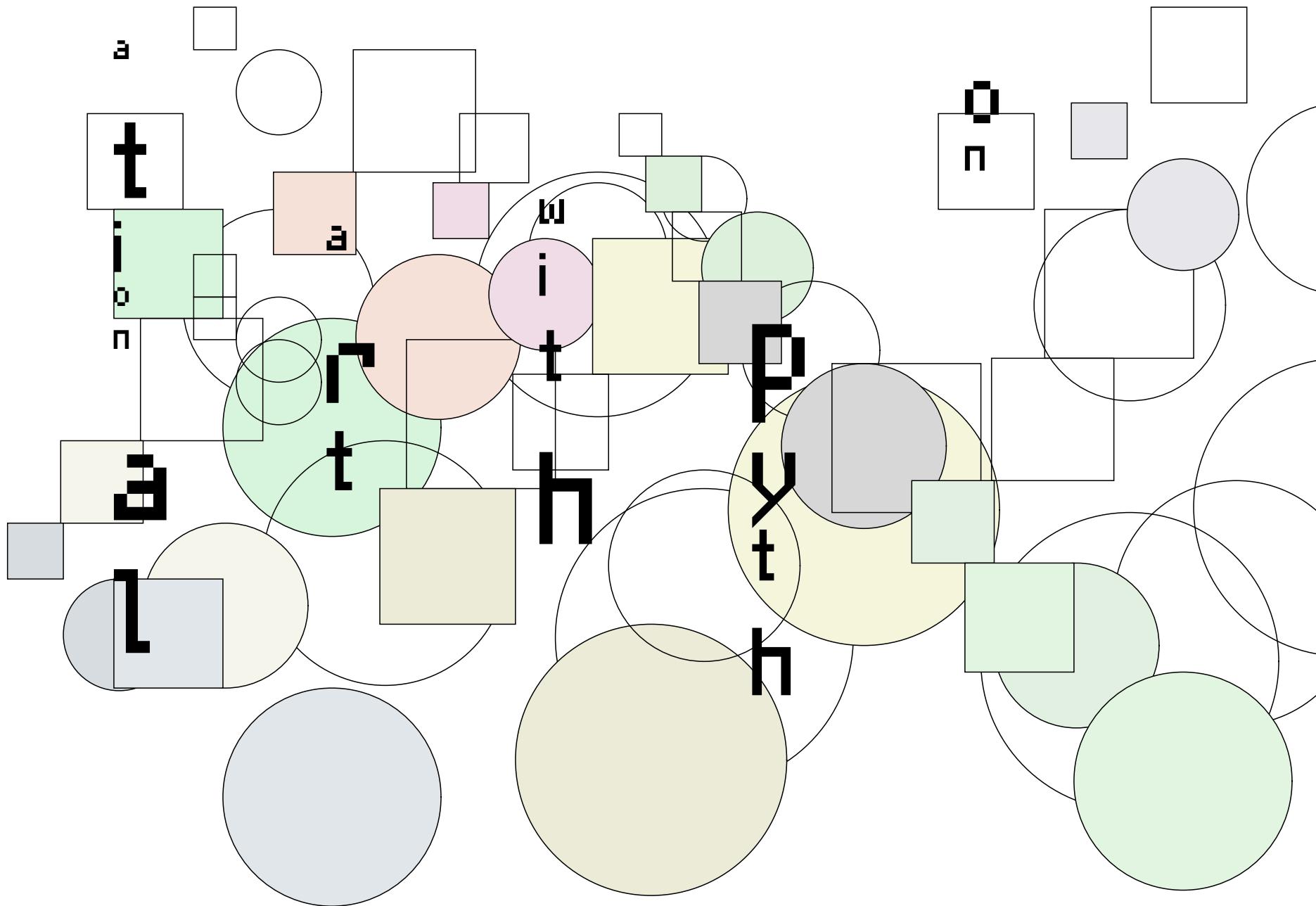


c  
o  
m  
p  
t



# computational art + python I, II, III

[Git repository of the course](#)

artistic references

python introduction/poems

python + scraping

python + composing

# artistic references

Computational Poems

The Great Netflix

Videogrep

Unerasable Characters

# The Great Netflix

by Linda Hilfling Ritasdatter and Kristofer Gansing

Media Infrastructures Beyond the Cloud  
artistic research into “de-clouding” and  
redistributing contemporary cloud-based  
and algorithmic media platforms



