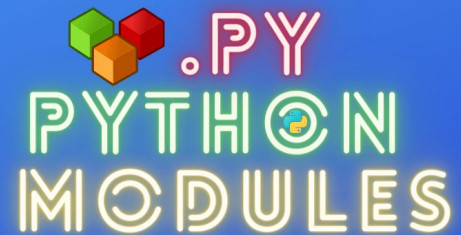


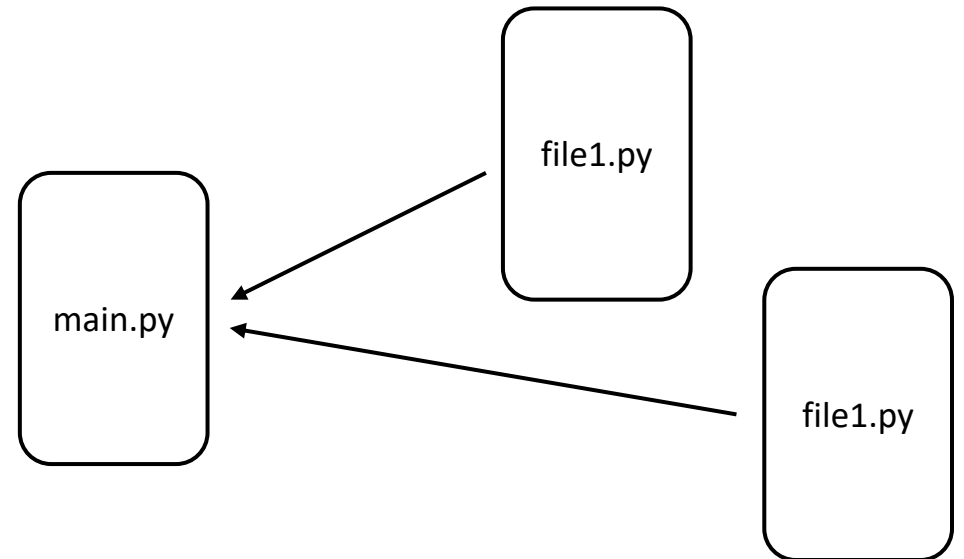
Moduły

Podstawy programowania w języku Python



Organizacja kodu

- w przypadku dłuższych programów, warto podzielić go na więcej mniejszych plików
- pomimo podziału nadal będzie można korzystać z kodu zapisanego w innych plikach
- wydzielone pliki nazywamy modułami



Moduły w Pythonie

- w Pythonie **moduły** są po prostu plikami z rozszerzeniem **py**, w których zawarto pewien zestaw funkcji
- **moduły** importujemy do swojego programu za pomocą komendy **import**
- standardowa biblioteka Pythona dostarcza sporą kolekcję **modułów**

Importowanie całego modułu

- do elementów modułu (funkcji, zmiennych) musimy odwoływać się z wykorzystaniem nazwy modułu

```
import nazwa_modułu
```

```
1 import math
2
3 print(math.pi)
4 print(math.sqrt(25))
```

Importowanie wybranych elementów modułu

- do zaimportowanych elementów modułu (funkcji, zmiennych) odnosimy się po ich nazwie
- odwoływanie do pozostałych elementów za pomocą nazwy modułu nie będzie dostępne

`from nazwa_modułu import nazwa_funkcji`

```
1 from math import pi, sqrt
2
3 print(pi)
4 print(sqrt(25))
5 print(math.pi) #wyjątek NameError
```

Importowanie wszystkich elementów modułu

- użycie gwiazdki spowoduje zaimportowanie prawie wszystkich nazw
- niezalecany sposób importu ze względu na utratę czytelności (nie jest jasne, które nazwy są obecne w przestrzeni nazw)

```
from nazwa_modułu import *
```

```
1 from math import *  
2  
3 print(pi)  
4 print(sqrt(25))  
5 print(factorial(3))
```

Importowanie pod nową nazwą (aliasowanie)

- importowanie całego modułu pod nową nazwą

```
import nazwa_modułu as nowa_nazwa_modułu
```

```
1 import math as m
2
3 print(m.sqrt(25))
```

