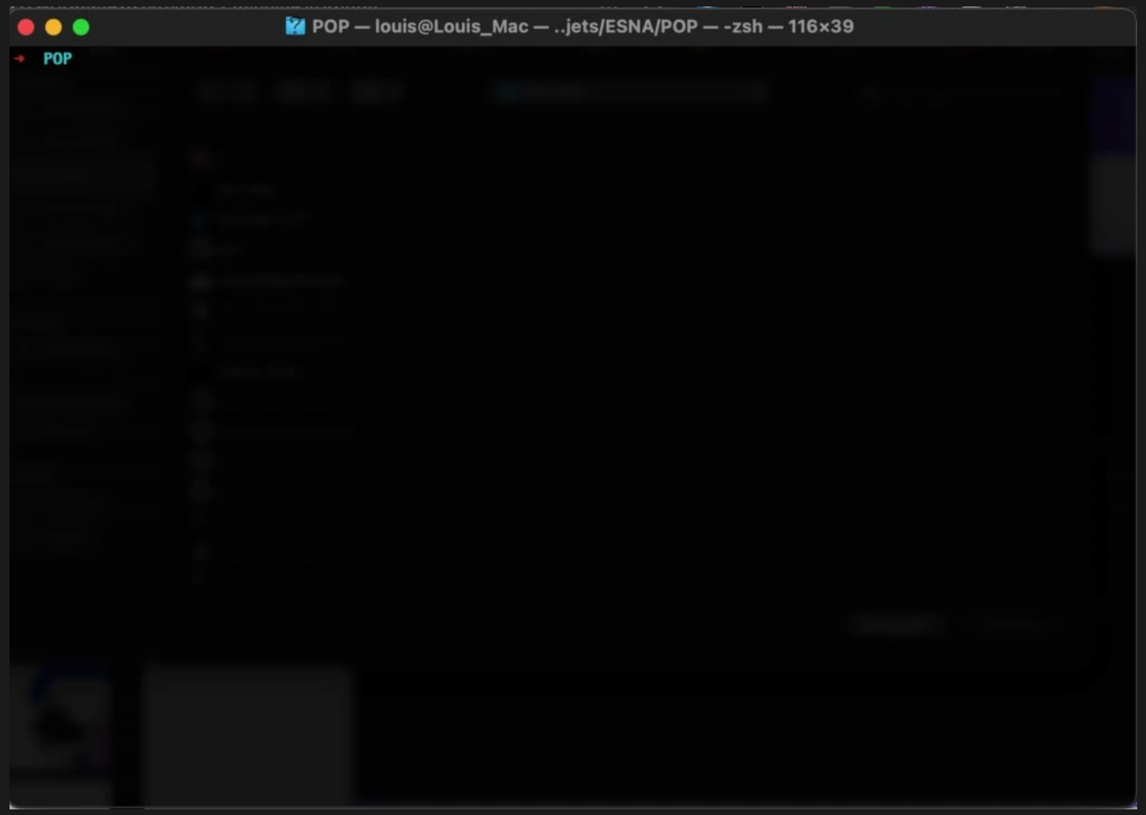

```python
from tabulate import tabulate
import pandas

data = {'Auteur': ["Marc Levy", "Laétitia Colombani", "Guillaume Musso"],
        'Titre': ["Et si c'était vrai...", "La tresse", "Central Park"],
        'Editeur': ["Flammarion", "Gallimard", "Milan"],
        'Prix': [10, 15, 20]
        }

dataframe = pandas.DataFrame(data)

print(tabulate(dataframe, headers = 'keys', tablefmt = 'psql'))
```

Colonne




```python
data = ["pets"+str(i) for i in range(1,52)]
print(data)

NumberCol = int(len(data)/3)
arrayShow = [""] * NumberCol
compt = 0
for i in data:
    if compt < 10:
        arrayShow[compt % NumberCol] += "[0" + str(compt) + "] - " + i + "//"
    else:
        arrayShow[compt % NumberCol] += "[" + str(compt) + "] - " + i + "//"
    compt += 1
for j in arrayShow:
    test = j.split("//")
    print(test[0] + " " * (30 - len(test[0])) +
          test[1] + " " * (30 - len(test[1])) +
          test[2] + " " * (30 - len(test[2])))
```

Rich


-bash
python -m rich
Rich features
Colors
√ 4-bit color
√ 8-bit color
√ Truecolor (16.7 million)
√ Dumb terminals
√ Automatic color conversion
Styles
All ansi styles: bold, dim, italic, underline, strikethrough, reverse, and even blink.
Text
Word wrap text. Justify left, center, right or full.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque in metus sed sapien ultricies pretium a at justo. Maecenas luctus velit et auctor maximus.
Asian language support
该库支持中文，日文和韩文文本！
ライブラリは中国語、日本語、韓国語のテキストをサポートしています
이 라이브러리는 중국어, 일본어 및 한국어 텍스트를 지원합니다
Markup
Rich supports a simple bbcode like markup for color, style, and emoji!
Tables
Date
Title
Production Budget
Box Office
Dec 20, 2019 Star Wars: The Rise of Skywalker $275,000,000 $375,126,118
May 25, 2018 Solo: A Star Wars Story $275,000,000 $393,151,347
Dec 15, 2017 Star Wars Ep. VIII: The Last Jedi $262,000,000 $1,332,539,889
May 19, 1999 Star Wars Ep. I: The phantom Menace $115,000,000 $1,027,044,677
Syntax highlighting & pretty printing
def iter_last(values: Iterable[T]) -> Ite
"""Iterate and generate a tuple with
iter_values = iter(values)
try:
previous_value = next(iter_values
except StopIteration:
return
for value in iter_values:
yield False, previous_value
previous_value = value
yield True, previous_value
{
'foo': [
3.1427,
(
'Paul Atreides',
'Vladimir Harkonnen',
'Thufir Hawat'
)
],
'atomic': (False, True, None)
}
Markdown
Markdown
Supports much of the *markdown*, __syntax__!
- Headers
- Basic formatting: **bold**, *italic*, `code`
- Block quotes
- Lists, and more...
Markdown
Supports much of the markdown, syntax!
• Headers
• Basic formatting: bold, italic, code
• Block quotes
• Lists, and more...
+more!
Progress bars, columns, styled logging handler, tracebacks, etc...

Bonus

AUTO RELOAD

Petit script permettant de relancer le code après modification du fichier.

Gain de temps lors du développement

Disponible sur le Discord ESN'HACK

```python
import os, sys, time, hashlib, subprocess

def md5(fname):
    hash_md5 = hashlib.md5()
    with open(fname, "rb") as f:
        for chunk in iter(lambda: f.read(4096), b""):
            hash_md5.update(chunk)
    return hash_md5.hexdigest()

def main():
    checkValue = ""
    process = None
    while True:
        newCheck = md5(sys.argv[-1])
        if newCheck != checkValue:
            os.system('cls' if os.name=='nt' else 'clear')
            print("##### RELOAD "+sys.argv[-1]+" #####")
            checkValue = newCheck

            if process != None:
                if process.poll() == None:
                    process.kill()
            process = subprocess.Popen(['python3', sys.argv[-1]])

        time.sleep(1)
if __name__ ==  '__main__':
    main()
```