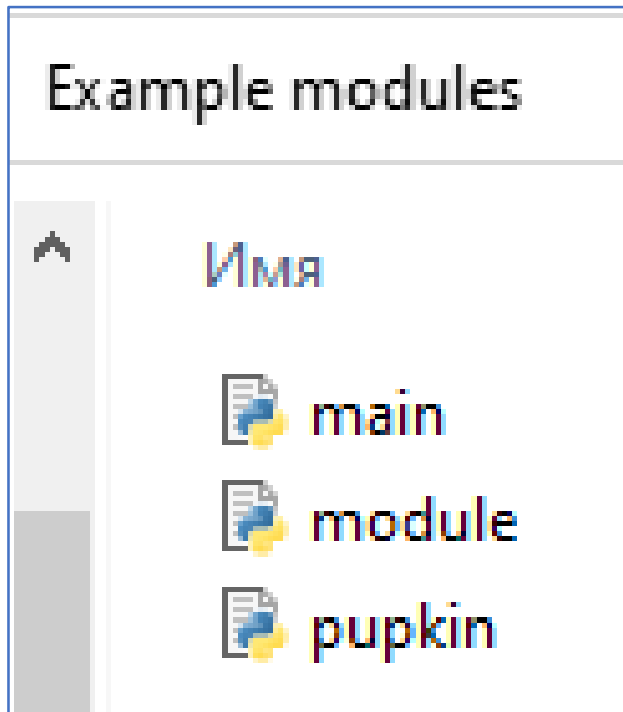


#1 Как создать модуль в Python

Home work 7_1



Stefan Zhauryd
Instructor



Шаг 1) Создать папку **Example modules**

Шаг 2) Создайте основной файл вашей программы
main.py

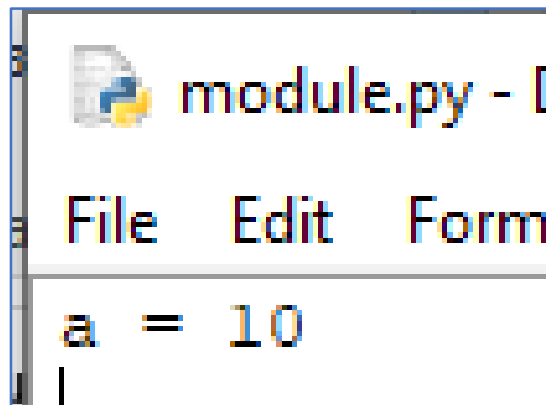
и создайте необходимые модули:

module.py

pupkin.py

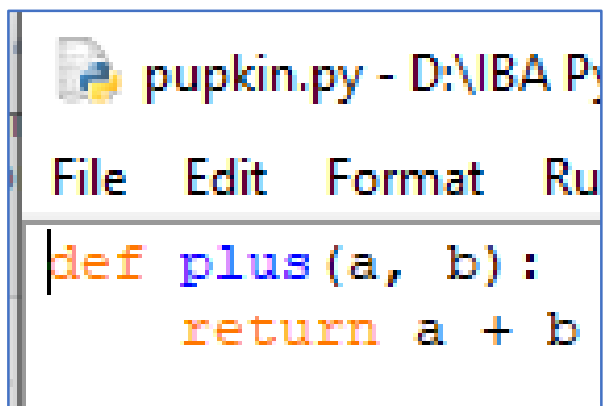


Шаг 3) Наполним модуль необходимыми функциями



```
module.py - [
File Edit Form
a = 10
|
```

В файл **module.py** положим переменную 'a'



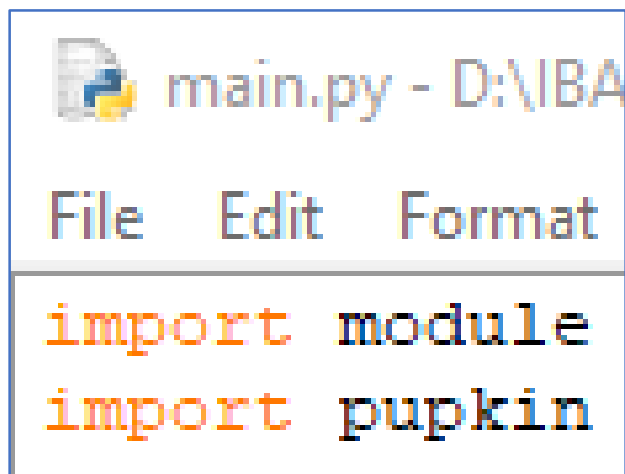
```
pupkin.py - D:\IBA Py
File Edit Format Ru
def plus(a, b):
    return a + b
```

