

Obsługa procesów

Tworzenie procesów

Tworzenie procesów:

- to uruchamianie z poziomu programu Python'a innych programów, np. systemowych,
- moduł 'subprocess' – pozwala na uruchamianie procesu, obsługę strumieni input/output/error i przechwytywania kodu wyjścia.

Tworzenie procesów – str arg.

Przykład 1a, argument typu 'str':

```
import subprocess

out = subprocess.run("echo Hello World!", shell=True)

print('---')
print(out)
print('---')
print("return code: %s" %(out.returncode))
print("stdout: %s" %(out.stdout))
print("stderr: %s" %(out.stderr))
```

#output:

Hello World!

CompletedProcess(args='echo Hello World!', returncode=0)

return code: 0

stdout: None

stderr: None

Tworzenie procesów – list arg.

Przykład 1b, argument typu lista:

```
import subprocess

out = subprocess.run(["echo", "Hello World!"], shell = True)

print('---')
print(out)
print('---')
print("return code: %s" %(out.returncode))
print("stdout: %s" %(out.stdout))
print("stderr: %s" %(out.stderr))
```

#output:

Hello World!

CompletedProcess(args=['echo', 'Hello World!'], returncode=0)

return code: 0

stdout: None

stderr: None

Tworzenie procesów - stdout

Przykład 1c, przekierowanie wyjścia do zmiennej stdout:

```
import subprocess

out = subprocess.run(["echo", "Hello World!"], stdout=subprocess.PIPE, shell=True)
print('---')
print(out)
print('---')
print("return code: %s" %(out.returncode))
print("stdout: %s" %(out.stdout))
print("stderr: %s" %(out.stderr))
```

#output:

```
CompletedProcess(args=['echo', 'Hello World!'], returncode=0, stdout=b'"Hello World!"\r\n')
```

return code: 0

stdout: b'"Hello World!"\r\n'

stderr: None

Tworzenie procesów - stderr

Przykład 1d, próba uruchomienia błędnego polecenia:

```
import subprocess

out = subprocess.run(["ech", "Hello World!"], stdout=subprocess.PIPE, shell=True)
print('---')
print(out)
print('---')
print("return code: %s" %(out.returncode))
print("stdout: %s" %(out.stdout))
print("stderr: %s" %(out.stderr))
```

#output:

Nazwa 'ech' nie jest rozpoznawana jako polecenie wewn@trzne lub zewn@trzne, program wykonywalny lub plik wsadowy.

```
CompletedProcess(args=['ech', 'Hello World!'], returncode=1, stdout=b'')
```

```
return code: 1
stdout: b''
stderr: None
```

Tworzenie procesów - stderr

Przykład 1e, przekierowanie 'standard error' do zmiennej stderr:

```
import subprocess

out = subprocess.run(["ech", "Hello World!"], stdout=subprocess.PIPE,
stderr=subprocess.PIPE, shell=True)
print('---')
print(out)
print('---')
print("return code: %s" %(out.returncode))
print("stdout: %s" %(out.stdout))
print("stderr: %s" %(out.stderr))
```

#output:

CompletedProcess(args=['ech', 'Hello World!'], returncode=1, stdout=b'', stderr=b'"Nazwa \'ech\' nie jest rozpoznawana jako polecenie wewn\x9a9trzne lub zewn\x9a9trzne,\r\nprogram wykonywalny lub plik wsadowy.\r\n"')

return code: 1

stdout: b''

stderr: b'"Nazwa \'ech\' nie jest rozpoznawana jako polecenie wewn\x9a9trzne lub zewn\x9a9trzne,\r\nprogram wykonywalny lub plik wsadowy.\r\n"'

Tworzenie procesów – except.

Przykład 2a, uruchamianie błędnego polecenia z generowaniem wyjątku:

```
import subprocess

out = subprocess.run(["ech", "Hello World!"], stdout=subprocess.PIPE,
stderr=subprocess.PIPE, check=True, shell=True)
print('---')
print(out)
print('---')
print("return code: %s" %(out.returncode))
print("stdout: %s" %(out.stdout))
print("stderr: %s" %(out.stderr))
```

#output:

Traceback (most recent call last):

File "D:\mwisniew_notatki\python\zajecia\przyklady\7_proc\p1g.py", line 5, in <module>

out = subprocess.run(["ech", "Hello World!"], stdout=subprocess.PIPE, stderr=subprocess.PIPE, check=True, shell=True)

File "C:\Users\mwisniew\AppData\Local\Programs\Python\Python36\lib\subprocess.py", line 418, in run

output=stdout, stderr=stderr)

subprocess.CalledProcessError: Command '['ech', 'Hello World!']' returned non-zero exit status 1.

Tworzenie procesów – except.

Przykład 2b, uruchamianie błędnego polecenia i przechwycenie wyjątku:

```
import subprocess

out=""

try:
    out = subprocess.run(["ech", "Hello World!"], check=True, shell=True,
        stdout=subprocess.DEVNULL, stderr=subprocess.DEVNULL)
except (subprocess.CalledProcessError) as ex:
    print(ex)
```

#output:
Command '['ech', 'Hello World!']' returned non-zero exit status 1.

Zadanie 1

Napisz skrypt, który wykonuje polecenie linii komend systemu tj.: "dir [katalog]" i przechwytyje wyjście do odpowiedniej zmiennej. Skrypt powinien także obsługiwać wyjątki związane z wykonaniem tego polecenia (np. podana błędna nazwa katalogu).

Zadanie 2

Korzystając z modułu subprocess i komendy shella ,mkdir' napisz skrypt, który tworzy na dysku strukturę katalogów podaną w formie (podkreślenie oznacza poziom podkatalogu):

```
dirStr="""
```

```
K1
```

```
_K2
```

```
_K3
```

```
__K4
```

```
K5
```

```
_K6
```

```
"""
```

Zadanie 3

Korzystając z modułu „subprocess” napisz program, który:

- pobierze listę plików – skryptów python’a z katalogu ./scripts (podkatalog katalogu programu),
- następnie wygeneruje dwie listy:
 - skrypty poprawnie wykonane,
 - skrypty generujące błędy.