



# PYTHON



## \$12: Một số ứng dụng



# os module

- <https://docs.python.org/3/library/os.html>
- `os.system("command")`  
VD: `os.system("dir")`
- `os.getcwd()`  
Lấy thư mục hiện hành
- `os.chdir(path)`: Thay đổi thư mục hiện hành
- `os.mkdir("newdir")`
- `os.rmdir('dirname')`: Xóa thư mục
- `os.rename(current_file_name, new_file_name)`

# os module (tt)

- `os.listdir()` : List Directories of CWD
- `os.listdir("c:\python39")`: List Directories
- `os.chmod('file1.txt', 0o444)`
- `os.getlogin()`: user đang login
- `print(os.environ)`: danh sách biến environment
  
- `os.path.isdir(path)`
- `os.path.isfile(path)`
- `dir = os.path.join(os.environ['PUBLIC'], 'root', 'api', 'file1.txt')`
- `dirname, basename = os.path.split(dir)`

# **Execute Commands (Linux/Windows)**



# Subprocess management

- Subprocess management

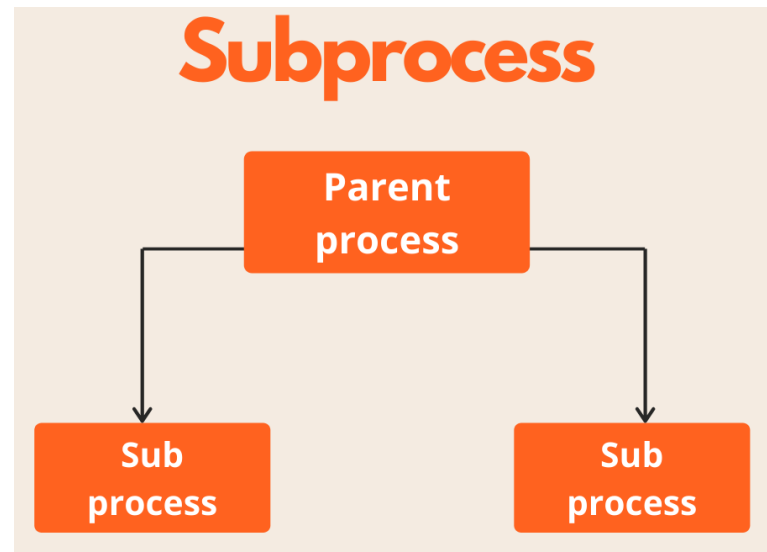
<https://docs.python.org/3/library/subprocess.html>

- Ex:

```
import subprocess
```

```
subprocess.call("command1")
```

```
subprocess.call(["command1", "arg1", "arg2"])
```



# Logger

- `import logging`
- `logging = logging.getLogger()`
- Logging levels: `DEBUG`, `INFO`, `WARNING`, `ERROR`, `CRITICAL`
- `logging.debug('Debug')`
- `logging.info('Info')`
- `logging.warning('Warning')`
- `logging.error('Error')`
- `logging.critical('Critical')`

# Config logger

- `logging.basicConfig(level=logging.DEBUG)`
- `logging.basicConfig(filename='app.log', filemode='w', format='%(name)s - %(levelname)s - %(message)s')`
- `logging.basicConfig(filename='applog_{:%Y-%m-%d}.log'.format(datetime.now()), filemode='a', format='%(asctime)s - %(name)s - %(levelname)s : %(message)s')`

<https://realpython.com/python-logging/>

# Message Format

<code>%(name)s</code>	Name of the logger
<code>%(levelname)s</code>	Numeric logging level
<code>%(levelname)s</code>	Name of logging level
<code>%(pathname)s</code>	Path of source file
<code>%(filename)s</code>	Filename of source file
<code>%(module)s</code>	Module name
<code>%(lineno)d</code>	Line number
<code>%(created)f</code>	Time when logging call executed
<code>%(asctime)s</code>	ASCII-formated date/time
<code>%(thread)d</code>	Thread-ID
<code>%(threadName)s</code>	Thread name
<code>%(process)d</code>	Process ID
<code>%(message)s</code>	Logged message



# Send Mail

- <https://docs.python.org/3/library/smtplib.html>
- `import smtplib`
- `smtplib.SMTP_SSL(host="", port=0, local_hostname=None, keyfile=None, certfile=None, [timeout, ]context=None, source_address=None)`
  - `server = smtplib.SMTP_SSL('smtp.gmail.com', 465)`
- `server.ehlo()`
- `server.login(username, password)`
- `server.sendmail(sent_from, to, email_text)`
- `server.close()`

# Send Mail

- Thực hành:
  - [pynhatnghe@gmail.com](mailto:pynhatnghe@gmail.com)
  - Bật Less Secure apps để gửi mail qua SMTP:  
<https://myaccount.google.com/lesssecureapps>
  - Thực hành gửi mail từ 01 danh sách cho trước. Nội dung chuẩn bị sẵn, load từ file.

# Crontab

- <https://pypi.org/project/python-crontab/>
- Cron là một quá trình hệ thống được sử dụng để thực hiện các background tasks trên cơ sở thường xuyên. Cron yêu cầu một tệp có tên là crontab chứa danh sách các tác vụ được thực thi tại một thời điểm cụ thể. Tất cả các job này được thực hiện trong background tại thời điểm quy định
- Python hỗ trợ ☺ - LINUX only

# PyQRCode

- pip install PyQRCode
- <https://pythonhosted.org/PyQRCode>

```
import pyqrcode
```

```
# String which represent the QR code
```

```
s = "https://www.youtube.com/c/HIENLTHChannel"
```

```
# Generate QR code
```

```
qr = pyqrcode.create(s)
```

```
# Create and save the png file naming
```

```
qr.svg("myyoutube.svg", scale=8)
```

```
qr.png("myyoutube.png", scale=8)
```

