Python
Les logs en python

Nicolas Delestre

INSA
ROUEN NORMANDIE

Les logs

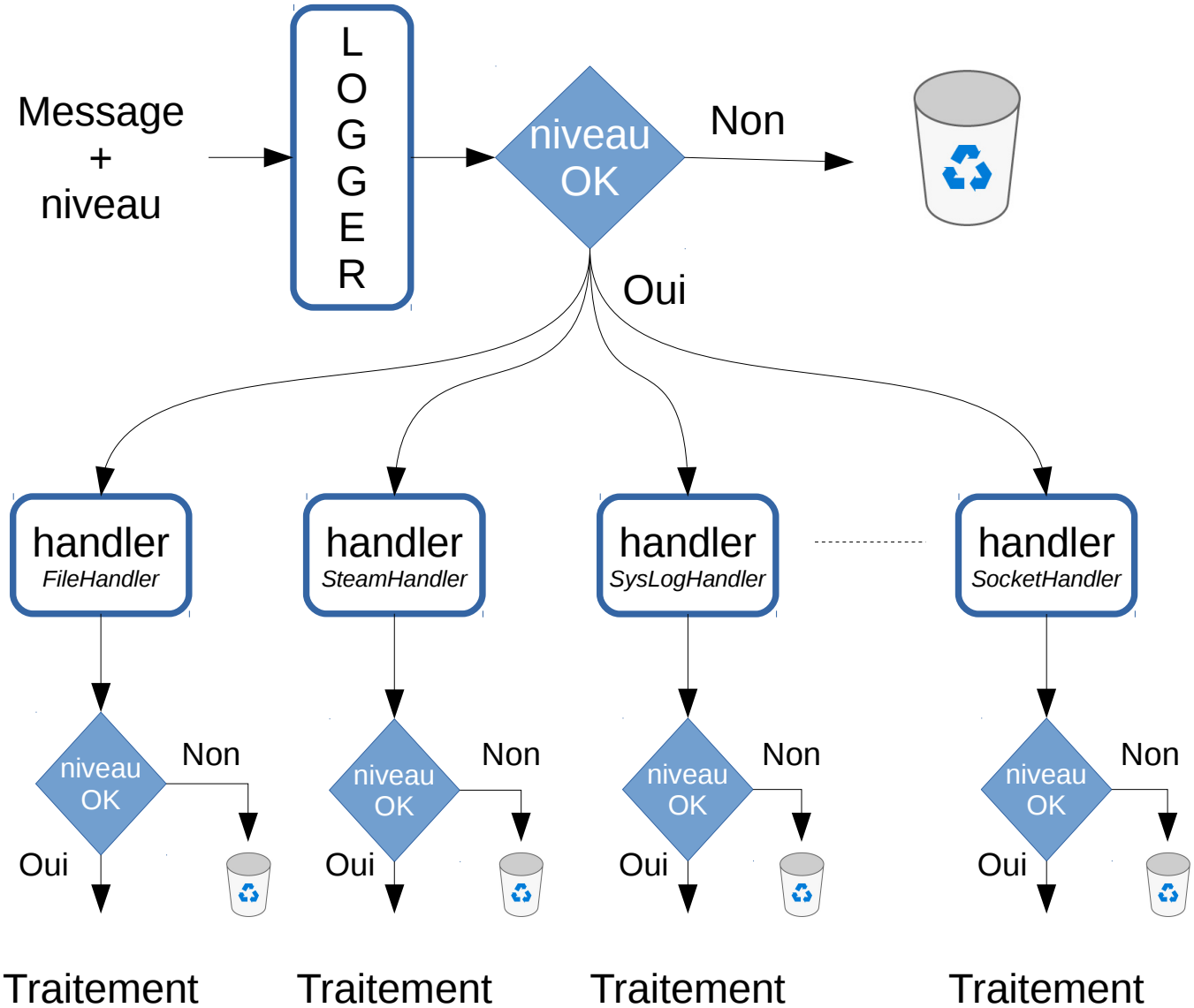
Lorsque l'on veut débugger un programme...

