**- System Applications**

**Built-in Modules in Python**

A large number of pre-defined functions are also available as a part of libraries bundled with Python distributions. These functions are defined in modules. A module is a file containing definitions of functions, classes, variables, constants or any other Python objects. Contents of this file can be made available to any other program. Each built-in module contains resources for certain system-specific functionalities such as OS management, disk IO, etc. The standard library also contains many Python scripts (with the .py extension) containing useful utilities.





- The sys module

<https://www.tutorialsteacher.com/python/sys-module>

<https://pythonprogramming.net/sys-module-python-3/>

<https://www.youtube.com/watch?v=iClG6Ua5qm4>

- The os module

<https://pythonprogramming.net/python-3-os-module/>

<https://www.tutorialsteacher.com/python/os-module>

- The platform module

The platform module in Python is used to access the underlying platform's data, such as, hardware, operating system, and interpreter version information. The platform module includes tools to see the platform's hardware, operating system, and interpreter version information where the program is running.





- The subprocess module

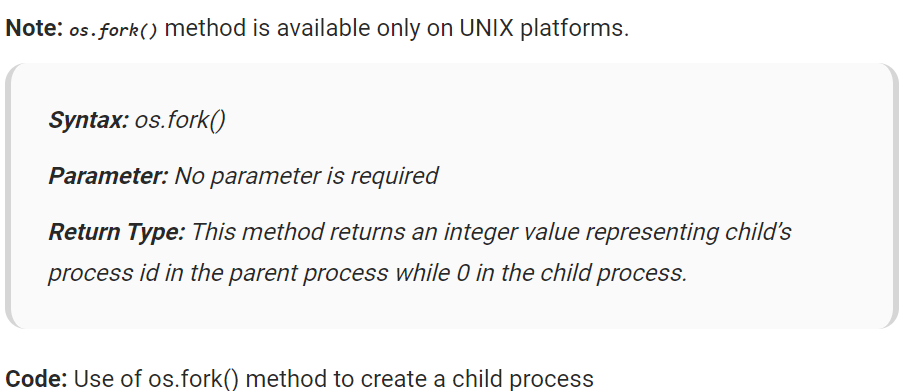
(OS can do, but using subprocess is more safe for security reason)

<https://www.youtube.com/watch?v=5XU-mAZVv4w>

<https://pythonspot.com/python-subprocess/>

- Forking and piping

os.fork(): <https://www.geeksforgeeks.org/python-os-fork-method/>



os.pipe(): <https://www.tutorialspoint.com/python/os_pipe.htm>

<https://www.youtube.com/watch?v=uHH7nHkgZ4w>

- The socket module

<https://www.geeksforgeeks.org/socket-programming-python/>

|  |
| --- |
| # An example script to connect to Google using socket # programming in Python # import socket # for socket # import sys # # try: # s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) # print("Socket successfully created") # except socket.error as err: # print("socket creation failed with error %s" % (err)) # # # default port for socket # port = 80 # # try: # host\_ip = socket.gethostbyname('www.google.com') # except socket.gaierror: # # # this means could not resolve the host # print # "there was an error resolving the host" # sys.exit() # # # connecting to the server # s.connect((host\_ip, port)) # # print("the socket has successfully connected to google \on port == %s" % (host\_ip)) |