Importing

Guttag 7

Goals

Learn terms related to importing in python

- Module
- package
- dot notation

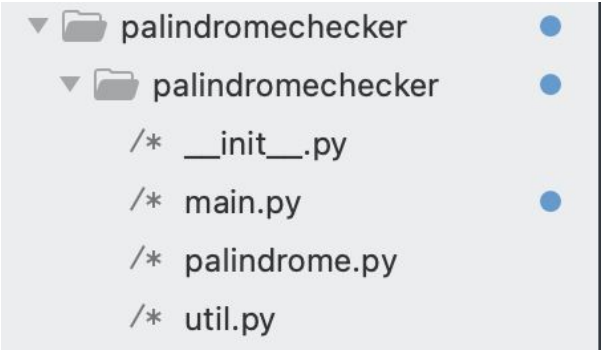
Terms: Module

Definition

- a module in python is a file that ends in .py

Examples from palindromes lab

- main.py
- util.py
- palidrome.py



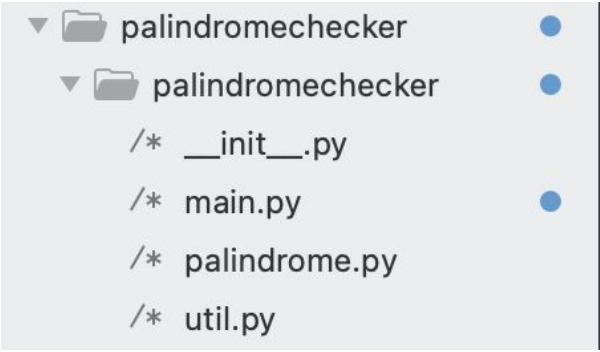
Terms: Package

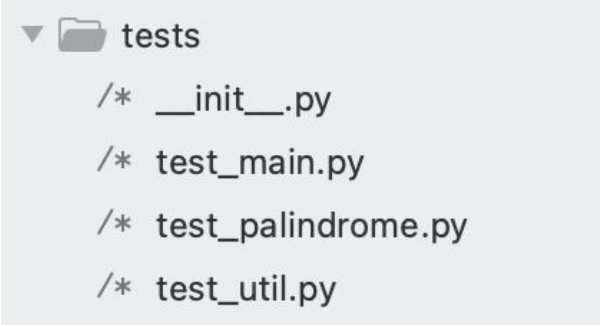
Definition

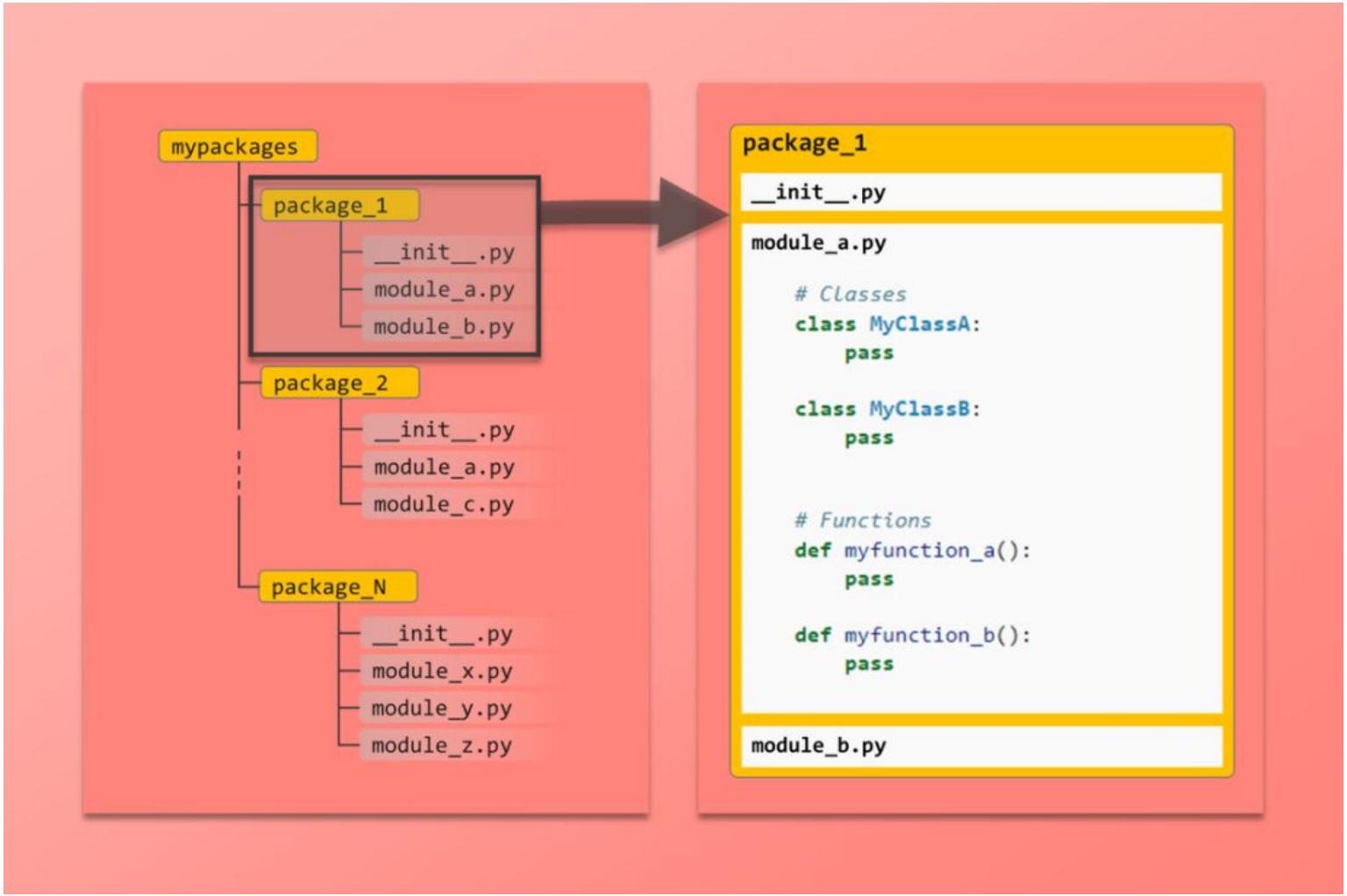
- a folder containing modules and an __init__ module

