

CONTENTS

ビジュアルヘルプ - PythonEnv	2
PythonEnv コマンドのヘルプ	2

PythonEnv コマンドのヘルプ

PythonEnv [/P=pathName /Z[=z]] [activate=envPathStr, venv=argumentStr]

PythonEnv コマンドは、Python 環境の基本的な環境管理機能を提供します。

一度に指定できるキーワードは1つだけです。

キーワードが指定されていない場合、PythonEnv は、現在アクティブな Python 環境に関する情報を含む文字列変数 `S_PythonEnvInfo` を作成し、Python インタプリターが現在実行中の場合、`V_PythonRunning` を 1 に設定します。

フラグ

`/P=pathName` 作業ディレクトリとして使うフォルダーを指定します。
`pathName` は、既存のシンボリックパス名です。

`/Z[=z]` エラーの発生を抑制します。
エラーを自分で処理したい場合に使ってください。

`z=0:` エラーを抑制します。`/Z` を省略した場合、これがデフォルト設定です。

`z=1:` Igor によって生成されたエラーが、実行を停止しません。エラーが発生した場合、エラーコードは `V_Flag` に格納されます。

パラメーター

`activate= envPathStr`

`envPathStr` で指定された完全パスまたは相対パスにある Python 環境を起動します。
`envPathStr` が相対パスである場合、そのパスは `/P` フラグで指定されたシンボリックパスを基準として相対パスとして解釈されます。

`/P` フラグが省略された場合は、ユーザーのホームディレクトリ (例: `C:/Users/<username>`) を基準として相対パスとして解釈されます。

環境を有効化することは、メニュー `Miscellaneous Settings` → `Python` の環境設定を変更することと同一です。

有効化された環境に関する情報は、出力文字列変数 `S_PythonEnvInfo` に格納されています。

`venv=argumentStr`

`venv` パッケージを使って新しい仮想 Python 環境を作成します。

`argumentStr` は、ドキュメント (<https://docs.python.org/3/library/venv.html>) に従った有効な `venv` 引数です。

`/P` フラグが指定された場合、そのパスが `venv` を実行する時の作業ディレクトリとして使われます。

その結果、`argumentStr` にフルパスが指定されていない場合、そのディレクトリに仮想環境が作成されます。

仮想環境で使われる Python のバージョンは、現在アクティブな Python 環境のベースの Python バージョンとなります。

新しく作成した仮想環境は、使う前に `activate` キーワードを使ってアクティブ化するか、または `Miscellaneous Settings` → `Python` で選択する必要があることに注意してください。

作成された仮想環境に関する情報は、出力文字列変数 S_PythonEnvInfo に格納されています。

出力変数

S_PythonEnvInfo

この文字列変数には、現在アクティブな Python 環境または直前に作成された環境に関するキーと値のリストが含まれます。

これは、使われたキーワードによって異なります。

V_PythonRunning

この数値変数は、Python インタプリターが現在実行中の場合、1 に設定され、そうでない場合は 0 に設定されます。

V_Flag

コマンドが成功した場合に 0 に設定し、エラーが発生した場合は 0 以外のエラーコードを設定します。

S_PythonEnvInfo のキーワード-値ペア :

<u>キーワード</u>	<u>値</u>
NAME	環境名 (ある場合)。 ベースの Python インストールには環境名はありません。
VERSION	Python のバージョン番号。
HOME	Python ホームフォルダーへのフルパス。
LIBRARY	環境によって使われる Python ライブラリーへのフルパス。
SITEPACKAGES	site-packages フォルダーへのフルパス。

新しい仮想環境の作成

仮想環境を作成する簡単な例 :

```
// 新しい仮想環境は、C:/MyProject/imageAnalysis として作成されます
NewPath/O environmentPath, "C:/MyProject"
PythonEnv/P = environmentPath venv = "imageAnalysis"
```

または、環境を作成するために完全なパスを使うこともできます :

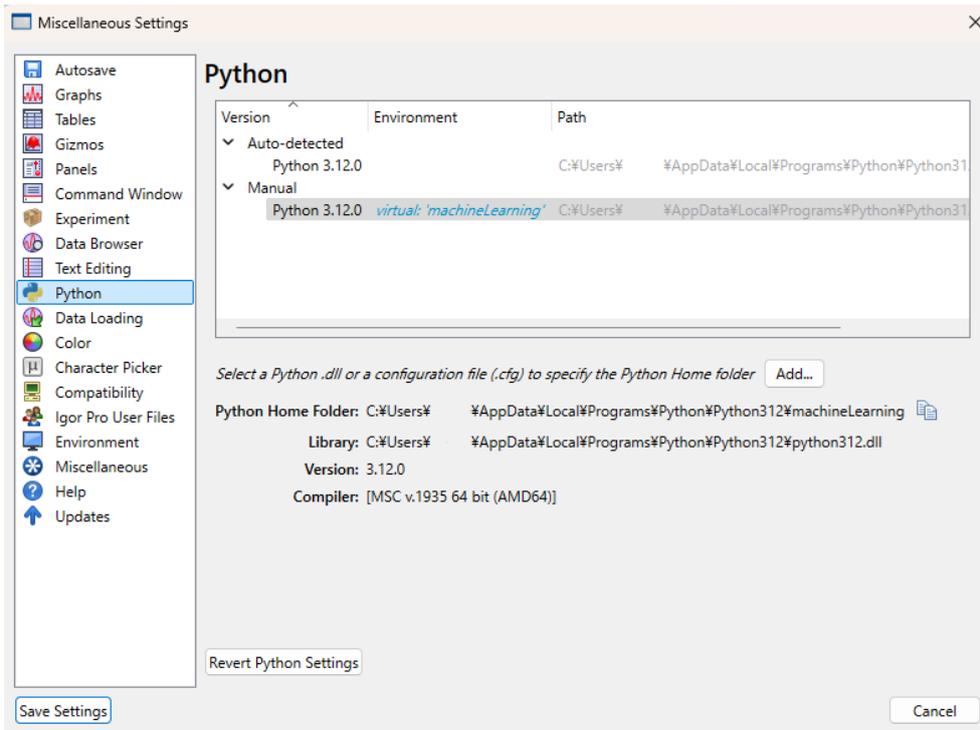
```
PythonEnv venv = "C:/MyProject/imageAnalysis"
```

venv の使用方法については、<https://docs.python.org/3/library/venv.html> を参照するか、Igor 内でヘルプドキュメントを表示するには、以下を実行してください。

```
PythonEnv venv = "-h"
```

新しい可能環境を有効化

環境を作成すると、その環境は手動で設定した環境のリストに自動的に追加されます。このリストは、Miscellaneous Settings の Python カテゴリで確認できます。



環境の設定が完了したら、Miscellaneous Settings の Python カテゴリで選択して有効にするか、PythonEnv コマンドで activate キーワードを使います。

```
PythonEnv/P = environmentPath activate = "imageAnalysis"
```

または、環境を有効にするために完全なパスを使うこともできます：

```
PythonEnv activate = "C:/MyProject/imageAnalysis"
```

Igor 以外のツールを使って仮想環境を作成した場合でも、その環境をアクティブ化することができます。その場合は、Igor の手動で設定された環境のリストに自動的に追加されます。

環境が有効化されている場合、以降の Python または PythonFile コマンドへの呼び出し、または Python コンソールの使用は、この環境を使って実行されます。

ただし、Igor から Python コードが一度実行されると、Igor を再起動するまで新しい環境の有効化は反映されません。

参照

Python コマンド、PythonFile コマンド、ヘルプ Python Console