В файл **pupkin.py** положим функцию 'plus'

```
if __name__ == "__main__":
    print("I prefer to be a module")
else:
    print("I like to be a module")
```

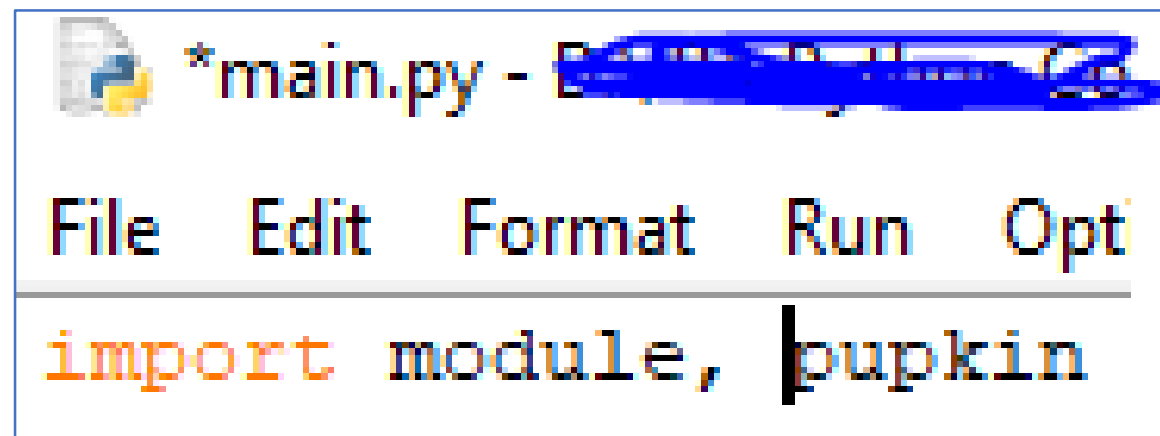
← At the
end of
each file

Шаг 4) Импортируем в нашу программу **main.py** модули:
module.py и **pupkin.py**



```
main.py - D:\IBA
File Edit Format
import module
import pupkin
```