- On utilise souvent `print`, mais ce n'est pas satisfaisant car :
  - les informations sont envoyées sur la sortie standard
  - on mélange information sur le fonctionnement du programme et utilisation normal de la sortie standard
  - il n'y a pas forcément de sortie standard (programme python utilisé comme démon)
  - les messages ne sont pas formatés
  - on ne sait pas distinguer ce qui est important de ce qui ne l'est pas

logging

- `logging` est un framework standard de python
- il s'inspire de Log4j du monde Java (identifiants en *CamelCase* :-( )
- il permet de *formater* les messages
- il permet de conditionner les messages
- il permet d'envoyer les messages sur un ou plusieurs flux (*handler*)

Fonctionnement de `logging`



inspiré de *http://sametmax.com/ecrire-des-logs-en-python/*

Les niveaux

- Il y a 5 niveaux : `DEBUG` < `INFO` < `WARNING` < `ERROR` < `CRITICAL`
- Un niveau est associé :
  - à un message
  - au *logger*
  - aux *handler*
- Le message est généré dans le flux associé à un *handler* si son niveau est supérieur ou égal au niveau du *logger* **et** du *handler*

INSA ROUEN NORMANDIE

Utilisation d'un *logger* dans votre code

Étapes

1. Importer le module `logging`
2. Obtenir le *logger* : fonction `getLogger`
3. Utiliser la méthode :
   - `log` avec comme premier paramètre effectif le niveau de log et comme deuxième le message
   - `debug`, `info`, `warning`, `error` ou `critical` avec comme paramètre effectif le message

INSA ROUEN NORMANDIE

Paramétrage du logger et création des *handler*

- À réaliser dans votre programme principal en fonction :
  - de paramètres/options données à l'exécution du programme
  - d'un fichier de configuration

Étapes

1. Importer le module `logging`
2. Obtenir le *logger* : fonction `getLogger`
3. Fixer le niveau de logger : méthode `setLevel` avec les constantes du module `logging`
4. Créer un *handler*, plusieurs classes disponibles : `StreamHandler` (sortie erreur standard), `FileHandler` (fichier de log), etc.
5. Paramétrer le *handler* : `setLevel` (niveau), `setFormatter` (formatage), etc.
6. Ajouter le *handler* au *logger* : méthode `addHandler`

Les étapes 4 à 6 peuvent être répétées plusieurs fois

Exemple 1 / 11

Des logs pour notre application sur les polylignes

On va utiliser trois niveaux de log :

- `debug` pour les créations et modification
- `info` pour les getter
- `error` lors des levées d'exception

`polyligne.py`

```
4 import logging
5
6 logger = logging.getLogger()
```

INSA ROUEN NORMANDIE

Exemple 2 / 11

Extrait de polyligne.py

```
12      @staticmethod
13      def _verifier_pt_ou_pts_absent(pt_ou_pts, pts):
14          try:
15              iter(pt_ou_pts)
16          except Exception:
17              if pt_ou_pts in pts:
18                  logger.error("Exception MemePointInterditErreur %s appartient déjà  à
    %s" % (pt_ou_pts, self))
19                  raise MemePointInterditErreur(repr(pt_ou_pts) + " appartient déja à
    la polyligne")
20              return
21          for pt in pt_ou_pts:
22              if pt in pts:
23                  logger.error("Exception MemePointInterditErreur %s appartient déjà  à
    %s" % (pt_ou_pts, self))
24                  raise MemePointInterditErreur(repr(pt) + " appartient déja à la polyligne")
```

INSA ROUEN NORMANDIE

Exemple 3 / 11

Extrait de polyligne.py

```
12    def __init__(self, est_ferme, pt1, pt2, *args):
13        logger.debug("Creation d'une polyligne : est_ferme : %s, points : %s" % (est_ferme,
14                                                                                  [pt1, pt2]))
15        if pt1 == pt2:
16            raise MemePointInterditErreur("pt1 et pt2 doivent être différents")
17        self._est_ferme = est_ferme
18        self._points = [pt1, pt2]
19        for pt in args:
20            self.ajouter(pt)
```

```
36    def _get_est_ferme(self):
37        logger.info("appel à est_ferme")
38        return self._est_ferme
39
40    def _set_est_ferme(self, est_ferme):
41        logger.debug("%s de la polyligne %s" % ("Fermeture" if est_ferme else "Ouverture",
    self))
42        self._est_ferme = est_ferme
43
44    est_ferme = property(_get_est_ferme, _set_est_ferme)
```

