**মডিউল এবং প্যাকেজ**

মডিউল হল লজিকালি কোড অর্গানাইজ করার উপায়। পাইথনিক ভাষায়, মডিউল হল একটা ফাইল যেখানে পাইথন ডেফিনিশন ও স্টেটমেন্ট থাকে। .py অংশ বাদ দিলে এই ফাইলই হচ্ছে মডিউলের নাম। import স্টেটমেন্টের মাধ্যমে আমরা একটা মডিউলকে সোর্স ফাইলে আমদানি করতে পারি, সোজা কথায় যোগ করতে পারি। আর মডিউলের নামের পর ডট . চিহ্ন দিয়ে আমরা এর বিভিন্ন অ্যাট্রিবিউটকে ব্যবহার বা অ্যাক্সেস করতে পারি।

আমরা চাইলে সরাসরি কোন মডিউল থেকে কোন নির্দিষ্ট অ্যাট্রিবিউটকে আমদানি করতে পারি। এজন্য আমাদের from ... import ... স্টেটমেন্ট ব্যবহার করতে হবে।

একটা মডিউলের সবগুলো অ্যাট্রিবিউটকে একবারে আমদানি করার জন্য from ... import \* স্টেটমেন্ট ব্যবহার করতে পারি আমরা।

from fibo import \*

এবার আসি if \_\_name\_\_ == "\_\_main\_\_" এর কাহিনিতে। যখন আমরা সরাসরি একটা স্ক্রিপ্ট রান করি কেবল ও কেবলমাত্র তখনই এর \_\_name\_\_, ‍"\_\_main\_\_" এর সমান হয়। আর এই ব্লকের কোড রান হবার সুযোগ পায়। পাইথনে প্রতিটি স্ক্রিপ্টই এক হিসাবে মডিউল। যখন একটা স্ক্রিপ্টে লেখা ক্লাস বা ফাংশনকে আমরা অন্য কোথাও আমদানি করে কাজ করতে যাব তখন ব্যাকগ্রাউন্ডে পুরো স্ক্রিপ্টটাই রান করবে ও আউটপুট তৈরি করবে। কিন্তু তা তো আমাদের দরকার নাই। এইজন্যই if \_\_name\_\_ == "\_\_main\_\_" ব্যবহার করা হয়। মডিউল আমদানি করার ক্ষেত্রে ঐ মডিউলের if \_\_name\_\_ == "\_\_main\_\_" এর অধীনে থাকা কোড রান হবে না। কারণ তখন \_\_name\_\_ আর "\_\_main\_\_" সমান হয় না। এজন্য পারফেক্ট ভাবে পাইথন স্ক্রিপ্ট লেখার উপায় হল:

def my\_function():

# do something

def your\_function():

# do something

def main():

# call all functions here

my\_function()

# play with them

your\_function()

if \_\_name\_\_ == "\_\_main\_\_":

# now call main function

main()

# মডিউল সার্চ পাথ

যখন আমরা fibo মডিউলটাকে আমদানি করেছি তখন পাইথন ইন্টারপ্রিটার fibo নামের একটা বিল্ট-ইন ফাংশনের খোঁজ করে প্রথমে। না পেলে, বিভিন্ন ডিরেক্টরিতে fibo.py নামের একটা ফাইলের খোঁজ করে। এই ডিরেক্টরিগুলোর লিস্ট sys.path ভ্যারিয়েবলে থাকে। এই ডিরেক্টরি সার্চিংয়ের ব্যাপারটাও নিয়ম মেনে হয়। রানিং স্ক্রিপ্টটা যে ডিরেক্টরিতে আছে ইন্টারপ্রিটার প্রথমে সেখানে খোঁজ করে। তারপর শেল ভ্যারিয়েবল PYTHONPATH এর ডিরেক্টরি লিস্টে সার্চ করে। আর সবার শেষে ইন্সটলেশন-ডিপেন্ডেন্ট ডিফল্ট পাথে সার্চ করে। তারপরও খুঁজে না পেলে ImportError এক্সেপশান থ্রো করে।

from omuk import tomuk

**আউটপুট**

Traceback (most recent call last):

File "/home/ugcoder/Desktop/test.py", line 1, in <module>

from omuk import tomuk

ImportError: No module named 'omuk'

# স্টান্ডার্ড লাইব্রেরি

ডিফল্টভাবেই পাইথনের সাথে অনেকগুলো মডিউল থাকে। এদেরকে স্টান্ডার্ড লাইব্রেরি বলে। কেউ কেউ অবশ্য স্টান্ডার্ড মডিউলও বলে। যেমন: time একটা স্টান্ডার্ড মডিউল। বিল্ট-ইন ফাংশন dir() দিয়ে আমরা এই মডিউলের সবগুলো ভ্যালিড অ্যাট্রিবিউট দেখতে পারি।

>>> import time

>>> dir(time)

