Модули

Докладчик: Евграфов Михаил

Модули

Мотивация использования

```python
$ python
>>> def print_greeting(name: str) ->
None:
...     greeting = f"Hello, {name}!"
...     print(greeting)
...
>>> print_greeting("Saul Goodman")
...
>>> exit()
$ python
>>> print_greeting("Lalo")
...
NameError: name 'print_greeting' is not defined
```

Создание модуля

```python
# greeting.py
def print_greeting(name: str) -> None:
    greeting = f"Hello, {name}!"
    print(greeting)
```

```python
>>> import greeting
>>> print(type(greeting).__name__)
module
```

Доступ к содержимому

```
>>> import greeting
>>> print(greeting.__name__)
>>> print_greeting("Gustavo Fring")
...
NameError: name 'print_greeting' is not defined
>>> greeting.print_greeting("Gustavo Fring")
Hello, Gustavo Fring!
```

Этап выполнения модуля

```python
# greeting.py
print("start import")
def print_greeting(name: str) -> None:
    greeting = f"Hello, {name}!"
    print(greeting)


print("finish import")


>>> import greeting
start import
finish import
```

Импорт отдельных объектов

```python
# greeting.py
GREET_TEMPLATE = "Hello, {}!"


def print_greeting(name: str) -> None:
    greeting = GREET_TEMPLATE.format(name)
    print(greeting)


>>> from greeting import print_greeting
>>> print_greeting("Hector")
Hello, Hector!
>>> print(greeting)
...
NameError: name 'greeting' is not defined
```

Связывание модуля

```python
# greeting.py
GREET_TEMPLATE = "Hello, {}!"

def print_greeting(name: str) -> None:
    greeting = GREET_TEMPLATE.format(name)
    print(greeting)


>>> import greeting as grt
>>> grt.print_greeting("Hector")
Hello, Hector!
>>> print(greeting)
...
NameError: name 'greeting' is not defined
```

Связывание содержимого

```python
# greeting.py
GREET_TEMPLATE = "Hello, {}!"


def print_greeting(name: str) -> None:
    greeting = GREET_TEMPLATE.format(name)
    print(greeting)


>>> from greeting import GREET_TEMPLATE as template
>>> print(template)
Hello, {}!
>>> print(GREET_TEMPLATE)
...
NameError: name 'GREET_TEMPLATE' is not defined
```

from … import *

```python
# greeting.py
GREET_TEMPLATE = "Hello, {}!"

def print_greeting(name: str) -> None:
    greeting = GREET_TEMPLATE.format(name)
    print(greeting)


>>> from greeting import *
>>> print(GREET_TEMPLATE)
>>> print_greeting("Mike")
Hello, {}!
Hello, Mike!
```

Проблемы from ... import *

```python
# greeting.py
GREET_TEMPLATE = "Hello, {}!"

def print_greeting(name: str) -> None:
    greeting = GREET_TEMPLATE.format(name)
    print(greeting)


>>> from greeting import *
>>> GREET_TEMPLATE = "wrong"
>>> print(GREET_TEMPLATE)
>>> print_greeting("Mike")
wrong
Hello, Mike!
```

Приватные поля модуля

```python
# greeting.py
def print_greeting(name: str) -> None:
    greeting = _greeting_template.format(name)
    print(greeting)


_greeting_template = "Hello, {}!"


>>> from greeting import _greeting_template
>>> print(_greeting_template)
Hello, {}!
```

Приватные поля модуля и *

```python
# greeting.py
def print_greeting(name: str) -> None:
    greeting = _greeting_template.format(name)
    print(greeting)


_greeting_template = "Hello, {}!"


>>> from greeting import *
>>> print_greeting("Mike")
>>> print(_greeting_template)
Hello, Mike!
...
NameError: name '_greeting_template' is not defined
```

