

МОДУЛИ

Собственный модуль

Модуль — это просто файл с кодом Python, например `my_module.py`. Он может содержать функции, классы и переменные.

Пример

```
# my_module.py

def greet(name):
    return f"Привет, {name}!"

PI = 3.1415
```

Использование:

```
# main.py

import my_module

print(my_module.greet("Дмитрий"))
print(my_module.PI)
```

Практикум

1. Создайте файл `math_utils.py`, добавьте в него функцию для расчёта факториала.
2. В `main.py` импортируйте `math_utils` и вызовите вашу функцию.

Пакеты

Пакет — это папка с модулями и файлом `__init__.py`. Это позволяет организовывать код по каталогам.

Пример

```
my_package/
├── __init__.py
├── geometry.py
└── algebra.py
```

Файл `__init__.py` — специальный файл, который обозначает папку как пакет Python. Он может быть:

- пустым (если просто нужно указать, что это пакет),
- содержать код (например, инициализацию, импорт или настройки по умолчанию).

Что делает `__init__.py`?

1. Позволяет импортировать модули из папки как из пакета.
2. Может объединять содержимое модулей, чтобы облегчить доступ.
3. Может использоваться для инициализации данных при первом импорте пакета.

Пример 1: Пустой `__init__.py`

```
# __init__.py
```

Позволяет использовать:

```
from my_package import module_a
```

Пример 2: Импорт по умолчанию

```
# __init__.py

from .module_a import func_a
from .module_b import func_b

Теперь можно обращаться так:
from my_package import func_a, func_b
```

Пример 3: Инициализация

```
# __init__.py

print("Пакет my_package инициализирован")

config = {
    "version": "1.0",
    "author": "Вы"
}
```

geometry.py:

```
def area_circle(r):  
    return 3.14 * r ** 2
```

main.py:

```
from my_package import geometry  
  
print(geometry.area_circle(5))
```

Практикум

1. Создаём пакет **text_utils** с файлами:
 - string_ops.py (функции для подсчёта слов, символов)
 - validators.py (функции для проверки email и паролей)
2. В **__init__.py** запишем:

```
from .string_ops import count_words  
from .validators import is_valid_email
```

3. В **validators.py** будет следующий код:

```
import re  
  
def is_valid_email(email: str) -> bool:  
    """Проверка, является ли строка  
    корректным email-адресом."""  
    pattern = r'^[\w\.-]+@[\w\.-]+\.\w+$'  
    return re.match(pattern, email) is not None
```

4. В **main.py** будет реализация:

```
from text_utils import count_words, is_valid_email  
  
print(count_words("Привет, мир!"))  
print(is_valid_email("test@example.com"))
```

5. Тестируем main.py.

Обзор полезных стандартных модулей Python

Модуль	Назначение
os	Работа с операционной системой
sys	Работа с системными параметрами
datetime	Дата и время
random	Случайные значения
math	Математические функции
json	Работа с JSON
re	Регулярные выражения

Пример

```
import random

print(random.randint(1, 100))
```

Установка стороннего модуля

Сторонние модули устанавливаются через pip:

```
pip install requests
```

Затем можно использовать их в коде:

```
import requests

response = requests.get('https://api.github.com')
print(response.json())
```

Работа с графикой и документами

Работа с Excel (модуль openpyxl)

pip install openpyxl

```
from openpyxl import Workbook

wb = Workbook()
ws = wb.active
ws['A1'] = "Привет"
ws['B1'] = 123
wb.save("example.xlsx")
```

Работа с DOCX (модуль python-docx)

pip install python-docx

```
from docx import Document

doc = Document()
doc.add_heading('Заголовок', 0)
doc.add_paragraph('Пример текста в документе.')
doc.save('example.docx')
```

Работа с Markdown

Для работы с Markdown можно использовать:

pip install markdown

Пример:

```
import markdown

text = "# Заголовок\n\nЭто **жирный текст**."
html = markdown.markdown(text)

with open("output.html", "w", encoding="utf-8") as f:
    f.write(html)
```

Практикум по документам

1. Создайте Excel-файл с 3 строками данных о студентах (имя, возраст, балл).
2. Сгенерируйте .docx документ с отчётом по этим данным.
3. Преобразуйте Markdown-файл в HTML.

Заключение

Теперь Вы умеете:

- создавать свои модули и пакеты,
- пользоваться стандартной библиотекой,
- устанавливать сторонние модули,
- работать с графикой и файлами DOC/XLS/MD.