

# Module và Package trong Python

## Tìm hiểu về Module trong Python

TS. Trần Quang Quý.  
ĐT: 0818981166(Zalo)

Bộ môn Khoa học Máy tính và Công nghệ  
Khoa Công nghệ Thông tin

Tháng 09 năm 2021



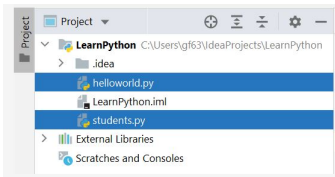
# Khái niệm module

- Trong nội dung này chúng ta sẽ học cách tạo module trong Python, qua đó các em sẽ hiểu được khái niệm module là gì? Làm thế nào để gọi đến một module tạo mới hoặc module có sẵn trong Python. Trong Python, modules là những file có phần mở rộng .py, các file này chứa mã Python và đó có thể là các biến, hàm, hoặc một lớp nào đó.
- Các module giúp chúng ta code một cách linh hoạt hơn, mỗi file sẽ chứa những đoạn code phục vụ cho một chức năng cụ thể, được sắp xếp tùy vào cách thiết kế của mỗi lập trình viên.
- Để sử dụng các hàm của module A trong module B thì ta phải sử dụng từ khóa *import*, sau đó là bạn có thể sử dụng toàn bộ tài nguyên của module A.



# Khởi tạo module

- Trước khi tạo một module thì bạn phải xác định rằng nhu cầu cần tạo một module tên gì? nhằm phục vụ một công việc gì? Để từ đó đặt tên module cho có ngữ nghĩa.



- File helloworld.py là file chương trình chính của project, còn file students.py là module chứa những hàm quản lý sinh viên chúng ta sắp viết vào.



# Lệnh import trong Python

- Để sử dụng các đoạn code ở module A vào trong module B thì ta sẽ phải sử dụng lệnh **import**. Có hai cách sử dụng, thứ nhất là dùng lệnh **import** đơn lẻ, thứ hai là lệnh **from-import**.
- Lệnh import sẽ gọi module A vào trong module B, lúc này bạn có thể sử dụng mọi tài nguyên của module A trong module B. Chúng ta có thể import một hoặc module. Ví dụ: import module hoặc import module1,module2,..... module n.
- Sử dụng **from-import**: Giả sử trong module A bạn định nghĩa 10 function, nhưng trong module B bạn chỉ muốn sử dụng 1 trong 10 function đó thôi thì sử dụng from .. import. Vậy sự khác nhau giữa import và from ... import là một bên sẽ gọi tất cả các function, còn một bên chỉ gọi một function nào đó thôi. Cú pháp: from < module-name> import <name 1>, <name 2>..,<name n>
- Trường hợp bạn muốn import tất cả các function thì sử dụng dấu sao \*. from <module> import \*



# Đổi tên module với **as** trong Python

- Nếu bạn muốn đổi tên module cho ngắn gọn và dễ hiểu thì có thể sử dụng từ khóa **as**. Từ khóa này rất hữu ích vì giúp bạn tiết kiệm được thời gian nhập những module có tên quá dài, thay vào đó chỉ cần một cái tên thật đặc biệt. Ví dụ: `import <module-name> as <specific-name>`

#Tên của module calculation ở ví dụ trước sẽ đổi thành cal.

```
import calculation as cal
a = int(input("Enter a?"))
b = int(input("Enter b?"))
print("Sum = ", cal.summation(a,b))
```

- Hàm **dir()** có công dụng trả về danh sách các thuộc tính và phương thức của module. Ví dụ: Xem những tên biến, hàm có sẵn trong module **json**.

```
import json
List = dir(json)
```

```
print(List)
```

```
['JSONDecoder', 'JSONEncoder', '__all__', '__author__', '__builtins__', '__cached__', '__doc__',
 '__file__', '__loader__', '__name__', '__package__', '__path__', '__spec__', '__version__',
 '_default_decoder', '_default_encoder', 'decoder', 'dump', 'dumps', 'encoder', 'load', 'loads', 'scanner']
```

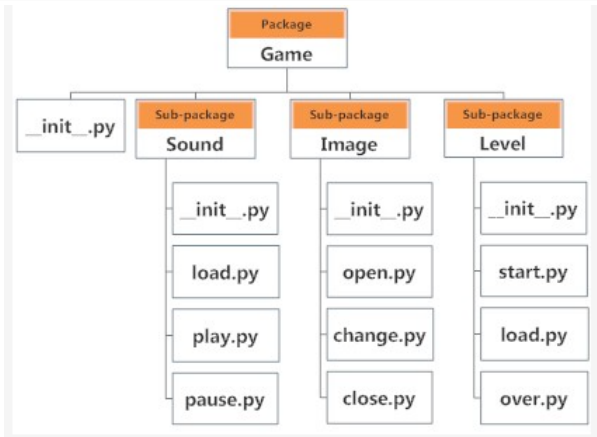


# Package trong Python I

- Chúng ta lưu ý rằng khi tạo ra module thì file module cần nằm cùng với file của chương trình chính. Giả sử trường hợp bạn tạo ra 100 modules thì phải tạo ra 100 files, như vậy việc quản lý các modules sẽ rất phức tạp. Để đơn giản hóa thì trong Python xuất hiện thêm khái niệm package
- Package là cách tạo ra các gói và mỗi gói sẽ chứa những module liên quan với nhau. Phương pháp này giúp việc phân chia modules trong Python một cách dễ dàng.
- Bạn có thể hiểu một package giống như folder trong máy tính vậy. Bên trong folder này có thể chứa thêm một folder khác (tức là package khác), và cũng có thể chứa thêm những file khác (tức là các modules).



# Package trong Python II



Chúng ta thấy:

- 1 Bên trong Game có thêm 3 package con là **Sound**, **Image**, **Level**
- 2 Bên trong 3 package con này chứa các module có liên quan với nhau.



- Điều đặc biệt là mỗi package bắt buộc phải có file `init.py`, đây là file được gọi đầu tiên mỗi khi bạn `import` package nó vào chương trình. Nếu bạn đã học lập trình hướng đối tượng thì file này chính là hàm khởi tạo của một lớp vậy đó.

