

Après le hard, le soft

OS, IDE and all that

Stéphane Gonnord

stephane@gonnord.org

www.mp933.fr

Lycée du parc - Lyon

Vendredi 20 septembre 2013 - Lycée du parc

Plan

Système d'exploitation

- Son rôle
- Les OS au cours du temps

Contenu d'un disque dur

- Quoi, où ?
- Structure arborescente
- Différents types de fichiers
- Droits sur les fichiers

Apprendre à travailler

- Dossiers et sous-dossiers
- Le choix des noms
- Sauvegardes et versioning

Utiliser un IDE

- Logique globale
- Python for nerds
- Python for newbies

Plan

1. Système d'exploitation :
 - ▶ à quoi ils servent ;
 - ▶ qui ils sont.
2. Le contenu d'un disque dur :
 - ▶ dossiers vs. fichiers ;
 - ▶ documents vs. exécutables ;
 - ▶ à toi, à moi, à lui.
3. Apprendre à travailler avec un ordinateur :
 - ▶ de l'ordre dans ses dossiers et fichiers ;
 - ▶ le choix des noms ;
 - ▶ sauvegardes et gestion des versions.
4. Utiliser un IDE :
 - ▶ Le casting Python
 - ▶ Python for nerds ;
 - ▶ Spyder vs. Idle.

Plan

Système d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

Sauvegardes et versioning

Utiliser un IDE

Logique globale

Python for nerds

Python for newbies

Système d'exploitation (O_perating S_ystem)

- ▶ Est lancé au boot de l'ordinateur, éventuellement après un choix (grub/lilo).
- ▶ Interface entre l'utilisateur, le matériel et les applications.
- ▶ Organisation du «système de fichier» (DD...).
- ▶ Gestion des différents utilisateurs :
 - ▶ dossiers personnels ;
 - ▶ droits en lecture, écriture, exécution.
- ▶ Lancement des applications/processus, gestion du parallélisme.

Après le hard, le soft

Stéphane Gonnord

Plan

Système
d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque
dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

Sauvegardes et versioning

Utiliser un IDE

Logique globale

Python for nerds

Python for newbies

OS : un peu d'histoire

It's a warm summer evening in ancient Greece...



Systèmes d'exploitations des ordinateurs personnels :

- ▶ Unix (1969)

Après le hard, le soft

Stéphane Gonnord

Plan

Système
d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque
dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

Sauvegardes et versioning

Utiliser un IDE

Logique globale

Python for nerds

Python for newbies

OS : situation en 2013

- ▶ Ordinateurs personnels :
 - ▶ Windows : 92%
 - ▶ OS X (macs, base Unix) : 7%
 - ▶ Base Unix (Linux, BSD...) : 1%

Grandes disparités selon les milieux...

- ▶ Serveurs d'entreprise, web, cloud : majoritairement Linux
- ▶ High Performance Computing (gros ordinateurs) :
 - ▶ Base Unix (Linux, BSD...) : 98%
 - ▶ Windows : 2%
- ▶ Smartphones :
 - ▶ Android (Linux) : 79%
 - ▶ iOS (base Linux) : 14%
 - ▶ Blackberry, Windows mobile, ... : < 7%.

Partitionnement du disque

Un disque, beaucoup de partitions

```
stephane@euler:~$ df -BM
```

Filesystem	1M-blocks	Used	Available	Use%	Mounted on
/dev/sda3	182677M	100205M	73266M	58%	/
udev	2001M	1M	2001M	1%	/dev
tmpfs	805M	1M	804M	1%	/run
none	5M	0M	5M	0%	/run/lock
none	2011M	2M	2010M	1%	/run/shm

```
stephane@meert:~$ df -BM
```

Sys. de fichiers	1M-blocs	Utilisé	Disponible	Uti%	Monté sur
/dev/sda1	18645M	6207M	11485M	36%	/
none	1M	0M	1M	0%	/sys/fs/cgroup
udev	992M	1M	992M	1%	/dev
tmpfs	201M	1M	200M	1%	/run
none	5M	0M	5M	0%	/run/lock
none	1002M	0M	1002M	0%	/run/shm
none	100M	0M	100M	0%	/run/user
/dev/sda3	213814M	160679M	42268M	80%	/home
/dev/sdb1	469325M	178937M	266541M	41%	/mnt/savedisk

(en fait, il y a plusieurs disques !)