['CLOCK\_MONOTONIC', 'CLOCK\_MONOTONIC\_RAW', 'CLOCK\_PROCESS\_CPUTIME\_ID', 'CLOCK\_REALTIME', 'CLOCK\_THREAD\_CPUTIME\_ID', '\_STRUCT\_TM\_ITEMS', '\_\_doc\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'altzone', 'asctime', 'clock', 'clock\_getres', 'clock\_gettime', 'clock\_settime', 'ctime', 'daylight', 'get\_clock\_info', 'gmtime', 'localtime', 'mktime', 'monotonic', 'perf\_counter', 'process\_time', 'sleep', 'strftime', 'strptime', 'struct\_time', 'time', 'timezone', 'tzname', 'tzset']

এই ফাংশনের ভিতর কোন আর্গুমেন্ট পাস না করলে তা বর্তমান রানটাইমে আমাদের ডিফাইন করা সব ভ্যারিয়েবল, মডিউল, ফাংশনের লিস্ট রিটার্ন করবে।

>>> dir()

['\_\_builtins\_\_', '\_\_doc\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'a', 'requests', 'sys', 'time']

# প্যাকেজ

প্যাকেজ হল মডিউল স্ট্রাকচার করার একটা উপায়। মডিউল, সাব-প্যাকেজ, সাব-সাব-প্যাকেজ ইত্যাদি নিয়ে একটা প্যাকেজ হয়।

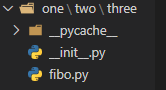
Orthat ekta main folder er moddhe onek gulo module niya pakage hote pare. Abr folder1/folder2/folder3… er modheo modules niya pakage hote pare . tahole kemne ey module pakage use kprbo? উদাহরণ দেখব।



এখানে আমরা sound প্যাকেজের বিভিন্ন সাব-প্যাকেজ ও মডিউল দেখতে পাচ্ছি। পাইথন যাতে ভিতরের ডিরেক্টরিগুলোকে সাব-প্যাকেজ মনে করে সেজন্য \_\_init\_\_.py ফাইল থাকা দরকার। \_\_init\_\_.py ফাইলে ইনিশিয়ালাইজেশন কোড থাকতে পারে। আবার খালি থাকলেও কোন সমস্যা নাই।

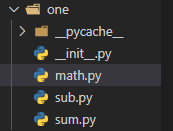
উপরের প্যাকেজ থেকে আমরা এখন echo মডিউলটা আমদানি করব। আর echo সাব-প্যাকেজের ফাংশন echofilter() ব্যবহার করব।

import sound.effects.echo



from one.two.three import fibo

Jodi ekoy pakage er under a multiple module take and ekta module a onno module import korete chi tobe



from . import Sum

যখন প্যাকেজের ভিতরের কোন ফাইলে সেই প্যাকেজের কোন মডিউল বা ফাংশনকে আমদানি করার দরকার পড়বে তখন নরমালি ইমপোর্ট করতে পারি। আবার রিলেটিভ ইমপোর্টও ব্যবহার করতে পারি। এই প্রক্রিয়ায় vocoder.py ফাইলে sound.effects প্যাকেজের echo মডিউলকে নিচের মত করে আমদানি করতে পারি।

from . import echo

বিষয়টা জটিল হয়ে গেল মনে হচ্ছে। আরেকটা নতুন প্যাকেজ স্ট্রাকচার বিবেচনা করি।

my\_package/

my\_package/\_\_init\_\_.py

my\_package/main.py

my\_package/name.py

my\_package/school.py

my\_package/college.py

এখানে main.py ফাইলে যদি আমরা school.py কে আমদানি করতে চাই অথবা school.py এর কোন ফাংশনকে আমদানি করতে চাই, তবে এভাবে করতে পারি:

from . import school

from .school import function1, function2

# পাইথন প্যাকেজ ইনডেক্স (PyPI)

[PyPI](https://pypi.python.org/pypi/) ( <https://pypi.python.org/pypi/> ) হল পাইথনের প্যাকেজ রিপোজিটরি (সংক্ষেপে রেপো বলে)। বর্তমানে এখানে প্রায় ৯১২৫৯ টি প্যাকেজ রয়েছে। এই সবগুলো প্যাকেজই কিন্তু থার্ড-পার্টি প্যাকেজ। মানে পাইথনের মূল ডেভেলপাররা নয়, বরং আমাদের মত তৃতীয় পক্ষ এগুলো ডেভেলপ করে। এগুলো পাইথনের সাথে ডিফল্টভাবে আসে না। আলাদাভাবে ইন্সটল করে ব্যবহার করতে হয়। এই থার্ড-পার্টি প্যাকেজগুলোই পাইথনের মূল শক্তি। এরাই পাইথনকে করে তুলেছে অনন্য, করে তুলেছে আনবিটেবল।

কথা হল, কিভাবে এই প্যাকেজগুলি আমরা ইন্সটল করতে পারি? এজন্য আমাদের দরকার পাইথন প্যাকেজ ইন্সটলার pip

Pip install pypi