

## Модули

- python файлы в папке с текущим модулем или в sys.path
- .py => {.pyc, .pyo}
- .pyo = .pyc - asserts
- .pyd / .so
- `import` mod\_name
- `import` mod\_name `as` new\_name
- `from` mod\_name `import` obj\_name, obj\_name2
- `from` mod\_name `import` obj\_name `as` obj\_name2
- `from` mod\_name `import` \* все кроме `__.*` или все из `__all__`

## Пример модуля - mod.py

```
1  __all__ = ["some_val", "func"]
2
3  some_val = 16
4
5  def func(x):
6      return x + some_val
```

## Использование

```
1  from mod import some_val, func
2
3  print func(1)
```

- `sys.path`
- `sys.modules`
- `__import__`
- `importlib.import_module`
- `imp`

\_\_import\_\_

```
1    fc = open(module_file_path).read()
2    module_code = compile(fc, module_file, "exec")
3    loc = {}
4    eval(module_code, loc)
5    sys.modules[module_name] = ModuleObject(loc)
6    func = loc['func']
```

## Подмена модуля

```
1 class SomeClass( object ):
2     def m( self ):
3         return 1
4
5 sys.modules[ 'mod' ] = SomeClass()
6 from mod import m
7
8 print m() # 1
```

## Пакеты

```
xml\  
__init__.py  
  expat\  
    __init__.py  
    bindings.py  
  dom\  
    __init__.py  
    ...  
  sax\  
    __init__.py  
    ...  
  xmltree\  
    __init__.py  
    XmlTree.py  
    ....
```

## Пакеты

- `import xml` => `xml/__init__.py`
- `import xml.expats` => `xml/expats/__init__.py`
- Циклические импорты
- `package.__path__`

- `__autor__`
- `__name__`
- `__doc__`

```
1 import sys
2
3 def main( argv=None ):
4     argv = argv if args is not None else sys.argv
5     ...
6     return code
7
8 if __name__ == "__main__":
9     sys.exit( main() )
```



- `python -m ....`