Wyklad7

December 11, 2019

1 Moduł sys

[1]:
```
import sys
```

[ ]:
```
help(sys)
```

[3]:
```
sys.path #atrybut znany z ostatnich zajęć
```

[3]:
```
['/usr/lib/python35.zip',
 '/usr/lib/python3.5',
 '/usr/lib/python3.5/plat-x86_64-linux-gnu',
 '/usr/lib/python3.5/lib-dynload',
 '',
 '/home/artur/.local/lib/python3.5/site-packages',
 '/usr/local/lib/python3.5/dist-packages',
 '/usr/lib/python3/dist-packages',
 '/home/artur/.local/lib/python3.5/site-packages/IPython/extensions',
 '/home/artur/.ipython']
```

[2]:
```
print(sys.version)
```

```
3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0]
```

[3]:
```
sys.platform
```

[3]:
```
'linux'
```

[4]:
```
sys.prefix
```

[4]:
```
'/home/artur/anaconda3'
```

[5]:
```
sys.executable
```

[5]:
```
'/home/artur/anaconda3/bin/python'
```

[6]:
```
sys.maxsize
```

[6]:
```
9223372036854775807
```

1.0.1 Atrybuty wejścia/wyjścia

[9]:
```
print("Hello")
```

```
Hello
```

[10]:
```
sys.stdout.write("Hello")
```

```
Hello
```

[11]:
```
print(input())
```

```
5
5
```

[ ]:
```
print(sys.stdin.readlines()) #patrz terminal
```

[12]:
```
plik0 = open('/home/artur/Dokumenty/Praca/python/Wyklad2019/Moduly/stdin_out.
↪py')
print(plik0.read())
plik0.close()
```

```
import sys

#plik_error = open("errors.txt","w")
#sys.stderr = plik_error

while True:
    try:
        number = input("Podaj liczbe: \n")
    except EOFError: #np. Ctrl+d
        print("Koniec")
        break
    else:
        number = int(number)
        if number == 0:
            print("Nie ma dzielenia przez 0", file=sys.stderr)
        else:
            print("Liczba odwrotna do {:d} to  {:f}".format(number, 1.0/number))


#plik_error.close()
```

Wykonujemy: python stdin_out.py oraz python stdin_out.py < numery.txt >output.txt

1.0.2 Argumenty wiersza polecenia - sys.argv

[13]:
```
plik1 = open('/home/artur/Dokumenty/Praca/python/Wyklad2019/Moduly/iloczyn.py')
print(plik1.read())
plik1.close()
```

```
"""Liczy iloczyn podanych argumentów."""

import sys

if len(sys.argv) < 2:  # sys.argv[0] = iloczyn.py
    print("Musisz podać co najmniej jedną liczbę .")
else:
    wyrazenie = "*".join(sys.argv[1:])
    print(wyrazenie, "=", eval(wyrazenie))
```

[ ]:
```
Wykonujemy: python3 iloczyn.py 1 2 3 4 5
```

[ ]:
```
help(eval)
```

[14]:
```
eval("2+2")
```

[14]: 4

[15]:
```
def pole_powierzchni(r):
    return 3.14*r**2

def obwod(r):
    return 2*3.14*r
```

[16]:
```
property = input("Wybierz funkcje: ")
r=float(input("Podaj promien: "))
print("{}={:f}".format(property,eval(property+"(r)")))
```

```
Wybierz funkcje: obwod
Podaj promien: 5
obwod=31.400000
```

[17]:
```
plik2 = open('/home/artur/Dokumenty/Praca/python/Wyklad2019/Moduly/modul4.py')
print(plik2.read())
plik2.close()
```

```
"""Nasz czwarty moduł. """

print('Jestem:', __name__)

def minmax(test, *args):
```

```
        res = args[0]
        for arg in args[1:]:
                if test(arg, res):
                        res = arg
        return res

def lessthan(x, y): return x < y
def grtrthan(x, y): return x > y

if __name__ == '__main__':
        print("wynik=",minmax(lessthan, 4, 2, 1, 5, 6, 3))
        print("wynik=",minmax(grtrthan, 4, 2, 1, 5, 6, 3))
```