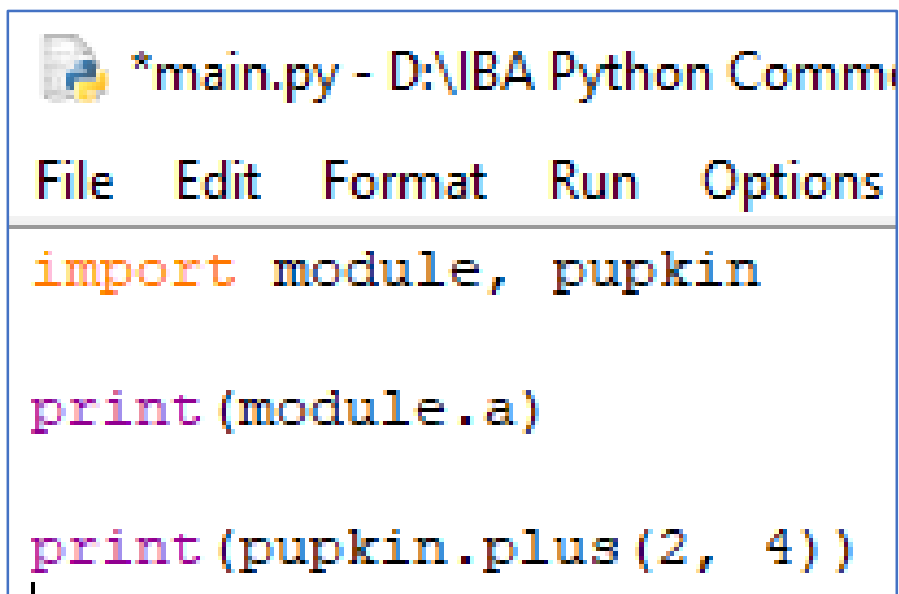
либо



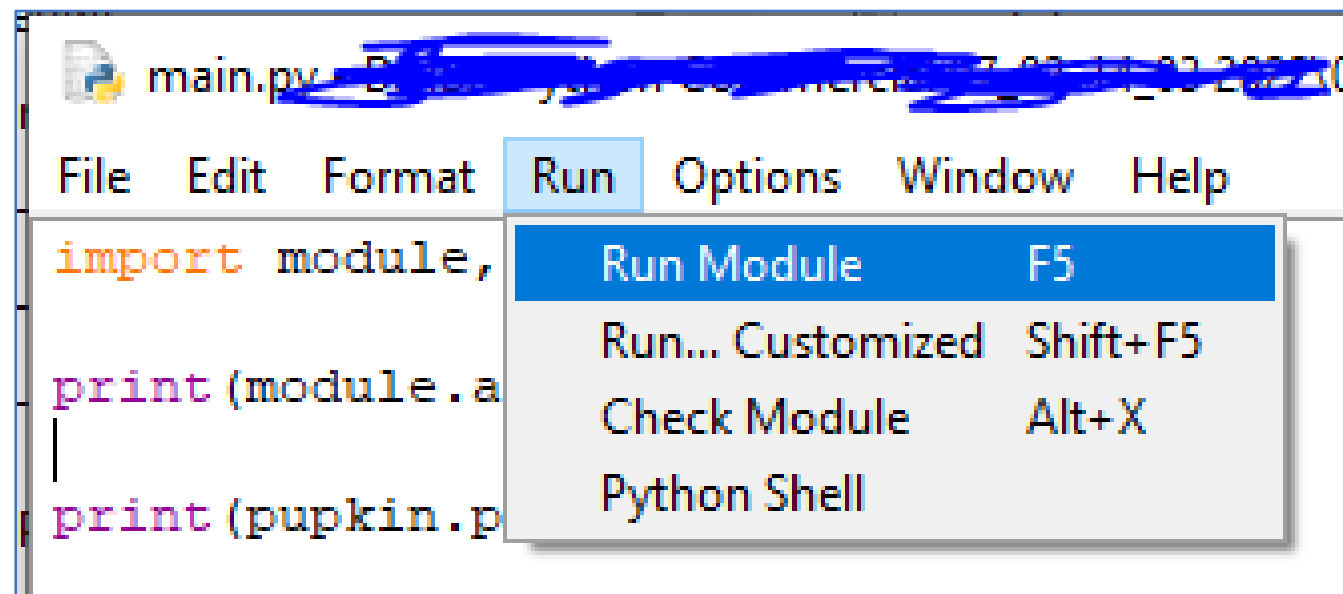
```
*main.py - D:\IBA
File Edit Format Run Opt
import module, pupkin
```



Шаг 5) Воспользуемся функционалом определенном в наших модулях



```
*main.py - D:\IBA Python Comm...
File Edit Format Run Options
import module, pupkin
print(module.a)
print(pupkin.plus(2, 4))
```



```
main.py - D:\IBA Python Comm...
File Edit Format Run Options Window Help
import module,
print(module.a)
print(pupkin.p
```

```
if __name__ == "__main__":
    print("I prefer to be a module")
else:
    print("I like to be a module")
```