- importowanie wybranych elementów modułu pod nową nazwą

```
from nazwa_modułu import nazwa_funkcji as nowa_nazwa_funkcji
```

```
1 from math import sqrt as pierwiastek_kwadratowy
2
3 print(pierwiastek_kwadratowy(25))
```

Funkcje dir oraz help

- funkcja **dir()** pozwala podejrzeć listę dostępnych funkcji/obiektów w danym zaimportowanym module

`dir(nazwa_modułu)`

```
1 import math
2
3 print(dir(math))
```

- funkcja **help()** dostarcza opis wskazanego zaimportowanego modułu lub konkretnej funkcji

`help(nazwa_modułu)`

```
1 import random
2
3 help(random)
```


Wybrane funkcje modułu math

- `sin(x)` – sinus x
- `atan(x)` – arcus tangens x
- `pi` – stała o wartości stanowiącej przybliżenie π
- `radians(x)` – konwertuje stopnie na radiany
- `degrees(x)` – konwertuje radiany na stopnie
- `floor(x)` – największa liczba całkowita mniejsza od lub równa liczbie x

```
1 import math
2
3 print(math.floor(4.2))
```

Wybrane funkcje modułu random

- `random()` – zwraca liczbę rzeczywistą z zakresu `[0.0, 1.0)`
- `seed()` – pozwala ustawić ziarno generatora liczb pseudolosowych
- `randint(left, right)` – losuje liczbę całkowitą ze zbioru `[left, right]`
- `choice(seq)` – wybiera losowy element z podanej sekwencji `seq`
- `sample(population, k)` – zwraca listę `k` niepowtarzających się elementów z podanej sekwencji lub zbioru `population`

```
1 import random
2
3 print(random.sample([1, 2, 3, 4], 2))
```

Wybrane funkcje modułu platform

- `machine()` – zwraca ogólną nazwę procesora
- `system()` – zwraca ogólną nazwę systemu operacyjnego
- `version()` – zwraca wersję systemu operacyjnego
- `python_version_tuple()` – zwraca wersję Pythona

```
1 import platform
2
3 print(platform.version())
```

Zmienna `__name__`

- podczas importowania modułu jego kod źródłowy jest interpretowany i wykonywany
- specjalna zmienna `__name__` przechowuje nazwę modułu w postaci napisu
- kiedy plik jest bezpośrednio uruchamiany, jego zmienna `__name__` jest ustawiona jako `"__main__"`
- kiedy plik importowany jest jako moduł, jego zmiennej `__name__` zostaje przypisana nazwa pliku (z pominięciem rozszerzenia `.py`)
- sprawdzenie wartości zmiennej `__name__` w module pozwala uniknąć wykonywania kodu modułu podczas jego importowania

Pytanie

Jakie rozszerzenie pliku posiadają moduły w Pythonie?

- a) .pyc – czyli specjalne cachowane pliki Pythona
- b) .py – bo to zwykłe skrypty Pythona
- c) .mpy – jak moduł Pythona
- d) moduły to foldery z plikami, więc nie posiadają rozszerzeń

Odpowiedź: b)

W Pythonie moduły są po prostu plikami z rozszerzeniem **.py**, w których zawarto pewien zestaw funkcji.

Pytanie

Które polecenie **importu** musimy zastosować, aby móc odwoływać się do funkcji **radians()** z modułu **math** jako **rad()**?

- a) `import math`
- b) `import radians as rad`
- c) `from math import *`
- d) `import math as rad`
- e) `from math import radians as rad`

Odpowiedź: e)

Z modułu `math` (`from math`) dokonujemy importu funkcji `radians` (`import radians`) pod nazwą `rad` (`as rad`).

Pytanie

Co wyświetli się na ekranie po wykonaniu poniższego skryptu?

```
1 from random import sample as s
2
3 print(s([1, 2, 3], 3))
```

- a) wystąpi wyjątek NameError
- b) za każdym razem wyświetli się dokładnie lista [1, 2, 3]
- c) lista z elementami [1, 2, 3] w losowej kolejności
- d) lista jako dowolna kombinacja elementów 1, 2, 3 np. [1, 1, 3]

Odpowiedź: c)

Funkcja `sample()` z modułu `random` losuje z podanej sekwencji pewną liczbę niepowtarzających się elementów. Dlatego za każdym razem dostaniemy listę z elementami [1, 2, 3] ułożonymi w dowolnej kolejności.