[18]:
```
plik3 = open('/home/artur/Dokumenty/Praca/python/Wyklad2019/Moduly/modul4sys.
 ↪py')
print(plik3.read())
plik3.close()
```

```
"""Nasz czwarty moduł. """

#print('Jestem:', __name__)

def minmax(test, *args):
        res = args[0]
        for arg in args[1:]:
                if test(arg, res):
                        res = arg
        return res

def lessthan(x, y): return x < y
def grtrthan(x, y): return x > y

if __name__ == '__main__':
        import sys

        if len(sys.argv) < 2:
                print("Modyl {} wymaga podania listy
argumentow.".format(sys.argv[0]))
        else:
                print("wynik=",minmax(lessthan, *sys.argv[1:]))
                print("wynik=",minmax(grtrthan, *sys.argv[1:]))
```

[ ]:
```
Wykonujemy: python3 modul4sys.py 4, 2, 1, 5, 6, 3
```

2 Moduł os

[1]:
```
import os
```

[ ]:
```
help(os)
```

[2]:
```
os.name
```

[2]:
```
'posix'
```

[3]:
```
current_path=os.getcwd()
print("Katalog roboczy:",current_path)
```

```
Katalog roboczy: /home/artur/Dokumenty/Praca/python/Wyklad2019
```

[22]:
```
new_path="/home/artur/Dokumenty/Praca/python/Wyklad2019/Moduly"
```

[23]:
```
os.chdir(new_path)
print("Nowy katalog:",os.getcwd())
os.chdir(current_path)
```

```
Nowy katalog: /home/artur/Dokumenty/Praca/python/Wyklad2019/Moduly
```

[ ]:
```
print(os.listdir())
```

[25]:
```
print(os.listdir(new_path))
```

```
['modul4sys.py', 'modul1c.py', 'numery.txt', 'output.txt', 'modul1b.py',
'__pycache__', 'modul1.py', 'modul3.py', 'stdin_out.py', 'iloczyn.py',
'modul2b.py', 'modul2c.py', 'modul4.py', 'modul2.py']
```

2.0.1 Tworzenie i usuwanie katalogow

[26]:
```
"Nowy_katalog" in os.listdir()
```

[26]:
```
False
```

[27]:
```
os.mkdir("Nowy_katalog")
"Nowy_katalog" in os.listdir()
```

[27]:
```
True
```

[28]:
```
os.rmdir("Nowy_katalog")
"Nowy_katalog" in os.listdir()
```

[28]:
```
False
```

[29]:
```
os.mkdir("Nowy_katalog")
file = open("Nowy_katalog/nowy_plik", 'w')
file.close()
print(os.listdir("Nowy_katalog"))
```

```
['nowy_plik']
```

[30]:
```
os.rmdir("Nowy_katalog")
```

```
␣
↪---------------------------------------------------------------------------

OSError                                   Traceback (most recent call last)

<ipython-input-30-21a1b281a953> in <module>
----> 1 os.rmdir("Nowy_katalog")


OSError: [Errno 39] Directory not empty: 'Nowy_katalog'
```

[31]:
```
os.remove("Nowy_katalog/nowy_plik")  # usuwa plik
os.rmdir("Nowy_katalog")           # usuwa katalog
```

[32]:
```
os.makedirs("level0/level1/level2") # utwórz "ciąg" katalogów
```

[33]:
```
os.removedirs("level0/level1/level2")  # usunie wszystko
```

[4]:
```
os.system("pwd") #działa w terminalu
```

[4]: 0

[34]:
```
os.system("geany")
```

[34]: 0

[37]:
```
os.system("mkdir Moduly/Nowy_katalog2")
```

[37]: 0

[38]:
```
os.system("rmdir Moduly/Nowy_katalog2")
```

[38]: 0

[39]:
```
path = "/my/path"  # katalog
file = "file.py"   # plik
```

```
print(os.path.join(path, file))
```

```
/my/path/file.py
```

[40]:
```
wynik=os.path.join(path, file)
print(os.path.split(wynik))
```

```
('/my/path', 'file.py')
```

[41]:
```
print(os.path.dirname(wynik))
```

```
/my/path
```

[42]:
```
print(os.path.split(wynik)[0])
```

```
/my/path
```

[43]:
```
print(os.path.basename(wynik))
```

```
file.py
```

[44]:
```
print(os.path.split(wynik)[1])
```

```
file.py
```

[45]:
```
print(os.path.splitext(wynik))
```

```
('/my/path/file', '.py')
```

2.0.2 Spacer

[ ]:
```
# os.walk "podróżuje po drzewie katalogów"
# na każdym kroku zwracając krotkę
# (obecny katalog, lista podkatalogów, lista plików)
for root, dirs, files in os.walk("/home/artur/Dokumenty/Praca/python/"):
    print(root, dirs, files, sep="\n", end="\n\n")
```