Examples from palindromes lab

- innermost palindromechecker directory
- tests directory






mypackages
package_1
__init__.py
module_a.py
module_b.py
package_2
__init__.py
module_a.py
module_c.py
package_N
__init__.py
module_x.py
module_y.py
module_z.py
package_1
__init__.py
module_a.py
# Classes
class MyClassA:
pass
class MyClassB:
pass
# Functions
def myfunction_a():
pass
def myfunction_b():
pass
module_b.py

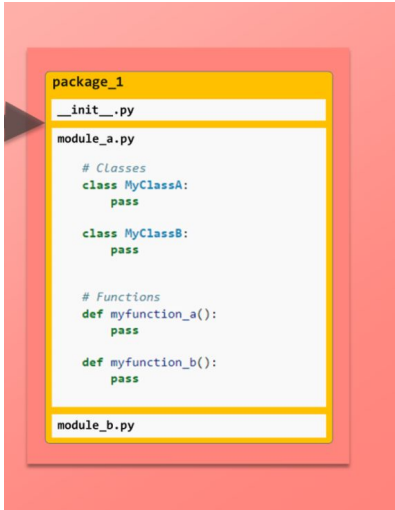
Terms: Symbol

Definition

- Anything defined within a module!
- recall, defined **variables** appear on the left-hand side of =
- recall, defined **functions** appear after keyword **def**
- we will also soon learn about **classes** appearing after keyword **class**

Examples

- **a** = 10
- **cli** = typer.Typer()
- def **is_prime**(n: int) -> bool:
- class **PalindromeCheckingApproach**(str, Enum):
- class **MyClassA**:
- class **MyClassB**:



Terms: Namespace & Fully-Qualified Name

Definition

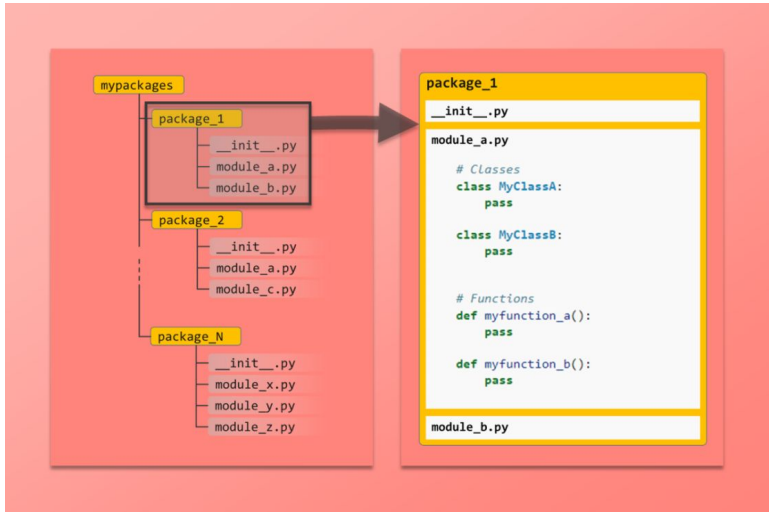
- Namespace refers to the module name
- Fully-Qualified Name specifies the Namespace and the symbol with dot notation