Après le hard, le soft

Stéphane Gonnord

Plan

Système
d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque
dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

Sauvegardes et versioning

Utiliser un IDE

Logique globale

Python for nerds

Python for newbies

Des tas de fichiers !

```
stephane@euler:~$ find / -type f 2>/dev/null | wc -l  
1398531
```

```
stephane@euler:~$ find ~/ -type f 2>/dev/null | wc -l  
295888
```

```
stephane@meert:~$ find / -type f 2>/dev/null | wc -l  
1512922
```

```
stephane@meert:~$ find ~/ -type f 2>/dev/null | wc -l  
149208
```

Après le hard, le soft

Stéphane Gonnord

Plan

Système
d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque
dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

Sauvegardes et versioning

Utiliser un IDE

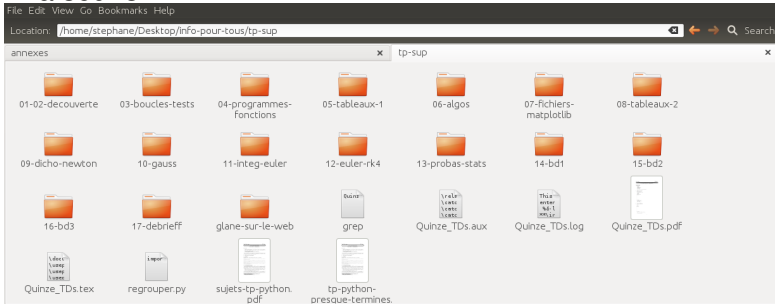
Logique globale

Python for nerds

Python for newbies

Dossiers vs fichiers

À la souris :



Ou en ligne de commande :

```
File Edit View Search Terminal Help
stephane@euler ~$ cd documents/info-pour-tous/tp-sup/
stephane@euler ~/documents/info-pour-tous/tp-sup$ ls
01-02-decouverte  08-tableaux-2  14-bd1          Quinze_TDs.aux  regroupier.py
03-boucles-tests  09-dicho-newton 15-bd2          Quinze_TDs.log  regrouper.py-
04-programmes-fonctions 10-gauss        16-bd3          Quinze_TDs.pdf  sujets-tp-python.pdf
05-tableaux-1      11-integ-euler  17-debrief     Quinze_TDs.tex  tp-python-presque-terminees.pdf
06-algos           12-euler-rk4    glane-sur-le-web Quinze_TDs.tex-
07-fichiers-matplotlib 13-probas-stats grep             regroupier.blnc
stephane@euler ~/documents/info-pour-tous/tp-sup$
```

