



Advanced Collections

Objectives

We will write a generator function using a custom built module, and use a dictionary comprehension. Finally we consider the occasions when a deep copy is required.

Reference Material

Chapter 9 Advanced Collections, Chapter 8 Functions, and Chapter 5 Collections.

Questions

1. We have a small module written using the Python C API which iterates through the processes running on Windows. There is also a Linux version, but that one is pure Python.

The advantage of this module is that it returns the parent process identifier (PPID) of each process, whereas conventional tools, like `tasklist`, do not.

The module is called `GetProcs`, and the interfaces required for this exercise are:

`GetFirstProc` Start iteration of running processes.

`GetNextProc` Next iteration of running processes.

Both functions take no parameters and return a tuple containing three items: (*integer process id*, *integer parent process id*, *string executable name*). `False` is returned at the end of the iteration.

Write a Python program which has a generator function called `iGetProcs`. It should yield the same tuple as returned by the `GetProcs` interfaces. Make sure that the first process is not omitted! You will need to write test code for `iGetProcs`, we suggest using a `for` loop.

2. Using `iGetProcs` written in the previous exercise, create a dictionary where the keys are the process ids and the values are each a list containing the parent process id and executable name.

Hints:

The list comprehension is simpler than the example shown in the course material - no `if` statement is required.

Python 2.7.1 and Python 3 support dictionary comprehensions, but we will assume you don't have these for this exercise.

Solutions

1. Here is our solution for iGetProcs:

```
import GetProcs

#####

def iGetProcs():

    Retn = GetProcs.GetFirstProc()
    yield Retn

    while Retn:
        Retn = GetProcs.GetNextProc()
        if Retn:
            yield Retn

#####

for proc in iGetProcs():
    print(proc)
```

2. The dictionary build is fairly straightforward:

```
pids = dict()

for value in iGetProcs():
    pids[value[0]] = value[1:3]

print(pids)
```

For your interest, a dictionary comprehension in 2.7.1 or Python 3 would be:

```
pids = {pid:value for pid,*value in iGetProcs()}
```