

# Moduły

- Modułem jest każdy plik zawierający kod napisany w języku Python.
- Nazwę modułu stanowi nazwa pliku z pominięciem rozszerzenia.
- Wewnątrz modułu jego nazwa dostępna jest poprzez zmienną `__name__`.
- Dzięki zmiennej `__name__` możliwe jest zdefiniowanie bloku kodu, który ma być wykonany tylko w przypadku, gdy plik używany jest jako program główny, a nie jako moduł (importowany w innym pliku).

# main

```
if __name__ == '__main__':  
    #kod umieszczony w tym bloku wykona się tylko ,  
    #jeśli użyjemy pliku , w którym się on znajduje  
    #jako programu głównego  
    pass
```

W ramach konwencji, pozostawia się powyższy warunek - z pustą instrukcją wewnątrz  
- w udostępnianych modułach, jako ślad po jego testowaniu.

- Każdy moduł określa własną przestrzeń nazw.
- Oprócz definicji funkcji czy klas może on zawierać także instrukcje wykonywalne, które zostaną wykonane raz w momencie pierwszego importu modułu.
- Przy pierwszym imporcie modułu generowany jest katalog `__pycache__`, w którym przechowywana jest skompilowana wersja modułu.
- Plik modułu ma rozszerzenie `pyc`, do jego nazwy dodawana jest także informacja o wersji Pythona użytej do jego wygenerowania (CPython, nr wersji).
- Przy ponownym imporcie modułu porównywane są daty modyfikacji pliku źródłowego oraz skompilowanego, jeśli pierwsza z nich jest późniejsza moduł jest ponownie kompilowany, w przeciwnym przypadku wykorzystywana jest wersja skompilowana.
- Przy imporcie modułu interpreter poszukuje go w ścieżkach zawartych w liście określonej przez zmienną `sys.path` inicjalizowanej m.in. ścieżką do katalogu bieżącego.

# import

- Zalecane jest importowanie modułu jako całości, tj.:

```
import module
```

- W uzasadnionych przypadkach można importować poszczególne składowe, bądź wszystkie składowe z wyłączeniem tych, które mają nazwy specjalne (tzn. rozpoczynające się znakiem podkreślenia):

```
from module import name
```

```
from module import *
```

# Łańcuch dokumentacyjny

- Zwłaszcza w przypadku udostępnianych modułów, on sam oraz jego składowe powinny zawierać łańcuchy dokumentacyjne.
- Łańcuch dokumentacyjny stanowi nienazwany obiekt typu **str** umieszczony na samym początku pliku/składowej.
- Użycie potrójnego cudzysłowu pozwala zachować formatowanie w dokumentacji wielowierszowej.
- Łańcuch dokumentacyjny można wyświetlić używając funkcji wbudowanej `help`:

**help**( module )

**help**( module . name )

# Pakiety

- Moduły mogą być strukturyzowane za pomocą pakietów, do tworzenia których wykorzystywana jest struktura katalogów.
- Katalog mający być pakietem musi przechowywać plik o nazwie `__init__.py`, który może być pusty, może też zawierać kod inicjalizacyjny dla pakietu.
- W pliku `__init__.py` może być zdefiniowana lista o nazwie `__all__`, w której zapisywane są nazwy modułów, które mają być importowane w przypadku wpisania:

```
from pakiet import *
```

- Dostęp do składowych pakietu możliwy jest przez użycie kropki.