[ ]:
```
for root, dirs, files in os.walk("/home/artur/Dokumenty/Praca/python/"):
    for file in files: # pętla po plikach w danym katalogu root
        print(os.path.join(root, file))
```

2.0.3 Moduł glob

[ ]:
```
print(os.listdir())
```

[49]:
```
import glob
```

[50]:
```
print(glob.glob("*.pdf"))
```

```
['ww2.pdf', 'Feathertheme.pdf', 'Wyklad3.pdf', 'Wyklad4.pdf']
```

[51]:
```
#(coś)(cyfra od 0 do 5)
print(glob.glob("*[0-5].pdf"))
```

```
['ww2.pdf', 'Wyklad3.pdf', 'Wyklad4.pdf']
```

[52]:
```
print(glob.glob("w?2.[ptd]*"))  # js-python_[znak]02.[p lub t lub d]cokolwiek
```

```
['ww2.pdf', 'ww2.toc', 'ww2.tex', 'ww2.dvi']
```

3 Moduł time

[1]:
```
import time
```

[54]:
```
print(time.gmtime())
```

```
time.struct_time(tm_year=2019, tm_mon=11, tm_mday=22, tm_hour=7, tm_min=50,
tm_sec=5, tm_wday=4, tm_yday=326, tm_isdst=0)
```

[55]:
```
print(time.localtime())
```

```
time.struct_time(tm_year=2019, tm_mon=11, tm_mday=22, tm_hour=8, tm_min=50,
tm_sec=43, tm_wday=4, tm_yday=326, tm_isdst=0)
```

[56]:
```
print(time.localtime()[3] - time.gmtime()[3])  # różnica Polska - GMT
```

```
1
```

[57]:
```
print(time.gmtime(0))
```

```
time.struct_time(tm_year=1970, tm_mon=1, tm_mday=1, tm_hour=0, tm_min=0,
tm_sec=0, tm_wday=3, tm_yday=1, tm_isdst=0)
```

[12]:
```
x=(time.ctime()).split()
print(x)
```

```
['Fri', 'Dec', '6', '09:58:20', '2019']
```

[9]:
```
x.split()
```

[9]:
```
['Fri', 'Dec', '6', '09:50:13', '2019']
```

[58]:
```
print(time.strftime("%c", time.localtime()))  # lub print(time.asctime(time.
↪localtime()))
```

```
Fri Nov 22 08:52:08 2019
```

[59]:
```
print(time.strftime("%Y-%m-%d %H:%M", time.localtime()))
```

```
2019-11-22 08:52
```

3.0.1 Pomiar czasu

[60]:
```
def poczekalnia(n):
    """Czeka n sekund."""
    time.sleep(n)
```

[67]:
```
t0 = time.time()
poczekalnia(10)
tk = time.time()
```

[68]:
```
print("Czekałem {} sekund.".format(tk - t0))
```

```
Czekałem 10.010932683944702 sekund.
```

[65]:
```
t0 = time.clock() #czas procesora
poczekalnia(10)
tk = time.clock()
```

[64]:
```
print("Czekałem {} sekund.".format(tk - t0))
```

```
Czekałem 0.0017439999999999678 sekund.
```

[66]:
```
print("Czekałem {} sekund.".format(tk - t0))
```

```
Czekałem 0.002367000000000008 sekund.
```

4 Moduł datatime

[69]:
```
import datetime
```

[70]:
```
t0 = datetime.datetime.now()
t1 = datetime.datetime(2020, 2, 4)  # początek sesji
```

[71]:
```
print(t1 - t0)
```

```
73 days, 15:04:09.761373
```

[72]:
```
print("Dzisiejsza data =", t0)
print("rok =", t0.year)
print("miesiąc =", t0.month)
```

```
print("dzień =", t0.day)
print("Do sesji zostało", t1 - t0)
```

```
Dzisiejsza data = 2019-11-22 08:55:50.238627
rok = 2019
miesiąc = 11
dzień = 22
Do sesji zostało 73 days, 15:04:09.761373
```

[73]:
```
t0.strftime('%Y_%m_%d_%H:%M:%S')
```

[73]:
```
'2019_11_22_08:55:50'
```

[ ]:
```

```