

Utilities

example

`example(cls, dur=5, toprint=True, double=False)`

[\[source\]](#)

Execute the documentation example of the object given as an argument.

Args: cls: PyoObject class

Class reference of the desired object example.

dur: float, optional

Duration of the example.

toprint: boolean, optional

If True, the example script will be printed to the console. Defaults to True.

double: boolean, optional

If True, force the example to run in double precision (64-bit) Defaults to False.

```
>>> example(Sine)
```

class_args

`class_args(cls)`

[\[source\]](#)

Returns the init line of a class reference.

This function takes a class reference (not an instance of that class) as input and returns the init line of that class with the default values.

Args: cls: PyoObject class

Class reference of the desired object's init line.

```
>>> print(class_args(Sine))
>>> 'Sine(freq=1000, phase=0, mul=1, add=0)'
```

getVersion

`getVersion()`

[\[source\]](#)

Returns the version number of the current pyo installation.

This function returns the version number of the current pyo installation as a 3-ints tuple (major, minor, rev).

The returned tuple for version '0.4.1' will look like: (0, 4, 1)

```
>>> print(getVersion())  
>>> (0, 5, 1)
```

getPyoKeywords

getPyoKeywords()

[\[source\]](#)

Returns a list of every keywords (classes and functions) of pyo.

```
>>> keywords = getPyoKeywords()
```

withPortaudio

withPortaudio()

Returns True if pyo is built with portaudio support.

withPortmidi

withPortmidi()

Returns True if pyo is built with portmidi support.

withJack

withJack()

Returns True if pyo is built with jack support.

withCoreaudio

withCoreaudio()

Returns True if pyo is built with coreaudio support.

withOSC

withOSC()

Returns True if pyo is built with OSC (Open Sound Control) support.

convertStringToSysEncoding

convertStringToSysEncoding(*strng*)

[\[source\]](#)

Convert a string to the current platform file system encoding.

Returns the new encoded string.

Args: *strng*: string

String to convert.

convertArgsToLists

convertArgsToLists(**args*)

[\[source\]](#)

Convert all arguments to list if not already a list or a PyoObjectBase. Return new args and maximum list length.

wrap

wrap(*arg*, *i*)

[\[source\]](#)

Return value at position *i* from *arg* with wrap around *arg* length.