# The Great Netflix – A video Store After the End of the World



scraping movies with VPN to change locations  
playing them and recording them in videotapes

# video recording



# Videogrep

by Sam Lavigne



The Meta Experience



Donald Trump saying his own name for over 4 minutes (sorry)

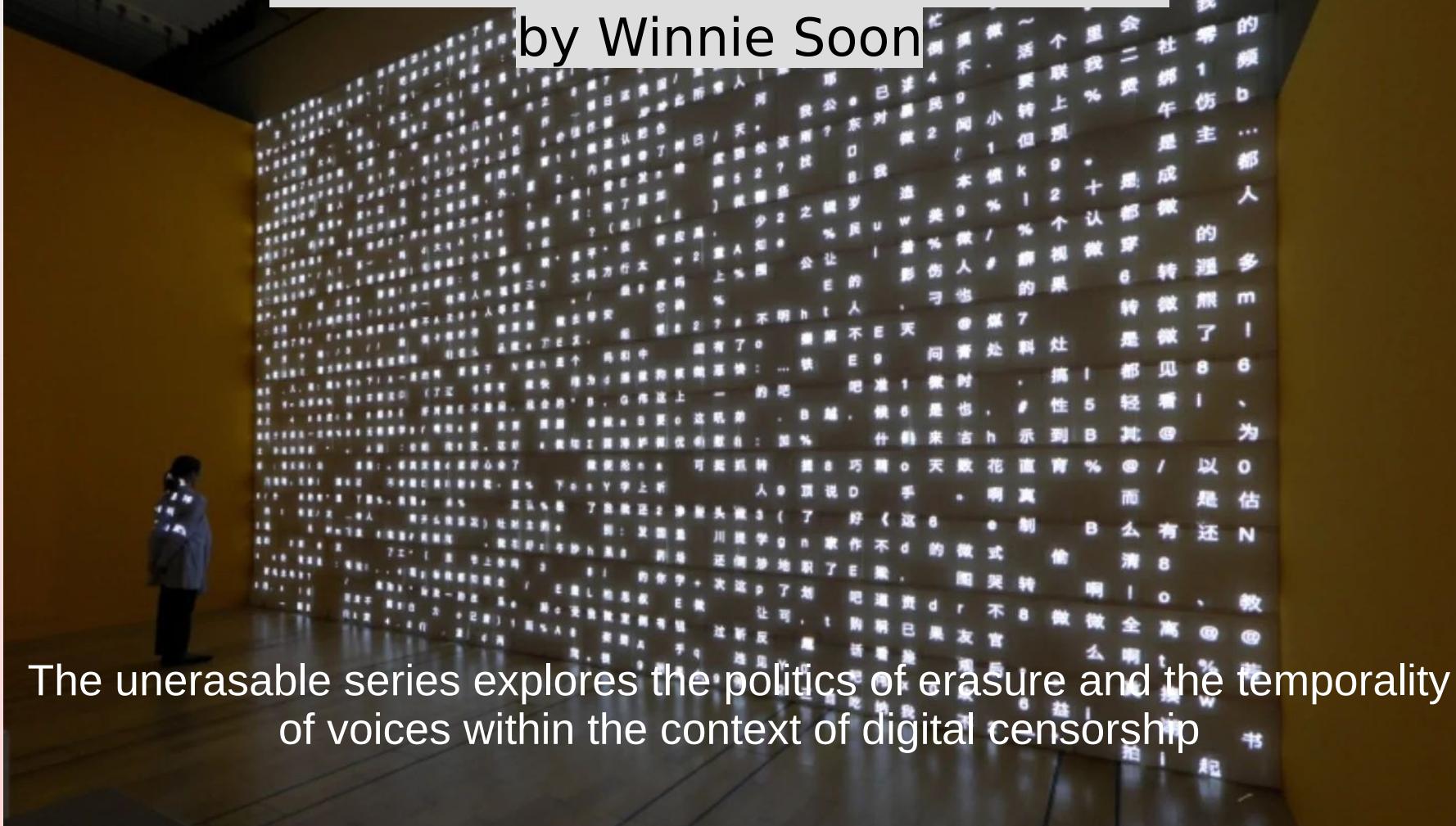
a python script that searches through dialog in videos and then cuts together a new video based on what it finds

