

Let's hack everything, everywhere, everyone.

[DailyHckr](#)

- [DailyHckrについて](#)

前の記事: [日本語リファレンスには書いてない話: urllibとurllib2の違いってなんだ](#)

次の記事: [Pythonで被はてなブックマーク数を取得する](#)

Pythonのwith構文と __enter__ 、 __exit__

[\[programming\]](#) [\[python\]](#)

2010年5月18日 20:27

Pythonのwith構文がいまいち掴めなかったので、ざっとまとめてみた(いまさらながら)。ドキュメントは[Python リファレンスマニュアルの7.5 with 文](#)にある。

withを使ったコード例は、下記のような感じ。

```
c = ClassHogeHoge()
with c:
    c.foobar()
```

上記のコードは、下記と等価となる。

```
c = ClassHogeHoge()
c.__enter__()
c.foobar()
c.__exit__()
```

つまり、withに続くインデントブロックを実行する前に指定したオブジェクトの「__enter__()」メソッドを呼び出し、実行後に「__exit__()」メソッドが暗に呼び出される、という仕組み。

__enter__()と__exit__()の定義は、[Python リファレンスマニュアルの3.4.9 with文とコンテキストマネージャ](#)にある。__enter__()の引数はselfのみだが、__exit__()はself、exc_type、exc_value、tracebackの4つの引数をとる。withに続くインデントブロックが正常に実行された(つまり、例外が送出されなかった)場合、(self以外の)引数にはNoneが与えられる。なにか例外が発生した場合、その例外に関する情報が与えられるらしい。

また、「with hogehoge as foo:」のような形でwith文を利用する場合、__enter__()の戻り値がfooに代入される。__exit__()の戻り値は例外処理の伝搬制御に使われ、Falseの場合例外が発生した場合でも例外を伝搬させず、Trueを返すと例外が伝搬されとのこと。

下記、使用例。

```
class CachedDB(object):
    DB_FILE = "database/db_dat"

    def __init__(self):
        self.con = None
        self.cur = None

    def __enter__(self):
        self.con = sqlite3.connect(self.DB_FILE)
        self.cur = self.con.cursor()

    def __exit__(self, exc_type, exc_value, traceback):
        if exc_type:
            self.con = None
            self.cur = None
```

```

        return False
    self.con.commit()
    self.cur.close()
    self.con.close()
    self.con = None
    self.cur = None
    return True

def add(self, foo, bar, hoge):
    try:
        self.cur.execute("""insert into data ( foo, bar, hoge )
                           values (?, ?, ?);""", (foo, bar, hoge))

def main():
    usage = "%s logfile" % sys.argv[0]
    db = CachedDB()
    try:
        fname = args[0]
    except IndexError:
        sys.exit(usage)

    f = open(fname, "r")
    with db:
        for l in f:
            term = l.strip().decode("utf-8").rsplit("\t", 3)
            db.add(foo=term[0],
                  bar=term[1],
                  hoge=term[2])

if __name__ == '__main__':
    main()

```

タブ区切りのデータファイルを1行ずつ読んでデータベースに突込む、という処理。withを使うことで、データベースアクセスの準備→データ挿入→コミットという流れをきれいに実装できました。

[permalink](#) | [コメントを付ける](#) | [コメントRSS](#) | [トラックバックURL](#)

コメント:

お名前(必須):

メールアドレス(公開されません、必須):

URL:

コメント本文:

[illegible]

利用可能なタグ: <abbr title=""> <acronym title="">
<blockquote cite=""> <cite> <code> <del datetime=""> <i> <q cite=""> <s>
<strike>

コメントを投稿

前の記事: [日本語リファレンスには書いてない話: urllibとurllib2の違いってなんだ](#)

次の記事:[Pythonで被はてなブックマーク数を取得する](#)

DailyHackrナビゲーション