Выполнение модуля как скрипта

```python
# greeting.py
def print_greeting(name: str) -> None:
    greeting = f"Hello, {name}!"
    print(greeting)


name = input("Enter your name: ")
print_greeting(name)
```

```
$ python greeting.py
```

__pycache__

```
└──greeting.py                                    T1


$ python greeting.py



├────__pycache__
│       │
│       │
│       └────greeting.cpython-311.pyc    T1
│
│
│
└────greeting.py                                  T1
```
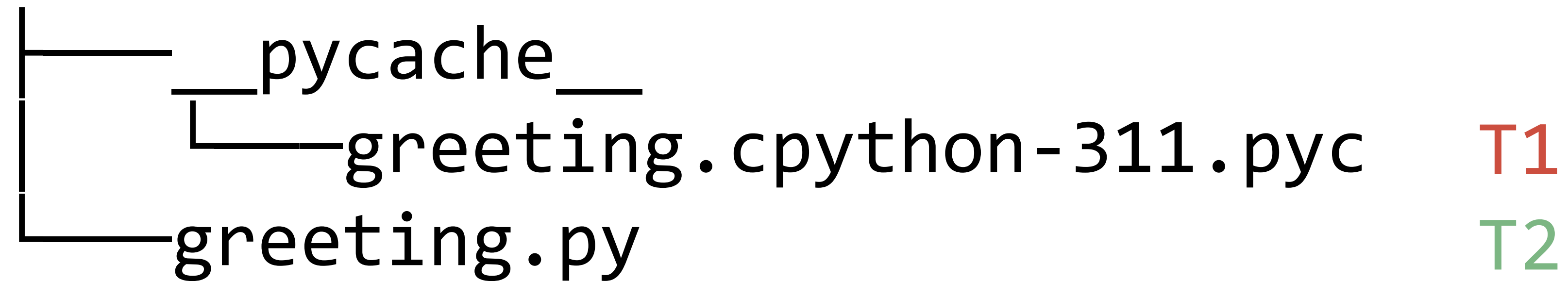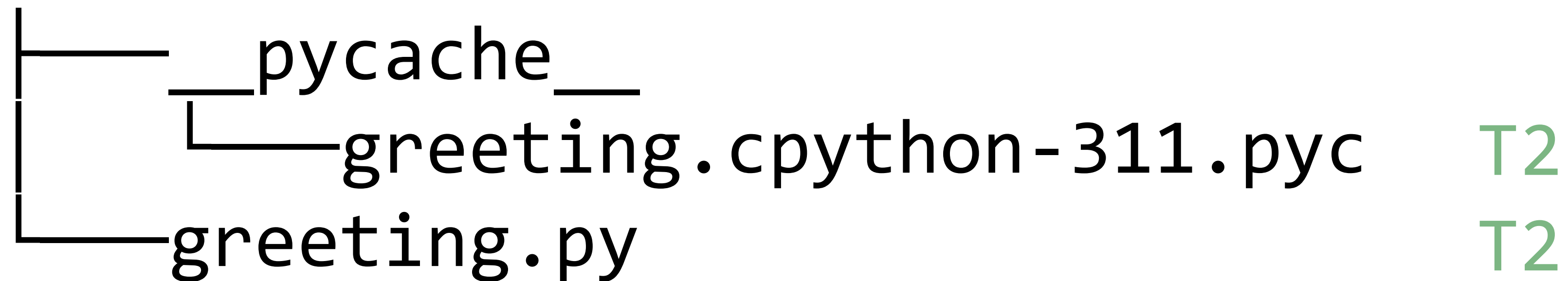
Обновление __pycache__

```
├───__pycache__
│    └───greeting.cpython-311.pyc    T1
└───greeting.py                      T1
```

Обновление __pycache__

```
├──────__pycache__
│      └──────greeting.cpython-311.pyc    T1
└──────greeting.py                        T2
```

```
$ python greeting.py
```

```
├──────__pycache__
│      └──────greeting.cpython-311.pyc    T2
└──────greeting.py                        T2
```

