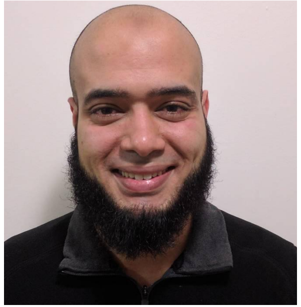
Python Programming

os and sys modules

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

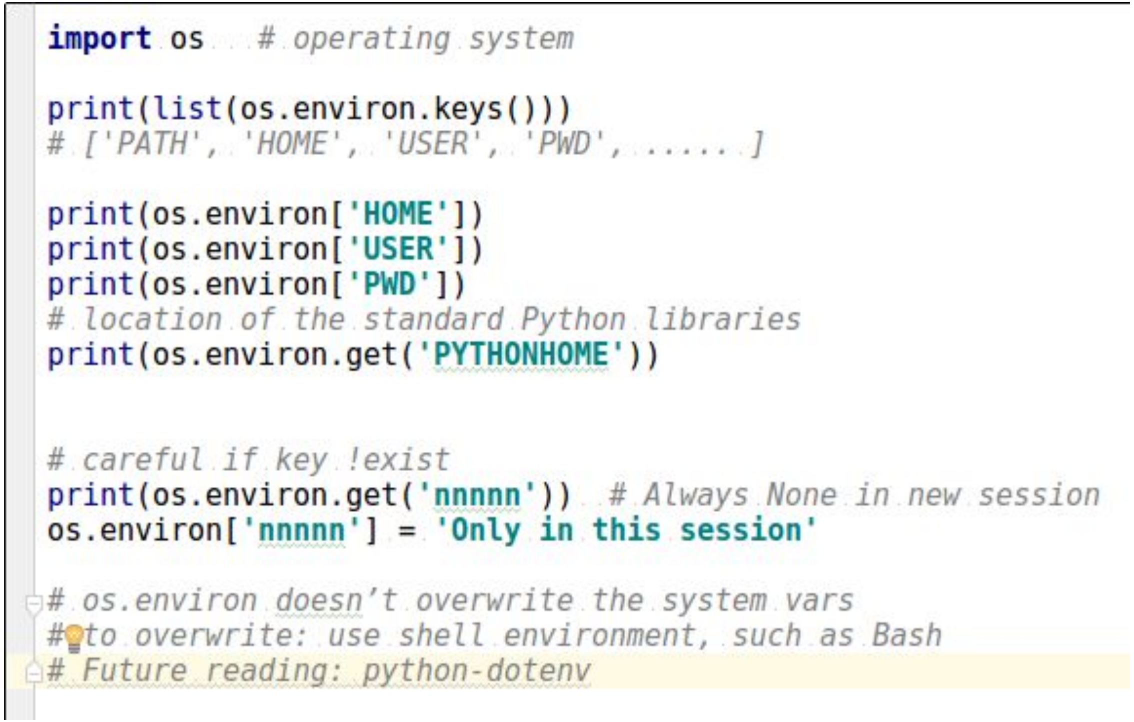
Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Environment Variables