Après le hard, le soft

Stéphane Gonnord

Plan

Système d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

Sauvegardes et versioning

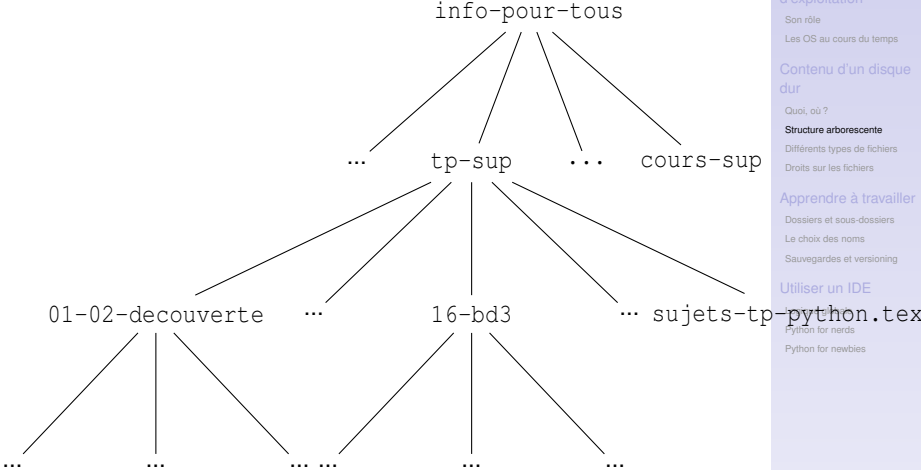
Utiliser un IDE

Logique globale

Python for nerds

Python for newbies

De façon plus abstraite



Plan

Système d'exploitation

- Son rôle
- Les OS au cours du temps

Contenu d'un disque dur

- Quoi, où ?
- Structure arborescente**
- Différents types de fichiers
- Droits sur les fichiers

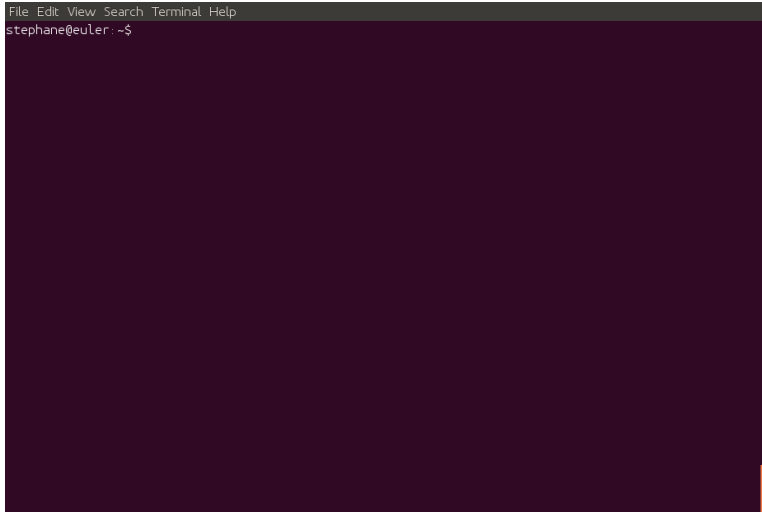
Apprendre à travailler

- Dossiers et sous-dossiers
- Le choix des noms
- Sauvegardes et versioning

Utiliser un IDE

- ipython
- Python for nerds
- Python for newbies

Les terminaux : pas que pour les geeks



```
File Edit View Search Terminal Help
stephane@euLer: ~$
```

```
File Edit View Search Terminal Help
```

Après le hard, le soft

Stéphane Gonnord

Plan

Système
d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque
dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

Sauvegardes et versioning

Utiliser un IDE

Logique globale

Python for nerds

Python for newbies

Fichiers : exécutables vs. documents

- ▶ **LibreOffice** : `titi.ods`, `toto.doc`, `tata.xls`, `tagada.txt`...
- ▶ **Word** : `toto.doc`, `tagada.txt`, ...
- ▶ **Chrome, Internet Explorer, Firefox, Safari** : `proutch.html`
- ▶ **L^AT_EX** : `ce_fichier.tex`, `ce_fichier.pdf`, `ce_fichier.log`...
- ▶ **WoW** : des tas de fichiers de configuration !
- ▶ `gcc` : `toto.c` (**source**) `toto.out` (**exécutable produit**)...
- ▶ **Idle, Spyder** : `tp-01.py`
- ▶ und so weiter.

Droits, groupes

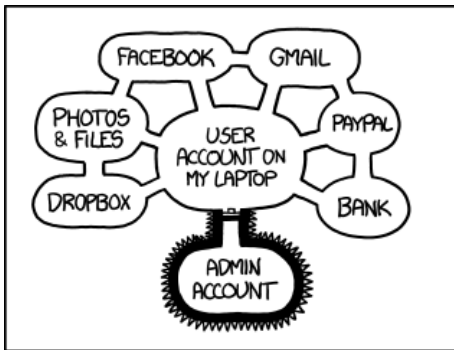
Les fichiers ont un *propriétaire*. Les utilisateurs des *droits* plus ou moins étendus.

