Модули

№ урока: 10 **Курс:** Python Essential

Средства обучения: PyCharm

Обзор, цель и назначение урока

После завершения урока обучающиеся будут иметь представление о модулях и пакетах и системе импортирования модулей в Python.

Изучив материал данного занятия, учащийся сможет:

- Создавать свои модули и объединять их в пакеты
- Импортировать модули, отдельные имена из модулей
- Пользоваться некоторыми стандартными модулями

Содержание урока

1. Что такое модули?
2. Как импортировать модули в Python?
3. Оператор импорта Python
4. Импорт с переименованием
5. from … import выражения
6. Импортирование всех имен
7. Путь поиска модуля Python
8. Что такое пакет?

Резюме

Модули - это файлы, что содержат обычные выражения Python.
Файл, содержащий код Python, например: **example.py**, называется модулем, и его имя модуля будет example.
Мы используем модули, чтобы разбить большие программы на небольшие управляемые и организованные файлы. Кроме того, модули обеспечивают возможность повторного использования кода.
Мы можем определить наши наиболее часто используемые функции в модуле и импортировать его, вместо того, чтобы копировать их определения в разные программы.
Создадим модуль. Введите следующее и сохраните как example.py.

```
# Пример модуля на Python example

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

Здесь мы определили функцию add() внутри модуля с именем example. Функция принимает два числа и возвращает их сумму.

Как импортировать модули в Python?
Мы можем импортировать функции\классы\переменные внутри модуля в другой модуль или в интерактивный интерпретатор в Python.
Для этого мы используем ключевое слово import. Чтобы импортировать наш ранее определенный пример модуля, мы вводим следующее в командной строке 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 этом имена функций, определенных в example, не импортируются непосредственно в текущий файл. Он только импортирует имя модуля example.
Используя имя модуля, мы можем получить доступ к функции с помощью точки. оператор. Например:

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

В Python есть множество стандартных модулей. Вы можете ознакомиться с полным списком стандартных модулей Python и вариантов их использования. Эти файлы находятся в каталоге **Lib** внутри места, где вы установили Python.
Стандартные модули можно импортировать так же, как мы импортируем наши пользовательские модули.
Есть разные способы импорта модулей.

Оператор импорта Python
Мы можем импортировать модуль с помощью оператора импорта и получить доступ к определениям внутри него с помощью оператора точки, как описано выше.

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

# > The value of pi is 3.141592653589793
```

Импорт с переименованием
Мы можем импортировать модуль, переименовав его следующим образом:

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

Мы переименовали модуль math в m. В некоторых случаях это может сэкономить нам время на набор текста.
Обратите внимание, что имя math не распознается в нашей области. Следовательно, math.pi больше недействителен, а m.pi - правильная запись.

Мы можем импортировать определенные имена из модуля, не импортируя модуль в целом. Вот пример.

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

Здесь мы импортировали только атрибут pi из модуля math.
В таких случаях мы не используем оператор точки. Мы также можем импортировать несколько атрибутов следующим образом:

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

Импортирование всех имен
Мы можем импортировать все имена (определения) из модуля, используя следующую конструкцию:

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

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

Здесь мы импортировали все определения из модуля math. Сюда входят все имена, в нашей области видимости, кроме тех, которые начинаются с подчеркивания (частные определения).

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

Импорт всего, что отмечено символом звездочки (*), не является хорошей практикой программирования. Это может привести к дублированию определений идентификатора. Это также затрудняет читаемость нашего кода.

Путь поиска модуля Python

При импорте модуля Python просматривает несколько мест. Интерпретатор сначала ищет встроенный модуль. Затем (если встроенный модуль не найден) Python просматривает список каталогов, определенных в sys.path. Поиск ведется в таком порядке.

1. Текущая директория
2. PYTHONPATH (переменная окружения со списком каталогов)
3. Каталог по умолчанию, зависящий от 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']
```

Мы можем добавлять и изменять этот список, чтобы добавить наш собственный путь.

Что такое пакет?

Пакеты являются еще более крупной единицей, чем модуль, и представляют собой набор взаимосвязанных модулей, предназначенных для решения задач определенного класса некоторой предметной области (например, пакет для решения систем уравнений, который может включать математический модуль, модуль со специальными типами данных и т.д.).

Пакеты в Python - это способ структуризации модулей. Пакет представляет собой папку, в которой содержатся модули и другие пакеты и обязательный файл __init.py__, отвечающий за инициализацию пакета.

Одна из основных целей использования как модулей, так и пакетов - реализация модели пространства имен, позволяющей логически группировать и в то же время изолировать различные идентификаторы. Например, при наличии глобальной переменной author в модуле A и B не произойдет конфликта, т.к. они находятся в разном пространстве имен: A.author и B.author соответственно.

Закрепление материала

- Что такое модуль?
- В чём смысл модульного программирования?
- Что такое пакет?
- Как импортировать модуль в Python?
- Как импортировать определённое имя из модуля?
- Как импортировать модуль из пакета?
- Как создать пакет?
- В чём разница между операторами import и from … import?

Дополнительное задание

Задание

Создайте модуль для получения простых чисел. Импортируйте его из другого модуля. Импортируйте отдельные его имена.

Самостоятельная деятельность учащегося

Задание 1

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

Перепишите домашнее задание предыдущего урока (сервис для сокращения ссылок) таким образом, чтобы у него была основная часть, которая отвечала бы за логику работы и предоставляла обобщённый интерфейс, и модуль представления, который отвечал бы за взаимодействие с пользователем. При замене последнего на другой, взаимодействующий с пользователем иным способом, программа должна продолжать корректно работать.

Задание 2
Повторите информацию о рассмотренных на уроке стандартных модулях. Ознакомьтесь также с модулями calendar, heapq, bisect, array, enum.

Рекомендуемые ресурсы

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

Статьи в Википедии о ключевых понятиях, рассмотренных на этом уроке
https://ru.wikipedia.org/wiki/Модуль_(программирование)

CyberBionic systematics

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