Examples

- module_a.MyClassA
- module_a.MyClassB

Further info

- Sometimes package names are included
- package_1.module_a.MyClassA



Terms: Import

Definition

- the python syntax used to "make available" symbols defined in different modules or packages.

Example

- import typing
- import random
- import typer

Further info

- The direct imports as shown above work for libraries, including the standard python libraries (included libraries)

Import Syntax

import LIBRARY

- import typing

from LIBRARY import MODULE

- from typing import List

import MODULE as ALIAS

- import numpy as np

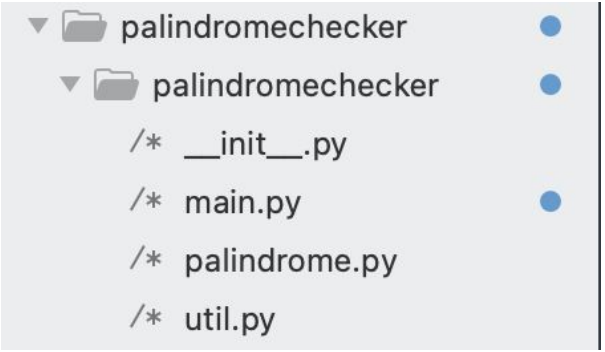
Import Syntax Continued

from PACKAGE import MODULE

- from palindromechecker import util

from PACKAGE.MODULE import SYMBOL

- from palindromechecker.util import human_readable_boolean



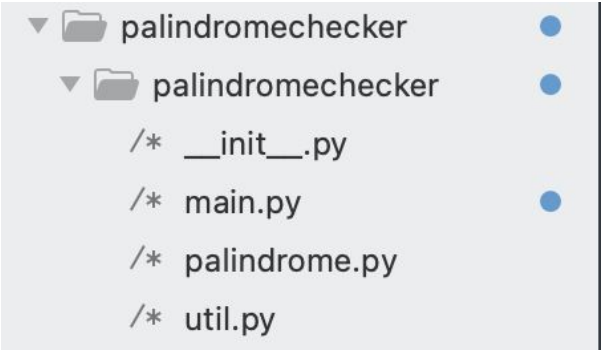
Import Syntax Continued

from PACKAGE import MODULE

- from palindromechecker import util ← **util.human_readable_boolean ****

from PACKAGE.MODULE import SYMBOL

- from palindromechecker.util import human_readable_boolean



Dot Notation

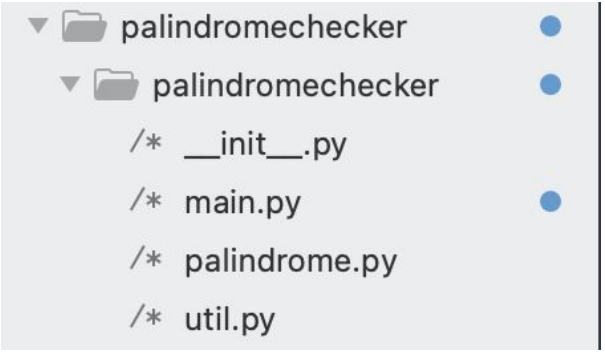
from PACKAGE import MODULE

- from palindromechecker import util ← **util.human_readable_boolean ****

from PACKAGE.MODULE import SYMBOL

- from palindromechecker.util import human_readable_boolean

Depending on the import statement, symbols inside a module must be accessed with dot notation



Dot Notation

Consider this import statement:

from package import module

Question:

How should all symbols in the module be accessed?

Dot Notation

Consider this import statement:

from package import module

Answer: use dot notation

module.symbol

Dot Notation

- from palindromechecker import util

util.human_readable_boolean(.....

- from palindromechecker.util import human_readable_boolean

human_readable_boolean(.....

- import Typing

Typing.List, Typing.Callable

- from Typing import List

List