- ▶ Droits :
 - ▶ écriture ;
 - ▶ lecture ;
 - ▶ exécution.
- ▶ Groupes :
 - ▶ root, superuser ;
 - ▶ utilisateur de base ;
 - ▶ groupe d'utilisateur plus ou moins vaste ;
 - ▶ tout le monde.
- ▶ Ces notions ont du sens :
 - ▶ sur un ordinateur personnel ;
 - ▶ sur un réseau (celui du lycée) ;
 - ▶ sur le cloud (dropbox, drive, ...) ;
 - ▶ mais aussi les réseaux sociaux, dans une certaine mesure.

Droits, groupes

Après le hard, le soft

Stéphane Gonnord



IF SOMEONE STEALS MY LAPTOP WHILE I'M
LOGGED IN, THEY CAN READ MY EMAIL, TAKE MY
MONEY, AND IMPERSONATE ME TO MY FRIENDS,
BUT AT LEAST THEY CAN'T INSTALL
DRIVERS WITHOUT MY PERMISSION.

Plan

Système
d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque
dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

Sauvegardes et versioning

Utiliser un IDE

Logique globale

Python for nerds

Python for newbies

Structure ni trop plate ni trop étirée

Après le hard, le soft

Stéphane Gonnord

Plan

Système
d'exploitation

Son rôle
Les OS au cours du temps

Contenu d'un disque
dur

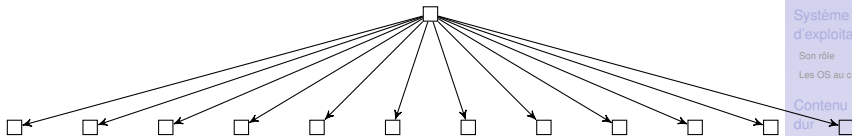
Quoi, où ?
Structure arborescente
Différents types de fichiers
Droits sur les fichiers

Apprendre à travailler

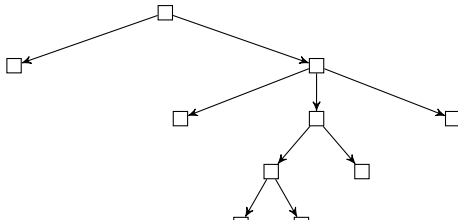
Dossiers et sous-dossiers
Le choix des noms
Sauvegardes et versioning

Utiliser un IDE

Logique globale
Python for nerds
Python for newbies



MAL !



Noms de fichiers

Plusieurs religions plus ou moins permissives.

Exemple de compromis :

- ▶ **Bien :**
 - ▶ `tp-1.py`
 - ▶ `tp_1.py`
 - ▶ `premier_tp_python_de_1_annee.py` (mais lourd)
- ▶ **Mal :**
 - ▶ `tp-1`
 - ▶ `tp 1.py`
 - ▶ `tépél.py`
 - ▶ `1er-tp.py`
 - ▶ `tp-1-avec-slash/ tsk-tsk.py`
 - ▶ `1 Super_Premier tépé! *&* .py`
- ▶ **Très mal :** `Documents/tp-1.py`

Noms de variables (identificateurs)

Plusieurs écoles/religions, pour chaque langage !

- ▶ Bien :
 - ▶ `debut`
 - ▶ `liste1`
 - ▶ `avec_underscore`
 - ▶ `PlutotAvecDesMajuscules`
 - ▶ `indice_debut_segment_recherche` (porte du sens)
- ▶ Mal (dont certains interdits) :
 - ▶ `début`
 - ▶ `l1`
 - ▶ `avec-tiret`
 - ▶ `lmauvais_debut`

Sauvegardes et versioning

On vous l'aura dit au moins une fois...