[videos](#)

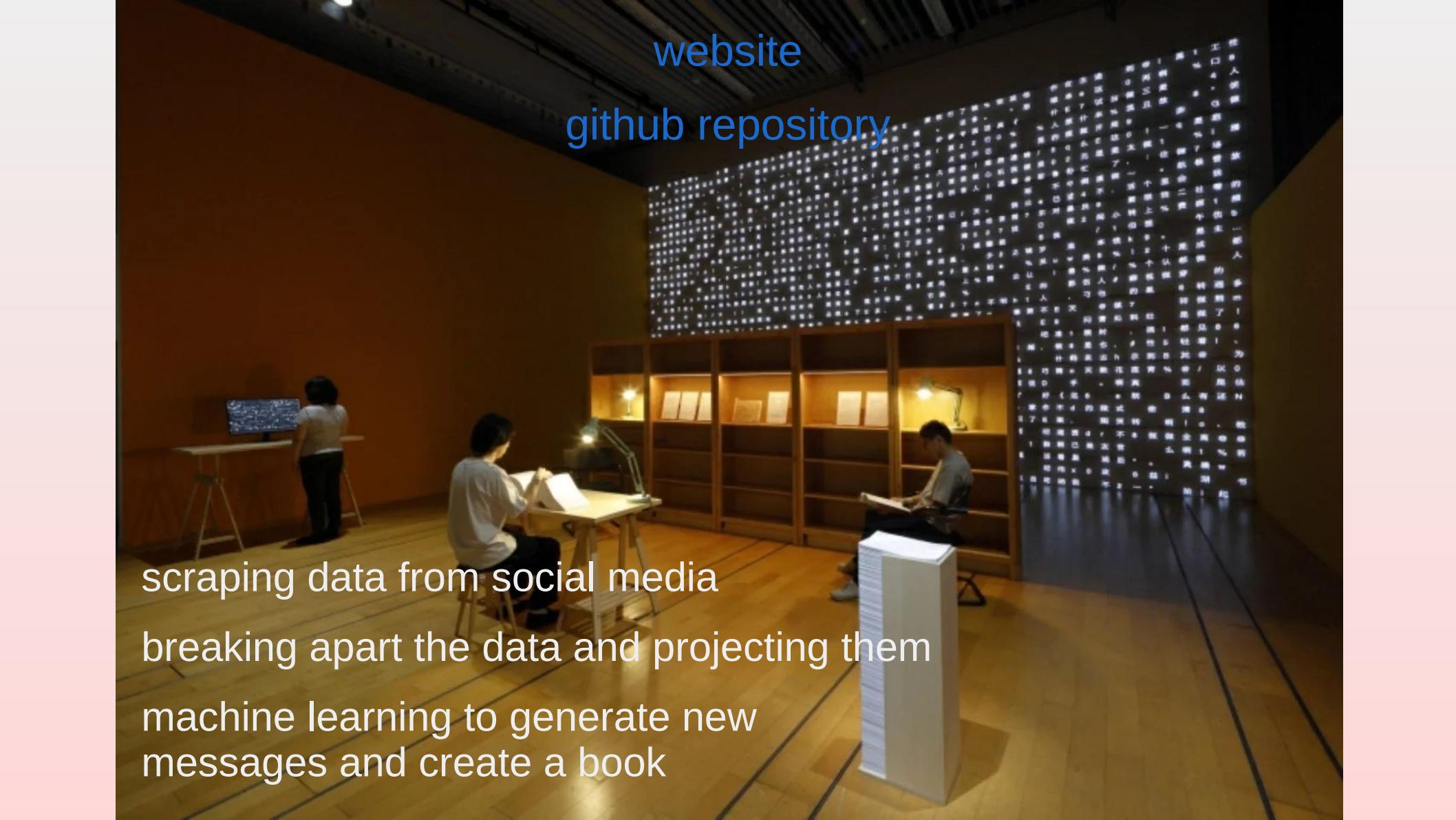
[github repository](#)

# Unerasable Characters

by Winnie Soon



The unerasable series explores the politics of erasure and the temporality of voices within the context of digital censorship



website

github repository

scraping data from social media

breaking apart the data and projecting them

machine learning to generate new  
messages and create a book

# Les Deux

by Nick Montfort

L'élève frappe à la porte du professeur.  
Elle le défie.  
Ils attendent en silence.