← At the
end of
each file

Шаг 6) В результате увидим что всё сработало:

```
10
```

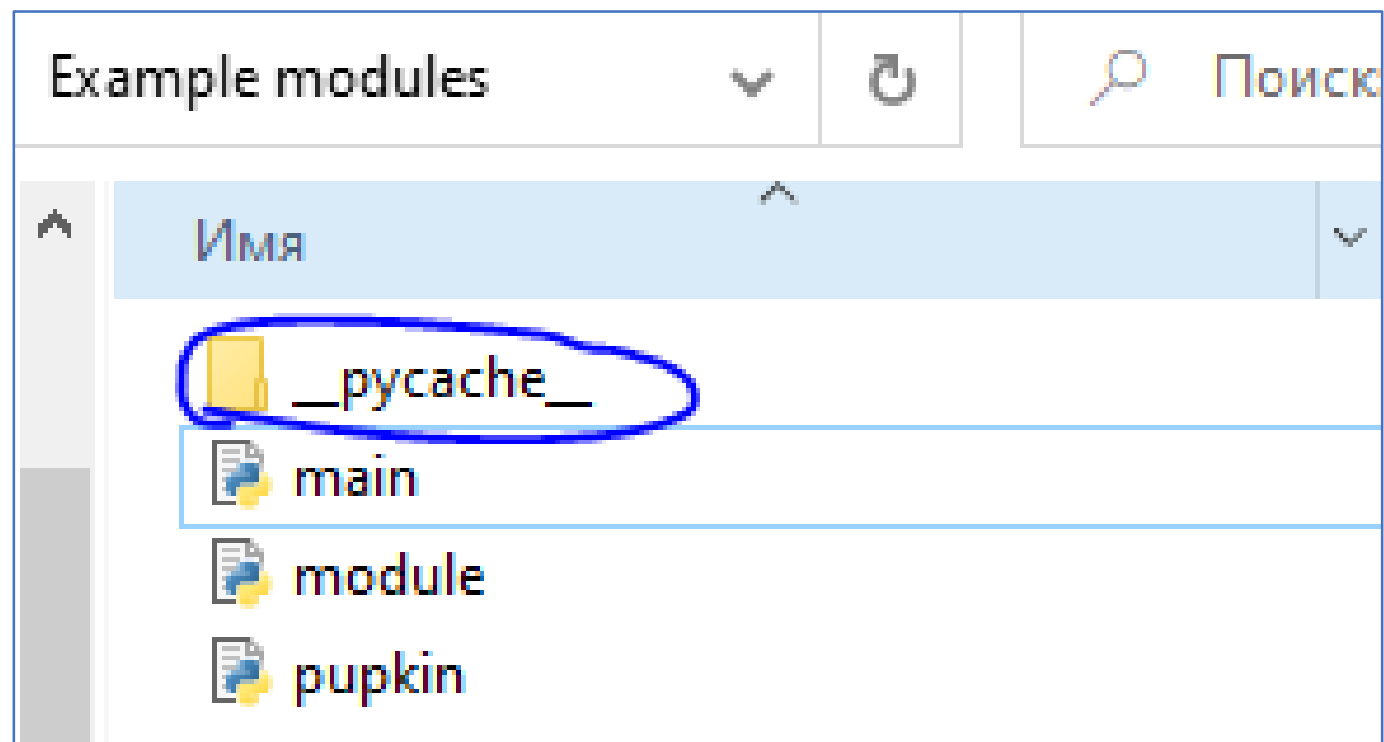
```
6
```

```
>>>
```



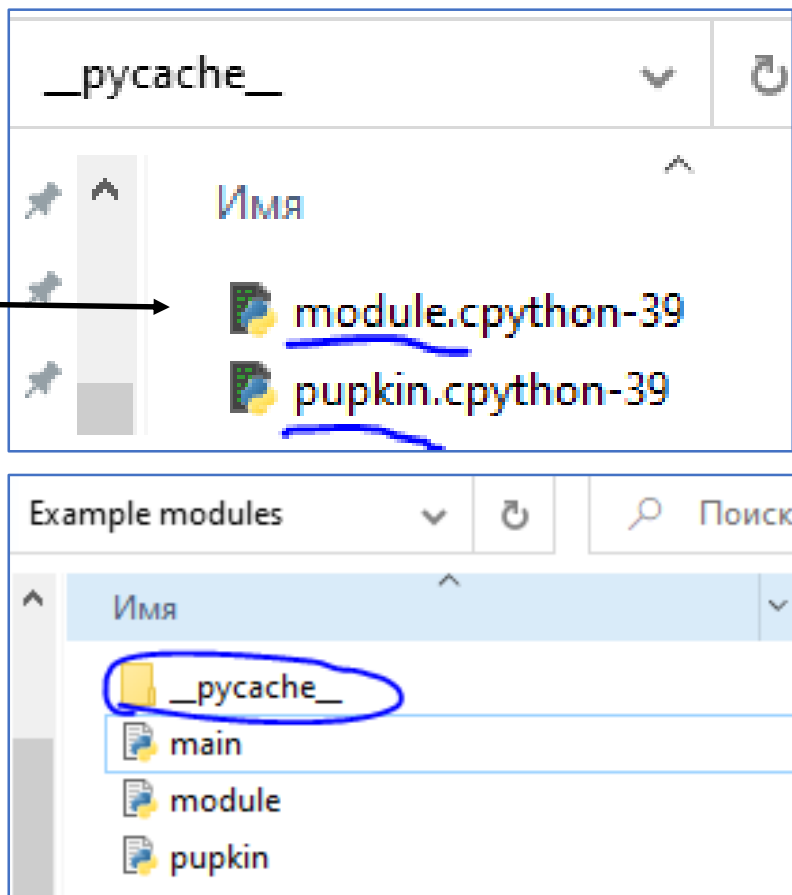
Шаг 7) Проверим нашу папку **Example modules**

Как видно появилась папка **__pycache__**



Шаг 8) Заглянем в папку `__pycache__`

это байткод с
расширением `.pyc`



Импорт модуля – относительно дорогостоящее мероприятие, поэтому Python предпринимает некоторые трюки для ускорения этого процесса. Один из способов – создать байт-компилированные файлы (или байткод) с расширением `.pyc`, которые являются некой промежуточной формой, в которую Python переводит программу.

Такой файл `.pyc` полезен при импорте модуля в следующий раз в другую программу – это произойдёт намного быстрее, поскольку значительная часть обработки, требуемой при импорте модуля, будет уже проделана. Этот байткод также является платформо-независимым.



Важно

Обращайте внимание на имена файлов, имена функций и переменных. Это важно при использовании модулей