- ▶ Sauvegardes :
 - ▶ régulières ;
 - ▶ (partiellement) automatisées ;
 - ▶ (de temps en temps) délocalisées.
- ▶ Gestion des versions :
 - ▶ à la grand-papa (cpold) ;
 - ▶ professionnelle : arch, git, cvs, svn, mercurial...

Exemple concret : votre (futur) TIPE...

Logique globale

Le casting :

- ▶ Fichier Python (« script ») : `toto.py`
Instructions à exécuter successivement.
- ▶ Un interpréteur interactif (ou pas). Exécute les instructions d'un script, ou données à la volée.
- ▶ Une fenêtre d'exécution.

Plein de combinaisons possibles !

- ▶ Fichier exécuté « dans une fenêtre ».
- ▶ Pas de fichier ; des commandes dans l'interpréteur interactif (on ne va pas loin !).
- ▶ Un fichier Python interprété, avec ensuite des commandes dans la fenêtre interactive.

Plan

Système d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

Sauvegardes et versioning

Utiliser un IDE

Logique globale

Python for nerds

Python for newbies

Un fichier Python

```
"""  
des commentaires  
de début de fichier  
"""  
  
n = 5 # encore un commentaire  
  
# et encore ici  
  
for i in range(n):  
    print("7 * "+str(i)+" = "+str(7*i))
```

Après le hard, le soft

Stéphane Gonnord

Plan

Système
d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque
dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

Sauvegardes et versioning

Utiliser un IDE

Logique globale

Python for nerds

Python for newbies

Un fichier Python exécuté

```
stephane: /home/stephane/documents/info-pou
stephane@euler:~$ cd documents/info-pour-tous/cours-sup/02-elements-ordina
stephane@euler:~/documents/info-pour-tous/cours-sup/02-elements-ordina
stephane@euler:~/documents/info-pour-tous/cours-sup/02-elements-ordina
stephane@euler:~/documents/info-pour-tous/cours-sup/02-elements-ordina$ python2.7 jouet.py
File "jouet.py", line 4
SyntaxError: Non-ASCII character '\xc3' in file jouet.py on line 5, but n
o encoding declared; see http://www.python.org/peps/pep-0263.html for det
ails
stephane@euler:~/documents/info-pour-tous/cours-sup/02-elements-ordina$
stephane@euler:~/documents/info-pour-tous/cours-sup/02-elements-ordina$ python3.2 jouet.py
7 * 0 = 0
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
stephane@euler:~/documents/info-pour-tous/cours-sup/02-elements-ordina$
```

- ▶ Avec l'entête `# -*- coding: utf-8 -*-` Python 2.7 ne pose plus de problème.
- ▶ Exercice : qu'est-ce qui cloche dans ce terminal ? Comment ai-je réalisé ce tour de magie ?

À la rude : avec Emacs

Après le hard, le soft

Stéphane Gonnord

Plan

Système
d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque
dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