L'élève frappe à la porte du professeur.  
Elle la défie.  
Six ans après, ni l'un ni l'autre ne se souvient de l'incident.

L'entraîneur s'adresse à l'haltérophile.  
Il le frappe.  
Six ans après, ni l'un ni l'autre ne se souvient de l'incident.

L'éclairagiste s'avance vers l'équilibriste.  
Elle le prend dans ses bras.  
Ils éclatent de rire.

L'éclairagiste s'avance vers l'équilibriste.  
Il la conseille.  
Ils éclatent de rire.

L'éclairagiste s'avance vers l'équilibriste.  
Elle le congratule.  
Six ans après, ni l'un ni l'autre ne se souvient de l'incident.

L'élève frappe à la porte du professeur.  
Elle le défie.  
Ils attendent en silence.

L'élève frappe à la porte du professeur.  
Elle la défie.  
Six ans après, ni l'un ni l'autre ne se souvient de l'incident.

L'entraîneur s'adresse à l'haltérophile.  
Il le frappe.  
Six ans après, ni l'un ni l'autre ne se souvient de l'incident.

L'éclairagiste s'avance vers l'équilibriste.  
Elle le prend dans ses bras.  
Ils éclatent de rire.

L'éclairagiste s'avance vers l'équilibriste.  
Il la conseille.  
Ils éclatent de rire.

L'éclairagiste s'avance vers l'équilibriste.  
Elle le congratule.  
Six ans après, ni l'un ni l'autre ne se souvient de l'incident.

# source code



A screenshot of a web browser window displaying the source code of a Python file. The URL in the address bar is [https://nickm.com/2/les\\_deux.py](https://nickm.com/2/les_deux.py). The page content includes a copyright notice, a compatibility notice, and the Python code for a story generator.

```
Copyright 2010-2015 Nick Montfort and Serge Bouchardon

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved. This file is offered as-is,
without any warranty.

Version JavaScript: http://nickm.com/2/les\_deux.html
29 juillet 2018: CompatibilitÃ© avec Python 2/3 en ajoutant des parenthÃ"ses
"""

from random import choice
from time import sleep

commencement = [
    "L'agent de police s'approche de l'automobiliste",
    "Le procureur regarde l'agresseur",
    "L'anesthÃ"site se tourne vers le mÃ"decin",
    "L'Ã"olÃ"ve frappe Ã  la porte du professeur",
    "L'Ã"clairagiste s'avance vers l'Ã"quilibriste",
    "L'entraÃ"neur s'adresse Ã  l'haltÃ"rophile",
    "L'Ã"conomiste interpelle l'entrepreneur"
]

verbe = [
    "sermonne",
    "conseille",
    "congratule",
    "rÃ"aconforte",
    "prend dans ses bras",
    "frappe",
    "rudoie",
    "fÃ"olicite",
    "dÃ"ifie",
    "rejette"
]
```

