

Модули Python

№ урока: 7 **Курс:** Python Базовый

Средства обучения: Персональный компьютер/ноутбук стандартной производительности

Обзор, цель и назначение урока

Познакомиться с тем, что такое модули/пакеты. На уроке будет рассмотрено как пользоваться модулями в задачах, для чего они нужны и какие встроенные модули уже есть в Python 3. Так же будет рассмотрено как написать собственный модуль.

Изучив материал данного занятия, учащийся сможет:

- Понимать, что такое модуль/пакет и какие есть встроенные модули в Python 3.
- Уметь решать задачи применяя изученные модули.

Содержание урока

1. Какие есть модули в Python 3
2. Какие готовые функции в них есть
3. Решение задач с данными модулями

Резюме

- Модуль (пакет) — это набор готовых классов, функций для решения определенного рода задач.
- Для того чтобы пользоваться модулем, его нужно импортировать, используя ключевое слово `import` и указать какое имя модуля `module_name`. Таким образом вы импортируете сразу весь модуль со всеми его классами и функциями.
- По сути, модуль — это папка, в которой могут быть другие папки с файлами или сразу файлы. Допустим в модуле есть файл `functions.py`, в котором есть класс `Function`. Если вы хотите импортировать только этот класс, чтобы потом создать его объект, то используйте следующий синтаксис: `from my_module.functions import Function`. Аналогично со всеми другими структурами внутри модуля. Если бы `functions.py` лежал в модуле в папке `methods`, импортировать его надо было бы так: `from my_module.methods.functions import Function`.
- Если вам нужно сделать несколько импортов из одного модуля, вы можете просто перечислить их через запятую:
`from my_module.functions import function_1, function_2, function_x`
- Так же в модулях поддерживается `alias` (альтернативное имя). Например, если вы делаете импорт (любой), например:
`import my_long_module_name_module`, то вы можете воспользоваться `alias` и сделать вот так:
`import my_long_module_name_module as short_mod`
или вообще:
`import my_long_module_name_module as m`
и пользоваться уже в коде выбранным вами именем.

- Одни из самых важных базовых модулей: collections (для работы со структурами данных), itertools (для работы с функциями комбинаторики), datetime (работа с датами и временем), random (генерация случайных значений), math (математические функции), re (работа с регулярными выражениями).
- Чтобы написать свой модуль, достаточно создать папку с именем, которое вы хотите дать модулю, а в ней создать пустой файл `__init__.py`, после чего внутри вы уже можете создавать любую структуру. Только для каждого подмодуля (то есть мини модули внутри основного, это хорошая практика) аналогично нужно создавать в папке пустой файл `__init__.py`. Данный файл является своеобразным конструктором модуля как и в классах метод `__init__`. То есть он обязан быть и код в нем (если есть) будет выполнен как подготовительный скрипт перед импортом самого модуля или его функций, переменных или классов.

Закрепление материала

- Что такое пакет/модуль?
- Зачем нужны модули?
- Что нужно сделать чтобы написать собственный модуль?
- Какие модули были изучены на уроке? За что они отвечают?

Дополнительное задание

Изучите какие еще существуют встроенные модули в Python. Зачем они нужны? Какие в них есть полезные методы? Найдите минимум 3.

Самостоятельная деятельность учащегося

Напишите функцию-калькулятор. Она должна выполнять помимо базовых операций (+, -, /, *) еще следующие: возведение в степень, квадратный корень, логарифм по выбранному основанию. Функция должна принимать 3 аргумента: a, b, op, где a и b это числа для операций над ними, а op - имя операции в виде строки, понятной пользователю. Для того чтобы использовать функцию по имени, сделайте словарь (он может быть как глобальной видимости, так и внутри функции), где ключи — это предполагаемые названия функций, а значения — это функции (без вызова).

Например: `f = {"sum_a_b": lambda a,b: a+b}, f["sum_a_b"](2, 2)` будет 4. Аналогично для других базовых операций. Для операций извлечения корня и других, используйте только имя самой функции, например `{"sqrt": math.sqrt}` если вы импортировали весь пакет math или sqrt, если вы сделали `from math import sqrt`

Рекомендуемые ресурсы

- [Modules](#)
- [Модули 1](#)
- [Модули 2](#)
- [Модули 3](#)
- [Модули 4](#)