

Практика 9- Модули в языке Питон. Создание и работа с модулями

Полезные ссылки:

- [Сайт со всеми модулями.](#)

Модули – это дополнительные файлы, что можно импортировать внутрь других файлов Python. Каждый модуль содержит набор переменных и функций, которые можно использовать после подключения модуля к какому-либо файлу.

Модули можно создавать свои или использовать уже готовые модули Python. С полным перечнем всех стандартных модулей вы можете ознакомиться [по этой ссылке](#).

Все модули запомнить невозможно, а главное это не нужно. Вам необходимо лишь понимать какой модуль за что отвечает и в случае необходимости вы сможете посмотреть документацию, которой в интернете полным полно. К примеру, вы хотите создать архив или добавить что-либо в уже существующий? Тогда вам понадобится модуль `ZipFile`, который позволяет работать с архивами: создавать их, добавлять файлы, удалять файлы и делать другие вещи.

Таких модулей огромное множество и каждый отвечает за какие-либо функции и новшества.

Чтобы подключить модуль необходимо использовать команду `import` и указать название модуля.

Пример:

```
import time # Импорт одного модуля
```

```
import random, os # Импорт нескольких модулей
```

Также вы можете создавать свои модули. Для этого необходимо создать новый файл и по его названию подключать файл как модуль в другие файлы.

Команда From

Чтобы импортировать лишь некоторые методы или же переменные можно использовать команду `from`. Эта команда записывается следующим образом:

```
from math import ceil # Импортируем лишь метод ceil
```

Псевдонимы для модулей

Также при работе с модулями можно использовать псевдонимы. Это позволяет сократить название модуля, если оно слишком большое и вам не хочется писать его каждый раз.

Пример:

```
import Template as tem # Теперь можно обращаться к модулю как  
tem
```

Исходный код

Модули Питон

```
import datetime as d, sys, os, platform  
from math import sqrt as s, ceil  
  
# print(d.datetime.now().time())  
print(ceil(s(25)))
```

Свои модули

```
from mymodule import add_three_numbers as add  
  
print(add(5, 3, 0))
```

Забавный модуль

```
import cowsay as c  
c.cow('Goodbye my dear! :)')
```

Задание 1 - Модуль математики

Из модуля math импортируйте в проект следующее:

значение числа π

функцию ceil

Для функции сделайте псевдоним «с».

Задание 2 - Свой модуль

Создайте модуль и добавьте в него следующее:

переменную со значением 23.4

переменную со значением "Some value"

функцию, что считает сумму из двух переданных параметров

Импортируйте модуль в другой файл.

Задание 3 - Угадывание чисел

Создайте случайное число в диапазоне от 0 до 20, используя модуль random.

Подсказка: функция randint.

Просите пользователя угадать число до тех пор, пока он его не угадает.

Пример:

❖ Введите число от 0 до 20: 10

Вы не угадали!

Число что вы пытаетесь угадать больше

Введите число от 0 до 20: 15

Вы не угадали!

Число что вы пытаетесь угадать больше

Введите число от 0 до 20: 20

Вы не угадали!

Число что вы пытаетесь угадать меньше

Введите число от 0 до 20: 18

Вы угадали!



Задание 4 - Модуль datetime

Импортируйте модуль datetime. Он отвечает за работу с датой и временем.

Выведите на экран полную дату. Для этого используйте объект datetime и его метод now.

Из модуля импортируйте только объект datetime и добавьте к нему псевдоним.

Самостоятельные задания

Простые задачи (40 баллов)

Задача 1: Использование модуля math

Напишите программу, которая запрашивает у пользователя радиус круга и использует модуль math для расчёта и вывода площади этого круга.

Задача 2: Форматирование времени с помощью datetime

Напишите скрипт, который использует модуль datetime, чтобы вывести текущую дату и время в формате "гггг-мм-дд чч:мм:сс".

Задача 3: Генерация случайных чисел

Используя модуль `random`, напишите программу, которая генерирует и выводит три случайных числа в диапазоне от 1 до 10.

Задача 4: Системная информация с помощью `platform`

Напишите скрипт, который использует модуль `platform` и выводит версию Python, которую вы используете, а также информацию об операционной системе.

Задача 5: Работа с файловой системой с помощью `os`

Напишите программу, которая использует модуль `os` для вывода текущего рабочего каталога и затем меняет его на другой каталог, путь к которому пользователь вводит с клавиатуры.

Задача 6: Создание собственного модуля

Создайте модуль с именем `greetings.py`, который содержит две функции - `hello()` и `goodbye()`, которые просто выводят "Привет!" и "До свидания!" соответственно. Импортируйте этот модуль в другой файл и вызовите обе функции.

Сложные задачи (60 баллов)

Задание 1: Работа с модулем `os`

Напишите скрипт на Python, который использует модуль `os` для выполнения следующих действий:

- Выводит имя текущей операционной системы.
- Выводит список файлов в текущем рабочем каталоге.
- Создаёт новый подкаталог в текущем рабочем каталоге.
- Удаляет этот подкаталог.

Задание 2: Шифрование и дешифрование текста

- Создайте модуль `cipher.py`, который содержит функции `encrypt(text, shift)` и `decrypt(text, shift)`, реализующие простой шифр Цезаря для шифрования и дешифрования текста.
- Используйте этот модуль в другом файле для шифрования и дешифрования введённого пользователем текста.

Задание 3: Визуализация данных с использованием `matplotlib`

Используйте модуль `matplotlib` для визуализации данных. Напишите скрипт, который:

- Генерирует случайный набор данных (например, оценок студентов).

- Строит гистограмму распределения этих данных.
- Сохраняет гистограмму в файл на диске.

Задание 4: Работа с модулем requests

Напишите скрипт, который использует модуль requests для отправки GET-запроса к какому-нибудь API (например, к публичному API для получения данных о погоде) и обработки полученного ответа (напечатать результат в консоль).

Задание 5: Создание CLI (интерфейса командной строки)

Используя модуль argparse, напишите скрипт, который принимает аргументы командной строки для выполнения различных задач (например, копирование файла, вывод списка файлов в каталоге, вычисление хеш-суммы файла).

Задание 6: Модуль json для работы с JSON-файлами

Создайте скрипт, который считывает JSON-файл, модифицирует данные и записывает изменения обратно в файл. Используйте модуль json для анализа и генерации JSON.

Задание 7: Создание простого веб-сервера с http.server

Используя модуль http.server, напишите скрипт, который запускает простой веб-сервер на вашем компьютере, который может отдавать статические файлы и отображать простую веб-страницу.