

Python

[[サンプルコード](#)][[まとめ](#)][[標準ライブラリ](#)][[ライブラリ](#)][[Django](#)][[Flask](#)][[Pyramid](#)][[Zope](#)][[IronPython](#)][[Google App Engine](#)][[Pillow](#)][[Python Imaging Library](#)][[Beautiful Soup](#)][[Python NumPy](#)][[Python matplotlib](#)][[Python Win32 Extensions](#)][[pyExceleator](#)][[Jython](#)][[PyDev](#)][[PyQt](#)][[PyScripter](#)]

■インストール

- ・ [CentOS6 に Python2.7 をインストール](#)
- ・ [Django インストール](#)

■言語まとめ

- ・ [言語まとめ](#)

■サンプルコード (ファイル、コレクション・・・)

- ・ [サンプルコード \(ファイル、コレクション・・・ \)](#)
- ・ [サンプルコード \(CookBook\)](#)

■サンプルコード (オブジェクト指向)

- ・ [サンプルコード \(オブジェクト指向 \)](#)

■標準ライブラリ概観

詳細

- ・ [標準ライブラリ概観](#)

概要

処理内容	標準ライブラリ
OS インターフェース	os
ファイルのワイルドカード	glob
コマンドライン引数	sys
標準エラーを出力しプログラムを終了	sys
正規表現 を利用する	re
数学	math
URL による任意のリソースへのアクセス	urllib
インターネットアクセス	urllib2
日付と時刻	date 、 datetime 、 time
時間計算	time

データ圧縮	zlib
パフォーマンス計測	timeit
品質管理	doctest
出力書式	repr 、 pprint 、 textwrap 、 locale
テンプレート	Template
シリアライズ	pickle
バイナリデータレコードレイアウトでの作業	struct
オブジェクトのコピー	copy
マルチスレッド	threading
ロギング	logging
XML Dom	xml.dom
XML ElementTree	xml.etree
データベースの使用	sqlite3

■ビルトインオブジェクト

- ・ [ビルトインオブジェクト](#)

■ Python サンプルコード

全般

- ・ [サンプルコード \(ファイル、コレクション・・・ \)](#)
- ・ [サンプルコード \(オブジェクト指向 \)](#)
- ・ [組込関数](#)

サンプル

- ・ [Python ファイル読み書き](#)
- ・ [Python ゼロ埋め](#)
- ・ [Python 3 項演算子](#)
- ・ [Python 環境変数 PATH からファイル検索](#)
- ・ [Python ファイルを削除](#)
- ・ [Python 数値の桁数を得る](#)

■ Python Osmosis

- ・ [Episode 6: Flow Control, if and for Statements](#)
- ・ [Episode 7: The Range Function](#)
- ・ [Episode 50: Standard Library: Decimal Floating Point Arithmetic](#)

IDE

- ・ [Python IDE にて矢印キーでヒストリー表示](#)
- ・ [Python IDE を CentOS で利用する](#)

パッケージ管理

setuptools

- ・ <http://peak.telecommunity.com/DevCenter/setuptools>
- ・ <https://pypi.python.org/pypi/setuptools>
- ・ [CentOS6 に Python2.7 をインストール](#)

■ Windows

上記ページからインストーラをダウンロードして実行 (2.7 の例)

`setuptools-0.6c11.win32-py2.7.exe`

Power Shell 3 以降

PS C:\WINDOWS\system32> (Invoke-WebRequest https://bootstrap.pypa.io/ez_setup.py).Content | python -

PATH に追加 (2.7 の例)

C:\Python27\Scripts

実行例

c:\work\python>easy_install virtualenv

プロキシを利用している場合、以下を環境変数に設定

`HTTP_PROXY=http://` プロキシサーバ: ポート

■ Windows PIP を利用する

- ・ [PIP](#)
- ・ [Python Windows に PIP をインストール](#)
- ・ [Unofficial Windows Binaries for Python Extension Packages](#)

■ Windows 用 バイナリパッケージ

- ・ <http://www.lfd.uci.edu/~gohlke/pythonlibs/>
- ・ [Windows で VirtualEnv の Python2.7 に pip と wheel を使って コンパイルエラーが発生するパッケージ \(例 scipy\) を Windows 用バイナリ提供サイトから入手してインストールする](#)

仮想環境

VirtualEnv を利用して Python の仮想環境を作成

Python 3 以降同梱されている

- ・ <https://docs.python.jp/3/library/venv.html>

```
python3 -m venv /path/to/new/virtual/environment
```

