

Nouveautés de Python 3.3



Pycon FR 2012, Paris

Victor Stinner
[<victor.stinner@gmail.com>](mailto:victor.stinner@gmail.com)

PEP 380 : yield from

```
>>> def g(x):  
...     yield from range(x, 0, -1)  
...     yield from range(x)  
...  
>>> list(g(5))  
[5, 4, 3, 2, 1, 0, 1, 2, 3, 4]
```

PEP 405 : venv

- Intégration de virtualenv
- Environnement Python avec ses modules privés
- L'environnement peut être dissocié des modules systèmes
- Module **venv** et programme **pyvenv**

Module lzma

- Plus rapide et compresse mieux que bzip2
- Compression plus gourmande en mémoire que bzip2
- Lecture/écriture des fichiers .xz et .lzma
- Somme de contrôle

Module ipaddress

- Adresses, réseaux et interfaces
- IPv4 et IPv6

```
>>> IPv4Address('127.0.0.2') + 3
IPv4Address('127.0.0.5')
>>> IPv4Address('192.0.2.6') in
IPv4Network('192.0.2.0/28')
True
```

Module faulthandler

- Traceback Python en cas de crash
- `python -X faulthandler ...`
- `PYTHONFAULTHANDLER=1 python ...`
- `faulthandler.dump_traceback()`
- `faulthandler.enable()`
- `faulthandler.register(signal.SIGUSR1)`
- `faulthandler.dump_tracebacks_later(60)`

PEP 3151 : OSError

- IOError, WindowsError et EnvironmentError sont des alias de OSError
- Ajout de FileNotFoundError (ENOENT), BlockingIOError (EAGAIN), InterruptedError (EINTR), ...
- Ajout de BrokenPipeError (EPIPE, ESHUTDOWN), ConnectionRefusedError (ECONNREFUSED), ...

importlib

- Système d'import réimplémenté en Python : module importlib
- Permet de personnaliser le système d'import (ex: import paresseux)
- Peut être réutilisé par les autres implémentations de Python (PyPy, Jython, IronPython)

PEP 409 : raise

```
def str2int(text):
    try:
        return int(text)
    except ValueError:
        raise Exception("!!!") from None
str2int("10x")
```

PEP 409 : raise

```
$ python pep409.py
Traceback (most recent call last):
  File "pep409.py", line 3, in str2int
    return int(text)
ValueError: invalid literal for int() with base 10:
'x'
```

During handling of the above exception, another exception occurred:

```
Traceback (most recent call last):
  File "pep409.py", line 7, in <module>
    str2int("x")
  File "pep409.py", line 5, in str2int
    raise Exception("!!!")
Exception: !!!
```

PEP 409 : raise

```
$ python pep409.py
```

```
Traceback (most recent call last):
  File "pep409.py", line 7, in <module>
    str2int("x")
  File "pep409.py", line 5, in str2int
    raise Exception("!!!!") from None
Exception: nombre invalide
```

PEP 3155 : __qualname__

```
>>> class C:  
...     def meth(self):  
...         pass  
>>> C.meth.__name__  
'meth'  
>>> C.meth.__qualname__  
'C.meth'
```

PEP 3155 : __qualname__

```
>>> class C:  
...     class D:  
...         ...  
...             def meth(self):  
...                 pass  
...  
>>> C.D.__qualname__  
'C.D'  
>>> C.D.meth.__qualname__  
'C.D.meth'  
>>> C.D.meth.__name__  
'meth'
```

PEP 420 : packages

- Un module n'a plus besoin de fichier `__init__.py` à la racine
- Un package peut être réparti dans plusieurs dossiers :
 - `module.a` : dossier1/module/a.py
 - `module.b` : dossier2/module/b.py

PEP 418 : time

- **get_clock_info(name)** : ajustable, implementation, monotonic, resolution
- **monotonic()** : non affecté par les changements d'horloge système
- **perf_counter()** : très précis (sur une courte durée)
- **process_time()** : temps CPU du processus courant

PEP 393 : Unicode

- Python 2 : "abc" utilise 3 octets (données)
- Python 3.2 : "abc" utilise 6 ou 12 octets
- Python 3.3 : "abc" utilise 3 octets
- Stockage compact en conservant une complexité de $O(1)$ pour `str[index]`
- `len("\U0010ffff")` donne toujours 1
- Codec UTF-8 : 2x à 4x plus rapide

PEP 412 : dict

- Le nom des attributs (clés de `obj.__dict__`) sont partagés entre les objets
- Baisse de 10% à 20% de la mémoire pour des applications orientées objet

Sécurité

- hash() aléatoire par défaut
- crypt.crypt() : ajout de sel
- ssl.RAND_bytes()
- hmac.compare_digest() protège des timing attacks
- shutil.rmtree() protégé contre les symlink attacks

Module os

- `sendfile()` : écriture sans copie entre un fichier et une socket réseau
- Accès au système de fichier par descripteur de fichier, protège des "symlink attacks" : `os.openat()`, `os.flistdir()`, ...
- Attributs étendus (SELinux) : `os.getxattr()`, ...

Module shutil



- chown(str, str)
- disk_usage()
- get_terminal_size()

Module signal

- pthread_sigmask(SIG_BLOCK, signals)
- pthread_kill(thread_id)
- sigtimedwait(signals, timeout)
- sys.thread_info : name, lock, version

PEP 3118 : buffer

- Amélioration notable du « protocole » buffer (memoryview)
- Gestion explicite de la durée de vie d'un buffer : corrige des crashes
- hash(buffer)
- buffer[::-1]

Module decimal

- 12x à 120x plus rapide
- Arrondit correct sur $\exp(x)$, $\ln(x)$ et $x^{**} y$

PEP 362 : signature

- Format standard et simple pour lire la signature d'une fonction
- `inspect.signature(func)`

Builtins

- `open("fichier.txt", "x")` crée un fichier ou échoue s'il existe déjà (atomique)
- `print("affiche moi ça immédiatement !", flush=True)`
- `'B'.casefold()` donne 'ss', utile pour la recherche ignorant la casse (`'B'.lower()=='ss'`)

Unicode

- `u"unicode"` est à nouveau accepté
- `base64.b64decode()` accepte une chaîne de caractère ASCII
- Ajout du codec "`cp65001`" sous Windows
(console en UTF-8)
- Le module `curses` gère mieux Unicode

Concurrency

- `subprocess.Popen.communicate(timeout)`
- `threading.Lock.acquire(timeout)`
- `multiprocessing.connection.wait()` permet d'attendre plusieurs objets

Vrac

- `math.log2(x)` plus précis que `math.log(x, 2)`
- `datetime.timestamp()`, opposé de `datetime.fromtimestamp()`
- socket : `sendmsg()` et `recvmsg()` pour envoyer-recevoir un descripteur de fichier

Questions & Discussion



Contact:

victor.stinner@gmail.com

Merci à David Malcom pour le
modèle LibreOffice

<http://dmalcolm.livejournal.com/>