Ćwiczenia 9

Moduły i pakiety

1. Stwórz nowy projekt (poza notatnikiem) i dodaj plik `mod.py` o zawartości:

```
s = "Data science."
a = [100, 200, 300]


def foo(arg):
    print(f'arg = {arg}')


class Foo:
    pass
```

2. Uruchom na konsoli podane instrukcje:

```
import mod

print(mod.s)
print(mod.a)
mod.foo(['abc', 'XYZ', '123'])
x = mod.Foo()
print(x)
```

3. Sprawdź, skąd instalowane są moduły uruchamiając komendę:

```
print(sys.path)
```

i sprawdźmy gdzie znajduje się nasz moduł:

```
print(mod.__file__)
```

4. Uruchom komendy:

```
print(mod)
print(s)
print(foo('abc'))
```

5. Inny sposób import modułu:

```
from mod import s, foo
print(s)
print(foo('abc'))

from mod import Foo
x = Foo()
print(x)
```

lub

```
from mod import *


print(s)
print(a)
print(foo)
print(Foo)
```

6. Sprawdź listę zdefiniowanych przestrzeni nazw.

```
print(dir())
```

7. Sprawdź uruchamiają różne konsole jak ma się wynik `dir()` do załadowanych modułów na różne sposoby.
8. Uruchom terminal i wpisz komendę:

```
python mod.py
```

9. Zmodyfikuj kod w pliku `mod.py` i jeszcze raz uruchom skrypt z poziomu terminala:

```
s = "Data science."
a = [100, 200, 300]


def foo(arg):
    print(f'arg = {arg}')


class Foo:
    pass


print(s)
print(a)
foo('XYZ')
x = Foo()
print(x)
```

10. Zmodyfikuj ponownie plik `mod.py` i uruchom skrypt z poziomu terminala:

```
s = "Data science."
a = [100, 200, 300]


def foo(arg):
    print(f'arg = {arg}')


class Foo:
    pass


if __name__ == '__main__':
    print(s)
    print(a)
    foo('XYZ')
```

```
    x = Foo()
    print(x)
```

11. W tym samym projekcie stwórz plik `fact.py` i następnie go uruchom z terminala.

```
def fact(n):
    return 1 if n == 1 else n * fact(n - 1)


if __name__ == '__main__':
    import sys

    if len(sys.argv) > 1:
        print(fact(int(sys.argv[1])))
```

komenda

```
python fact.py 4
```

12. Stwórz nowy projekt, stwórz w projekcie folder `pkg` a w nim 4 pliki i poniższej zawartości:

Plik `mod1.py`:

```
def foo():
    print('[mod1] foo()')


class Foo:
    pass
```

Plik `mod2.py`:

```
def bar():
    print('[mod2] bar()')


class Bar:
    pass
```

Plik `mod3.py`:

```
def baz():
    print('[mod3] baz()')


class Baz:
    pass
```

Plik `mod4.py`:

```
def qux():
    print('[mod4] qux()')


class Qux:
    pass
```

13. Poćwicz różne sposoby importu i sprawdź różnice:

```
from pkg.mod3 import *
from pkg import *
```

14. Dodaj w folderze `pkg` plik `__init__.py` o treści:

```
__all__ = ['mod1', 'mod2', 'mod3', 'mod4']
```

Sprawdź teraz różne sposoby importu.

15. Stwórz moduł symulujący pracę kalkulatora.
16. Stwórz moduł lub pakiet związany z obsługą czasu nie korzystając z gotowych zawartości.