Exemple 4 / 11

Extrait de polyligne.py

```
77      def __iter__(self):
78          logger.info("Début d'une itération")
79          for pt in self._points:
80              logger.info("obtention de %s dans une itération" % pt)
81              yield pt
```

Un script qui utilise Polyligne : exemple_sans_logging.py

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

if __name__ == "__main__":
    from polyligne import Polyligne
    from point import Point2D

    pl = Polyligne(True, Point2D(0,0), Point2D(1,1), Point2D(2,0))
    for pt in pl:
        print(pt)
```

Résultats

```
$ python3 exemple_sans_logging.py
(0, 0)
(1, 1)
(2, 0)
```

INSA ROUEN NORMANDIE

Exemple 6 / 11

logging_sortie_erreur_standard.py

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

import logging

logger = logging.getLogger()

sortie_standard = logging.StreamHandler()
sortie_standard.setLevel(logging.INFO)
logger.addHandler(sortie_standard)
```

INSA ROUEN NORMANDIE

exemple_logging_un_seul_handler.py

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

if __name__ == "__main__":
    from polyligne import Polyligne
    from point import Point2D

    import logging
    import logging_sortie_erreur_standard

    logger = logging.getLogger()
    logger.setLevel(logging.DEBUG)

    pl = Polyligne(True, Point2D(0,0), Point2D(1,1), Point2D(2,0))
    for pt in pl:
        print(pt)
```

Exemple 8 / 11

Résultats

```
$ python3 exemple_logging_un_seul_handler.py
Début d'une itération
obtention de (0, 0) dans une itération
obtention de (1, 1) dans une itération
Début d'une itération
obtention de (0, 0) dans une itération
(0, 0)
obtention de (1, 1) dans une itération
(1, 1)
obtention de (2, 0) dans une itération
(2, 0)
```

Résultats en redirigeant le flux d'erreur standard

```
$ python3 exemple_logging_un_seul_handler.py 2> sortie_erreur_standard.txt
(0, 0)
(1, 1)
(2, 0)
```

ROUEN NORMANDIE

logging_fichier_temp.py

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

import logging
import __main__ as main

logger = logging.getLogger()

formatter = logging.Formatter('%(asctime)s :: %(levelname)s :: %(message)s')

fichier_temp = logging.FileHandler("/tmp/" + main.__file__[:-3] + ".log")
fichier_temp.setLevel(logging.DEBUG)
fichier_temp.setFormatter(formatter)
logger.addHandler(fichier_temp)
```

INSA ROUEN NORMANDIE

exemple_logging_deux_handlers.py

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

if __name__ == "__main__":
    from polyligne import Polyligne
    from point import Point2D

    import logging
    import logging_sortie_erreur_standard
    import logging_fichier_temp

    logger = logging.getLogger()
    logger.setLevel(logging.DEBUG)

    pl = Polyligne(True, Point2D(0,0), Point2D(1,1), Point2D(2,0))
    for pt in pl:
        print(pt)
```

Exemple 11 / 11

Résultats

```
$ python3 exemple_logging_deux_handlers.py
Début d'une itération
obtention de (0, 0) dans une itération
obtention de (1, 1) dans une itération
Début d'une itération
obtention de (0, 0) dans une itération
(0, 0)
obtention de (1, 1) dans une itération
(1, 1)
obtention de (2, 0) dans une itération
(2, 0)
$ cat /tmp/exemple_logging_deux_handlers.log
2018-04-04 08:39:12,461 :: DEBUG :: Creation d'une polyligne : est_ferme : True, points : [Point2D(0,0), Point
2018-04-04 08:39:12,462 :: INFO :: Début d'une itération
2018-04-04 08:39:12,462 :: INFO :: obtention de (0, 0) dans une itération
2018-04-04 08:39:12,462 :: INFO :: obtention de (1, 1) dans une itération
2018-04-04 08:39:12,462 :: DEBUG :: Ajout des points [Point2D(2,0)] à la polyligne ((0, 0), (1, 1))
2018-04-04 08:39:12,463 :: INFO :: Début d'une itération
2018-04-04 08:39:12,463 :: INFO :: obtention de (0, 0) dans une itération
2018-04-04 08:39:12,463 :: INFO :: obtention de (1, 1) dans une itération
2018-04-04 08:39:12,463 :: INFO :: obtention de (2, 0) dans une itération
```

INSA ROUEN NORMANDIE