Порядок поиска модулей

```python
# greeting.py
from typing import Optional
from third_party import cool_greeting

def print_greeting(
    name: Optional[str] = None
) -> None:
    greeting = f"Hello, {name}!"
    print(greeting)
    cool_greeting()
```

Порядок поиска модулей

Директории

```
$ python greeting.py

>>> from typing import ...        ./        ?
```

Порядок поиска модулей

Директории

```
$ python greeting.py

>>> from typing import ...        ./          NO

                                  .../Python/...   ?
```

Порядок поиска модулей

```
$ python greeting.py

>>> from typing import ...        ./              NO

                                  .../Python/...  YES
```

Порядок поиска модулей

Директории

```
$ python greeting.py

>>> from typing import ...        ./         ?

>>> from third_party import ...
```

Порядок поиска модулей

Директории

```
$ python greeting.py

>>> from typing import ...        ./              NO

>>> from third_party import ... .../Python/...   ?
```

Порядок поиска модулей

Директории

```
$ python greeting.py

>>> from typing import ...        ./            NO

>>> from third_party import ... .../Python/...  NO

                                .../site-packages/...  ?
```

Порядок поиска модулей

```
$ python greeting.py

>>> from typing import ...          ./              NO

>>> from third_party import ... .../Python/...  NO

                                .../site-packages/...   YES
```

builtins

```
>>> import greeting
>>> greeting.__builtins__
{
    ...
    'abs': <function abs(x, /)>,
    'all': <function all(iterable, /)>,
    'any': <function any(iterable, /)>,
    ...
}
>>> import builtins
>>> greeting.__builtins__["len"] is builtins.len
True
```

Исполняемые модули

__name__

```
# name_printer.py
print(__name__)

$ python name_printer.py
__main__

$ python
>>> import name_printer
>>> print(name_printer.__name__)
name_printer
name_printer
```

Идиома __name__ == "__main__"

```python
# name_printer.py
if __name__ == "__main__":
    print(__name__)
```

```
$ python name_printer.py
__main__

$ python
>>> import name_printer
>>> print(name_printer.__name__)
name_printer
```

