Модулі та пакети

№ уроку: 9 **Курс:** Python Essential

Засоби навчання: PyCharm

Огляд, ціль та призначення уроку

Після завершення уроку учні матимуть уявлення про модулі, пакети та систему імпортування модулів у Python.

Вивчивши матеріал даного заняття, учень зможе:

- Створювати свої модулі та об'єднувати їх у пакети
- Імпортувати модулі, окремі імена з модулів
- Користуватись деякими стандартними модулями

Зміст уроку

1. Що таке модулі?
2. Як імпортувати модулі в Python?
3. Оператор імпорту Python
4. Імпорт із перейменуванням
5. from ... import виразу
6. Імпортування всіх імен
7. Шлях пошуку модуля Python
8. Що таке пакет?

Резюме

Модулі - це файли, що містять звичайні вирази Python.

Файл, що містить код Python, наприклад **example.py**, називається модулем, і ім'я модуля буде example. Ми використовуємо модулі, щоб розбити великі програми на невеликі керовані та організовані файли. Крім того, модулі забезпечують можливість повторного використання коду.

Ми можемо визначити наші найчастіше використовувані функції в модулі та імпортувати його, замість того, щоб копіювати їх визначення в різні програми.

Створимо модуль. Введіть наступне та збережіть як example.py.

```
# Приклад модуля на Python example

def add(a, b):
   result = a + b
   return result
```

Тут ми визначили функцію add() всередині модуля під назвою example. Функція приймає два числа та повертає їхню суму.

Як імпортувати модулі в Python?
Ми можемо імпортувати функції\класи\змінні всередині модуля в інший модуль або в інтерактивний інтерпретатор в Python.
Для цього ми використовуємо ключове слово import. Щоб імпортувати наш раніше визначений приклад модуля, ми вводимо наступне у командному рядку Python.

```
>>> import example
```


CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

При цьому імена функцій, визначених у example, не імпортуються безпосередньо до поточного файлу. Він лише імпортує ім'я модуля example.
Використовуючи ім'я модуля, можна отримати доступ до функції за допомогою оператора «крапка» (.). Наприклад:

```
>>> example.add(4,5.5)
9.5
```

У Python є безліч стандартних модулів. Ви можете ознайомитися з повним списком стандартних модулів Python та варіантів їх використання. Ці файли знаходяться в каталозі **Lib** всередині місця, де ви встановили Python.
Стандартні модулі можна імпортувати так само, як ми імпортуємо наші користувацькі модулі.
Існують різні способи імпорту модулів.

Оператор імпорту Python
Ми можемо імпортувати модуль за допомогою оператора імпорту та отримати доступ до визначень всередині нього за допомогою оператора точки, як описано вище.

```
import math
print("The value of pi is", math.pi)

# > The value of pi is 3.141592653589793
```

Імпорт з переіменуванням
Ми можемо імпортувати модуль, переіменувавши його наступним чином:

```
import math as m
print("The value of pi is", m.pi)
```

Ми перейменували модуль math на m. У деяких випадках це може заощадити час на набір тексту.
Зверніть увагу, що ім'я math не розпізнається у нашій області. Отже, math.pi більше недійсний, а m.pi – правильний запис.

Ми можемо імпортувати певні імена з модуля, не імпортуючи модуль в цілому. Ось приклад.

```
from math import pi
print("The value of pi is", pi)
```

Тут ми імпортували тільки атрибут pi з модуля math.
У таких випадках ми не використовуємо оператора крапки. Ми також можемо імпортувати кілька атрибутів наступним чином:

```
>>> from math import pi, e
>>> pi
3.141592653589793
>>> e
2.718281828459045
```

Імпортування всіх імен
Ми можемо імпортувати всі імена (визначення) з модуля, використовуючи наступну конструкцію:

```
# import all names from the standard module math

from math import *
print("The value of pi is", pi)
```

Тут ми імпортували всі визначення з модуля math. Сюди входять усі імена у нашій області видимості, крім тих, що починаються з підкреслення (приватні визначення).

Імпорт всього, що позначено символом зірочки (*), не є гарною практикою програмування. Це може призвести до дублювання визначень ідентифікатора. Це також ускладнює читаність нашого коду.

Шлях пошуку модуля Python

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Під час імпортування модуля Python переглядає декілька місць. Інтерпретатор спершу шукає вбудований модуль. Потім (якщо вбудований модуль не знайдено), Python переглядає список каталогів, визначених у sys.path. Пошук ведеться у такому порядку.

1. Поточна директорія
2. PYTHONPATH (змінна оточення зі списком каталогів)
3. Каталог за замовчуванням, котрий залежить від установки.

```
>>> import sys
>>> sys.path
['',
'C:\\Python33\\Lib\\idlelib',
'C:\\Windows\\system32\\python33.zip',
'C:\\Python33\\DLLs',
'C:\\Python33\\lib',
'C:\\Python33',
'C:\\Python33\\lib\\site-packages']
```

Ми можемо додавати та змінювати цей список, щоб додати наш власний шлях.

Що таке пакет?

Пакети є ще більшою одиницею, ніж модуль, і являють собою набір взаємопов'язаних модулів, призначених для вирішення завдань певного класу деякої предметної області (наприклад, пакет для вирішення систем рівнянь, який може включати математичний модуль, модуль зі спеціальними типами даних тощо).

Пакети в Python - це спосіб структуризації модулів. Пакет є папкою, в якій містяться модулі та інші пакети і обов'язковий файл init.py , який відповідає за ініціалізацію пакета.

Одна з основних цілей використання як модулів, так і пакетів - реалізація моделі простору імен, що дозволяє логічно групувати й водночас ізолювати різні ідентифікатори. Наприклад, за наявності глобальної змінної author у модулі A і B не станеться конфлікту, оскільки вони перебувають у різному просторі імен: A.author і B.author відповідно.

Закріплення матеріалу

- Що таке модуль?
- У чому сенс модульного програмування?
- Що таке пакет?
- Як імпортувати модуль у Python?
- Як імпортувати певне ім'я із модуля?
- Як імпортувати модуль із пакета?
- Як створити пакет?
- У чому різниця між операторами import та from … import?

Додаткове завдання

Завдання

Створіть модуль для отримання простих чисел. Імпортуйте його з іншого модуля. Імпортуйте його окремі імена.

Самостійна діяльність учня

Завдання 1

Перепишіть домашнє завдання попереднього уроку (сервіс для скорочення посилань) таким чином, щоб у нього була основна частина, яка відповідала би за логіку роботи та надавала узагальнений інтерфейс, і модуль представлення, який відповідав би за взаємодію з користувачем. При заміні останнього на інший, який взаємодіє з користувачем в інший спосіб, програма має продовжувати коректно працювати.

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Завдання 2
Повторіть інформацію про розглянуті на уроці стандартні модулі. Ознайомтеся також із модулями calendar, heapq, bisect, array, enum.

Рекомендовані ресурси

https://docs.python.org/3/tutorial/modules.html
https://docs.python.org/3/reference/import.html
https://docs.python.org/3/library/index.html
https://docs.python.org/3/library/datetime.html
https://docs.python.org/3/library/calendar.html
https://docs.python.org/3/library/heapq.html
https://docs.python.org/3/library/bisect.html
https://docs.python.org/3/library/array.html
https://docs.python.org/3/library/copy.html
https://docs.python.org/3/library/decimal.html
https://docs.python.org/3/library/fractions.html
https://docs.python.org/3/library/random.html

Статті у Вікіпедії про ключові поняття, розглянуті на цьому уроці
https://ru.wikipedia.org/wiki/Модуль_(программирование)

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com