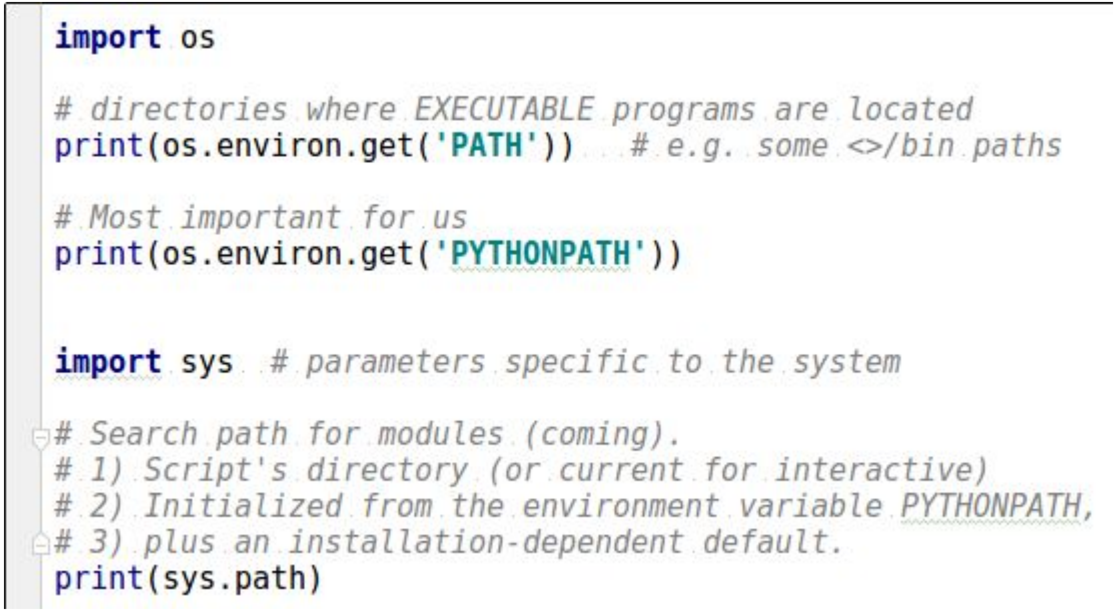
- Environment variable: (name ⇒ value) that a process may access to get/set some info (e.g. for configuration)
    - Some popular ones in linux: HOME, USER, PWD (for current working directory)
        - E.g. echo $PWD ⇒ /home/moustafa/workspaces/
- PYTHONPATH: is an Environment variable, its value are **list of directories**
    - Soon
        - It is used to **ADD PATHS** for the **user-defined** modules [primary reason]
        - Its directories are added to **sys.path** directory list
- Through **os module**, we can access environment variables from python
- Practical usages: Future reading