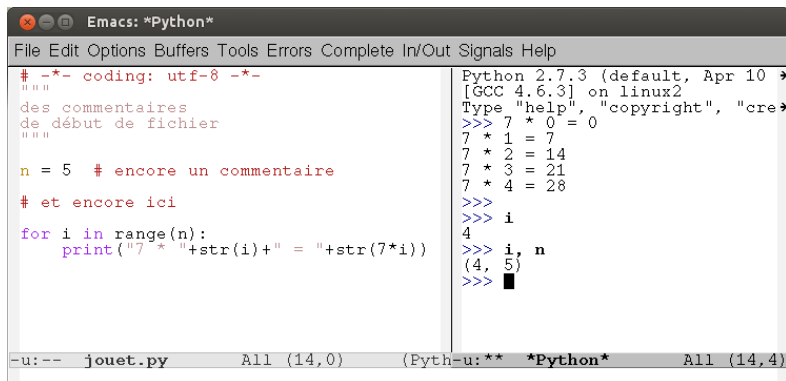
Sauvegardes et versioning

Utiliser un IDE

Logique globale

Python for nerds

Python for newbies



The screenshot shows the Emacs editor window titled "Emacs: *Python*". The menu bar includes "File Edit Options Buffers Tools Errors Complete In/Out Signals Help". The left pane contains Python code with comments in French. The right pane shows the Python interpreter's output, including version information and a list of powers of 7.

```
# -*- coding: utf-8 -*-
"""
des commentaires
de début de fichier
"""

n = 5 # encore un commentaire

# et encore ici

for i in range(n):
    print("7"*i +str(i)+" = "+str(7*i))
```

```
Python 2.7.3 (default, Apr 10 2010)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits() or "help()" to see full list.
>>> 7 * 0 = 0
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
>>>
>>> i
4
>>> i, n
(4, 5)
>>> █
```

At the bottom, the status bar shows: "-u:-- jouet.py All (14,0) (Pyth-u:** *Python* All (14,4))".

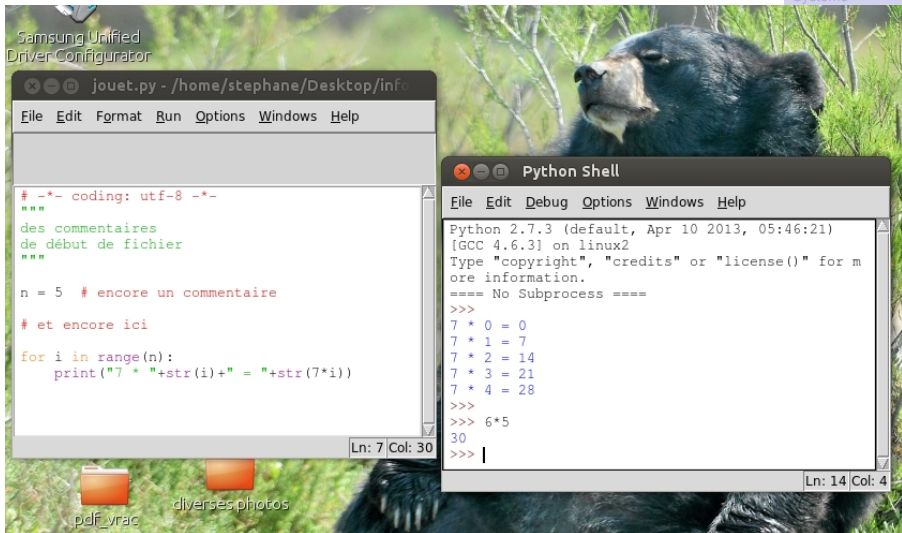
The screenshot displays the Spyder Python IDE interface. The main window is divided into three panes:

- Editor:** Contains a Python script named `jouet.py`. The code is as follows:

```
1# -*- coding: utf-8 -*-
2"""
3des commentaires
4de début de fichier
5"""
6
7n = 5 # encore un commentaire
8
9# et encore ici
10
11for i in range(n):
12    print("7 * "+str(i)+" = "+str(7*i))
13
```
- Console:** Shows the execution output of the script:

```
7 * 0 = 0
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
>>>
>>>
>>> 6 * 5
30
>>>
```
- Variable explorer:** A table showing the current state of variables in the environment:

Name	Type	Size	Value
i	int	1	4
n	int	1	5



End game

Merci de votre attention !

Après le hard, le soft

Stéphane Gonnord

Plan

Système d'exploitation

Son rôle

Les OS au cours du temps

Contenu d'un disque dur

Quoi, où ?

Structure arborescente

Différents types de fichiers

Droits sur les fichiers

Apprendre à travailler

Dossiers et sous-dossiers

Le choix des noms

Sauvegardes et versioning

Utiliser un IDE

Logique globale

Python for nerds

Python for newbies