# Python introduction I

**variables, input, print()**

**conditional statements if/elif/else**

**break**

**list, dictionary**

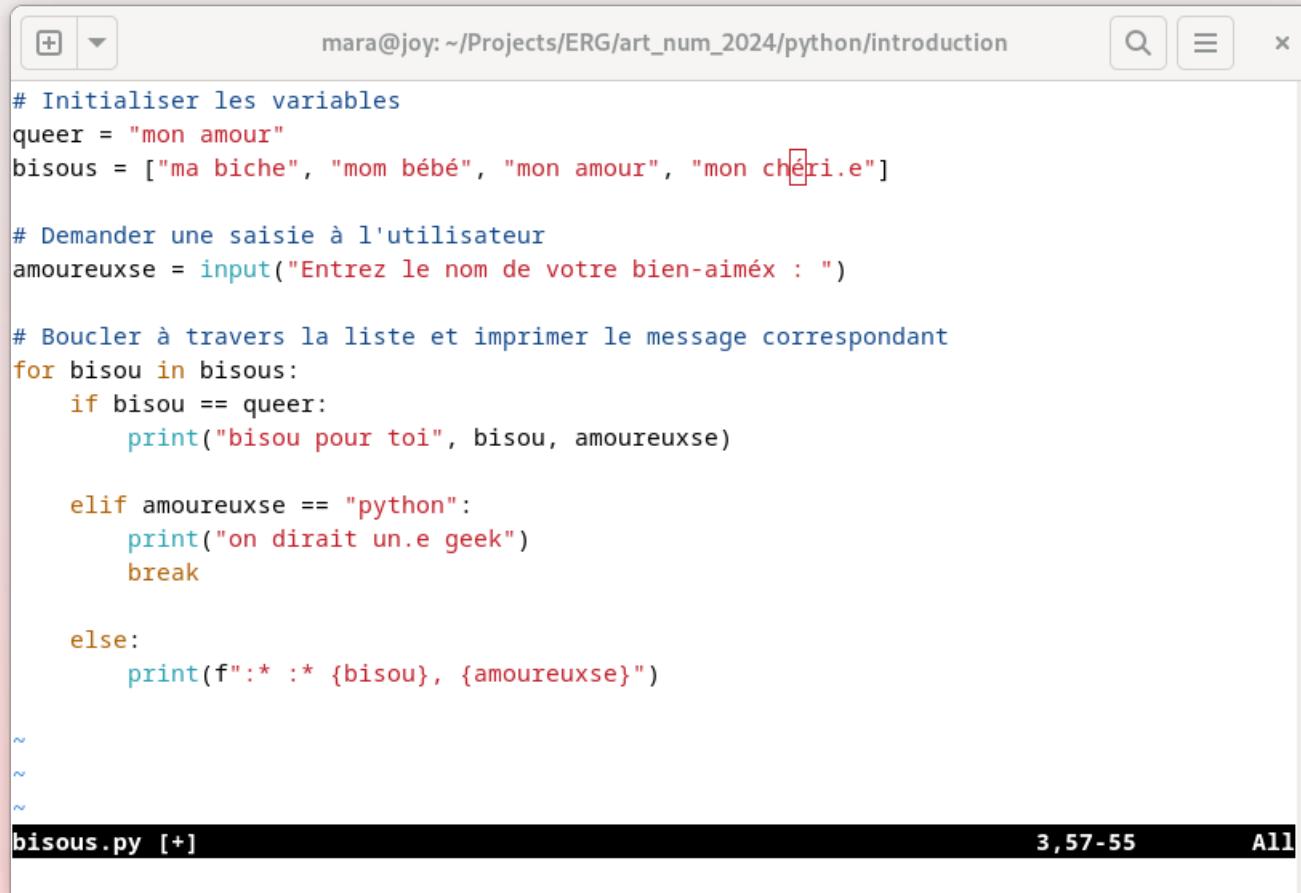
**for loop, while loop**

**random**

**functions**

**return**

# lettres d'amour



The image shows a terminal window with the following content:

```
mara@joy: ~/Projects/ERG/art_num_2024/python/introduction
```

```
# Initialiser les variables
queer = "mon amour"
bisous = ["ma biche", "mon bébé", "mon amour", "mon chérie"]

# Demander une saisie à l'utilisateur
amoureuxse = input("Entrez le nom de votre bien-aimé : ")

# Boucler à travers la liste et imprimer le message correspondant
for bisou in bisous:
    if bisou == queer:
        print("bisou pour toi", bisou, amoureuxse)

    elif amoureuxse == "python":
        print("on dirait un.e geek")
        break

    else:
        print(f":* :* {bisou}, {amoureuxse}")

~
```

The terminal window has a title bar with a search icon, a list icon, and a close button. The bottom status bar shows the file name `bisous.py` and the line number `3,57-55`. The status bar also includes a "All" button.

# Python + scraping II

videogrep for  
making video or audio collage  
working with virtualenv

# videogrep tutorial

Use it from the command line

```
videogrep --input path/to/video.mp4 --search 'search phrase'
```

Use it as a module

```
from videogrep import videogrep

videogrep('path/to/your/files', 'output_file_name.mp4', 'search_term', 'search_type')
```

# Python + composing III

working with [requests](#) and [beautiful-soup](#)

see video examples [here](#)

see audio examples [here](#)