os module



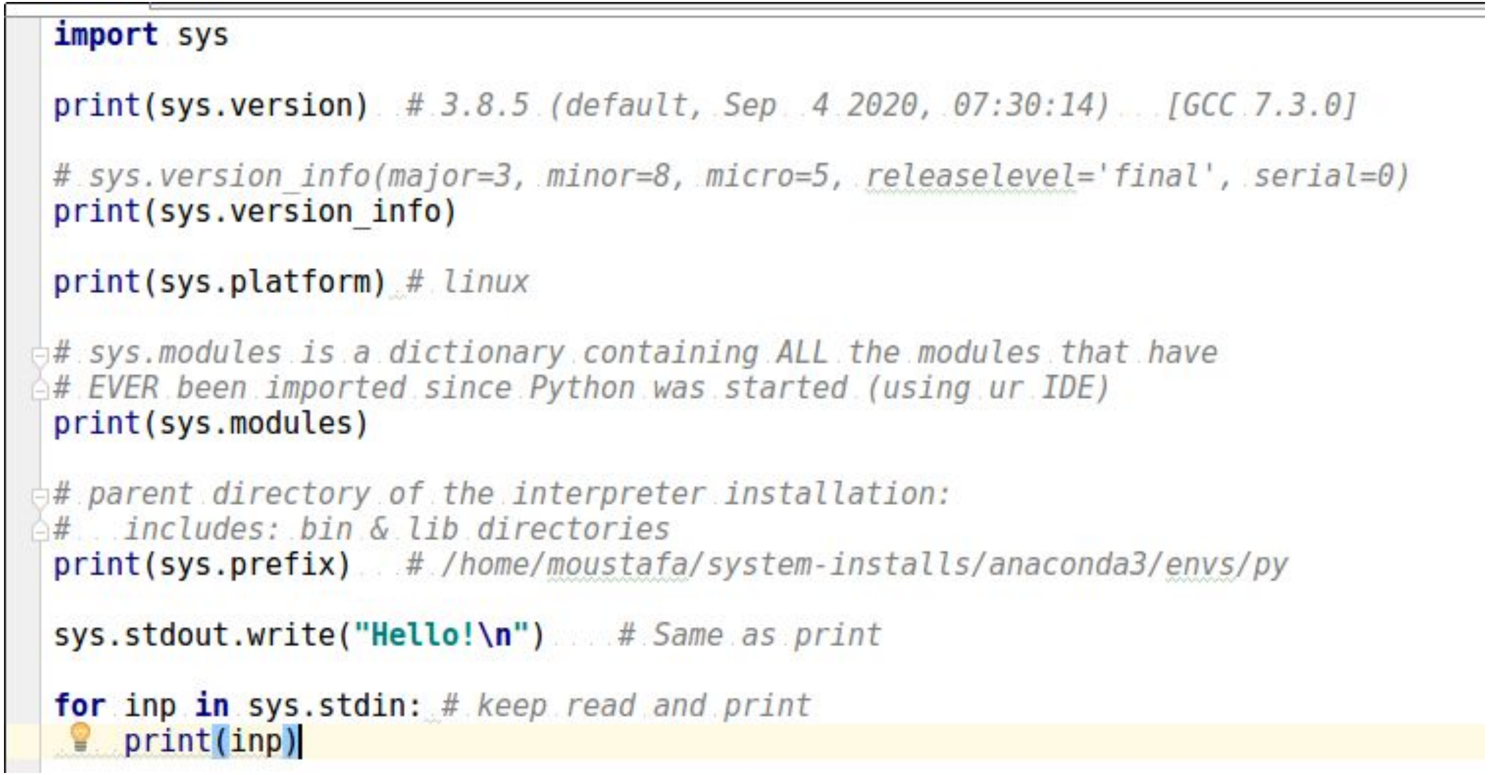
```python
import os   # operating system

print(list(os.environ.keys()))
# ['PATH', 'HOME', 'USER', 'PWD', ......]

print(os.environ['HOME'])
print(os.environ['USER'])
print(os.environ['PWD'])
# location of the standard Python libraries
print(os.environ.get('PYTHONHOME'))


# careful if key !exist
print(os.environ.get('nnnnn'))  # Always None in new session
os.environ['nnnnn'] = 'Only in this session'

# os.environ doesn’t overwrite the system vars
# to overwrite: use shell environment, such as Bash
# Future reading: python-dotenv
```

os Directories!



```
import os

# directories where EXECUTABLE programs are located
print(os.environ.get('PATH'))   # e.g. some <>/bin paths

# Most important for us
print(os.environ.get('PYTHONPATH'))


import sys  # parameters specific to the system

# Search path for modules (coming).
# 1) Script's directory (or current for interactive)
# 2) Initialized from the environment variable PYTHONPATH,
# 3) plus an installation-dependent default.
print(sys.path)
```

sys module



```
import sys

print(sys.version)  # 3.8.5 (default, Sep  4 2020, 07:30:14)   [GCC 7.3.0]

# sys.version_info(major=3, minor=8, micro=5, releaselevel='final', serial=0)
print(sys.version_info)

print(sys.platform) # linux

# sys.modules is a dictionary containing ALL the modules that have
# EVER been imported since Python was started (using ur IDE)
print(sys.modules)

# parent directory of the interpreter installation:
#   includes: bin & lib directories
print(sys.prefix)   # /home/moustafa/system-installs/anaconda3/envs/py

sys.stdout.write("Hello!\n")    # Same as print

for inp in sys.stdin:  # keep read and print
    print(inp)
```

Changing PYTHONPATH from OS level

- Consider the following in the *future*
- We can change pythonpath from OS itself
    - It could be changed per **session or permanently**
    - It is common question, google it if facing issues
- For Ubuntu/Linux
    - **printenv** command: print all the environment variables
    - **echo** $PYTHONPATH ⇒ print current value
    - **export** PYTHONPATH=$PYTHONPATH:/home/moustafa/misc   [change in session]
- We may also want to change permanently: check out for different OSes
    - Windows - Mac/Linux - More ways

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."