Идиома - не точка входа

Мотивация использования

```python
# argv_user.py
import sys

if __name__ == "__main__":
    print(sys.argv)
```

```
$ python argv_user.py
['.\\argv_user.py']

$ python argv_user.py 42
['.\\argv_user.py', '42']
```
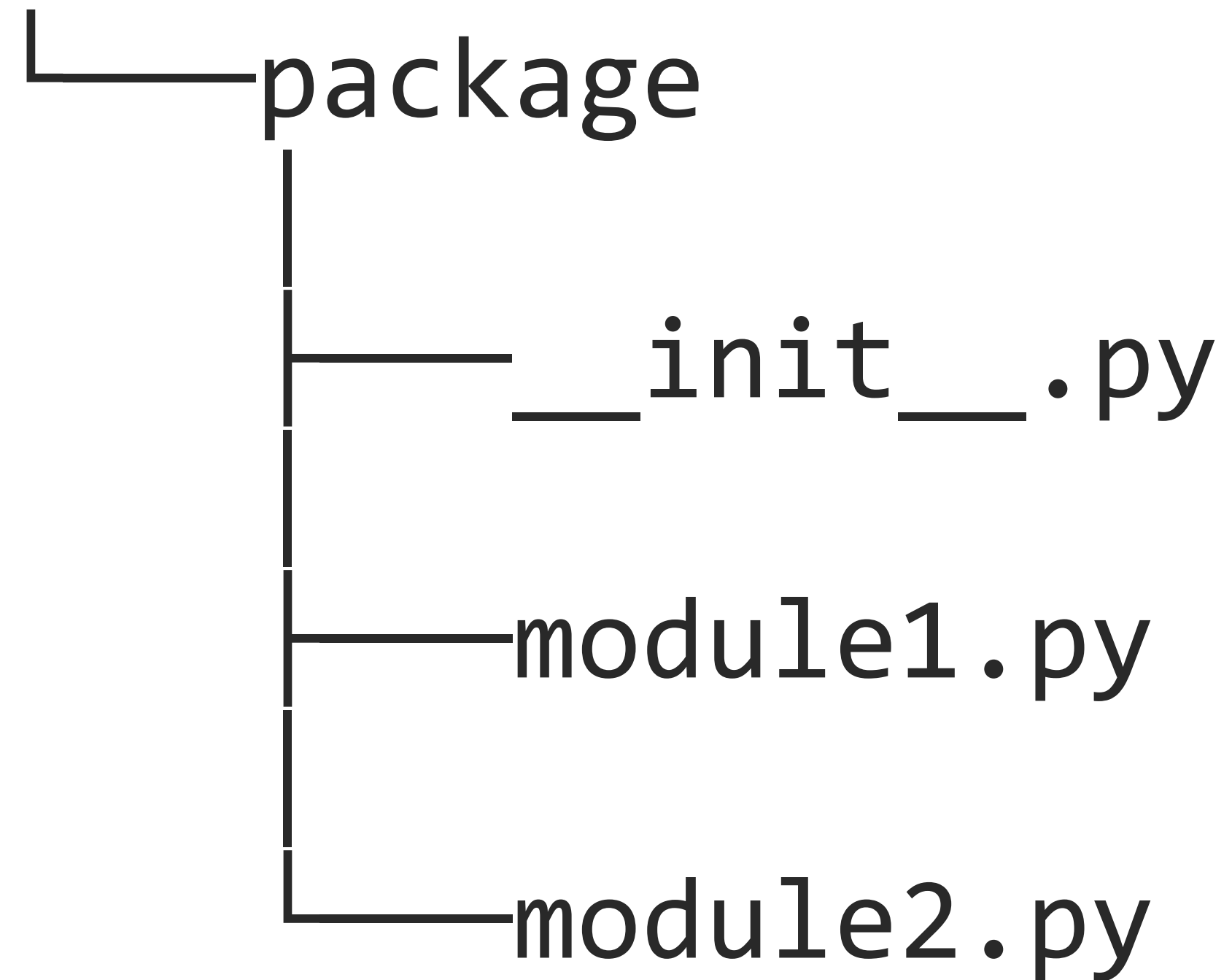
Пакеты

Файловая структура

```
└──package
    │
    ├──__init__.py
    │
    ├──module1.py
    │
    └──module2.py
```

Импорт модулей пакета

```python
# package/module1.py
def print_module_info() -> None:
    print(f"I am function from {__name__}")


# package/module2.py
def print_module_info() -> None:
    print(f"I am function from {__name__}")


>>> import package
>>> package.module1
...
AttributeError: module 'package' has no attribute
'module1'
```

Импорт модулей пакета

```python
# package/module1.py
def print_module_info() -> None:
    print(f"I am function from {__name__}")


# package/module2.py
def print_module_info() -> None:
    print(f"I am function from {__name__}")



>>> import package.module1 as mod1
>>> mod1.print_module_info()
I am function from package.module1
```

__init__.py

```python
# package/__init__.py
from package import module1
from package import module2

print("Init package")


>>> import package
>>> print(package.module1)
>>> print(package.module2)
Init package
<module 'package.module1' ...>
<module 'package.module2' ...>
```

import ... from *

```
>>> from package import *
>>> module1.print_module_info()
>>> module2.print_module_info()
...
NameError: name 'module1' is not defined
```

__all__

```python
# package/__init__.py
__all__ = ["module1", "module2"]


>>> from package import *
>>> module1.print_module_info()
>>> module2.print_module_info()
I am function from package.module1
I am function from package.module2
```

Семинар