それ以前

■ CentOS

- ・ [Virtualenv のインストールと起動](#)

■ Windows

インストール

上記 パッケージ管理 を参照し、easy_install を利用出来るようにしておく

```
c:\easy_install virtualenv
```

ワークスペースの作成

```
c:\work\python>python -m virtualenv env
New python executable in env\Scripts\python.exe
Installing setuptools.....done.
Installing pip.....done.
```

仮想環境の実行

- ・ Scripts\activate の実行

```
c:\work\python>cd env\Scripts
```

```
c:\work\python\env\Scripts>activate
(env) c:\work\python\env\Scripts>
```

仮想環境の停止

- ・ Scripts\deactivate の実行

```
(env) c:\work\python\env\Scripts>deactivate
```

Tcl ライブラリが見つからない

- ・ <http://typea.info/blg/glob/2015/07/python-windows-virtualenv-tcl.html>

Python VirtualEnv で IDLE を利用する

- ・ <http://typea.info/blg/glob/2015/09/python-virtualenvidle.html>

■ Ubuntu

インストール

```
$ sudo apt-get install python-virtualenv
```

WinPython

- ・ [WinPython](#)

フレームワーク

Web

■ Django

- ・ [Django](#)

■ Google App Engine

- ・ [Google App Engine](#)

O/R マッパー

■ SQLAlchemy

- ・ [SQLAlchemy](#)
- ・ [SQLAlchemy 0.6.5 ドキュメント \(和訳\)](#)

プラグイン

■ PyDev

- ・ [PyDev Eclipse プラグイン](#)

Python ライブラリ

- ・ [Python ライブラリ](#)

デバッグ

- ・ PyDb <http://bashdb.sourceforge.net/pydb/>

GUI

- ・ [PyQt](#)

実践

自然言語処理

- ・ [自然言語処理](#)
- ・ [Python NLTK\(Natural Language Toolkit\)](#)
- ・ [Python MeCab\(日本語形態素解析\)](#)

Tips

文字コード

■ ファイルの文字コード

■ URL エンコード

■ Python UnicodeEncodeError の対処

■ Python ユニコードエスケープをデコード

■ Python CSV ファイル

書式

■ 文字列を 16 進数に変換

```
print 'hoge'.encode('hex')
print '686f6765'.decode('hex')
```

■ 2 進数文字列を 10 進数に変換

```
>>> int('10101100',2)
172
```

■ IP アドレスを 2 進数で表示

```
>>> "{0:08b}.{1:08b}.{2:08b}.{3:08b}".format(172,31,16,0)
'10101100.00011111.00010000.00000000'
```

- ・ 戻す

```
>>> ' '.join([str(int(x,2)) for x in "10101100.00011111.00010000.00100000".split(".")])
'172.31.16.32'
```

ログ

- ・ [Python 任意のファイルにログを出力する](#)

サンプル

■ Python 組合せと順列の計算

■ Python パスワード生成

その他

■ help 関数の使い方

- ・ [Python help 関数の使い方](#)

■ プロパティを持っているか

`hasattr( インスタンス , プロパティ名 )`

■ 正規表現による分割

- ・ `re.split` を利用

```
>>> import re
>>> re.split('[\t\n\.\,]', 'this is\ta\npen.')
['this', 'is', 'a', 'pen', '']
```

■ 正規表現による置換

- ・ `re.sub(pattern, repl, string, count=0, flags=0)` を利用する

```
>>> s = r'AaBbCc'
>>> re.sub(r'[a-z]', r'', s)
'ABC'
```

■ ユーザーからの入力を得る

- ・ `raw_input` を利用する

```
number = int(raw_input("please input int:"))
```

日本語のクエリパラメータ

```
import urllib
import urllib2
url = r'https://maps.googleapis.com/maps/api/place/nearbysearch/json?%s'
queries = {
    'key':GOOGLE_API_SERVER_KEY,
    'location':'35.1814464,136.906398',
    'radius':'500',
    'keyword':u'まんが喫茶 '.encode('utf-8')
}
data = urllib.urlencode(queries)
req = url % data
print req
response = urllib2.urlopen(req)
for l in response:
    print l
```

文字化け

- ・ [Python 対話シェルへユニコードを文字として表示する](#)

- PyScripter 日本語ファイルエラーの対応
- Python URL エンコード
- Python UnicodeEncodeError の対処
- Python ファイルの文字コード
- Python ユニコードエスケープをデコード
- Python 対話シェルへユニコードを文字として表示する