SRI AKILANDESWARI WOMEN'S COLLEGE, WANDIWASH

PYTHON PROGRAMMING

Mr.S.BHARATHI
Assistant Professor
Department of BCA

SWAMY ABEDHANADHA EDUCATIONAL TRUST, WANDIWASH

Lecture contents

- **Section-1: Introduction to python**
- **Section-2: Features of python**
- **Section-3: Idle mode/shell mode**
- **Section-4: import packages sample codings**
- **Section-5: Conclusion**

Section-1: introduction to python programming:

Some open questions for students:

- What is python?
- What are the idle mode?
- What is the shell mode?
- What is indentation?
- How to import packages?
- How to write sample codes?
- How to implements that?

Introduction to python programming:

➢Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- ➢ web development (server-side),
- ➢ software development,
- ➢ mathematics,
- ➢ system scripting.

Why Python?

✓Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).

Python has a simple syntax similar to the English language.

✓Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

✓Python runs on an interpreter system, meaning that code can be executed as soon as it is written.

✓This means that prototyping can be very quick.

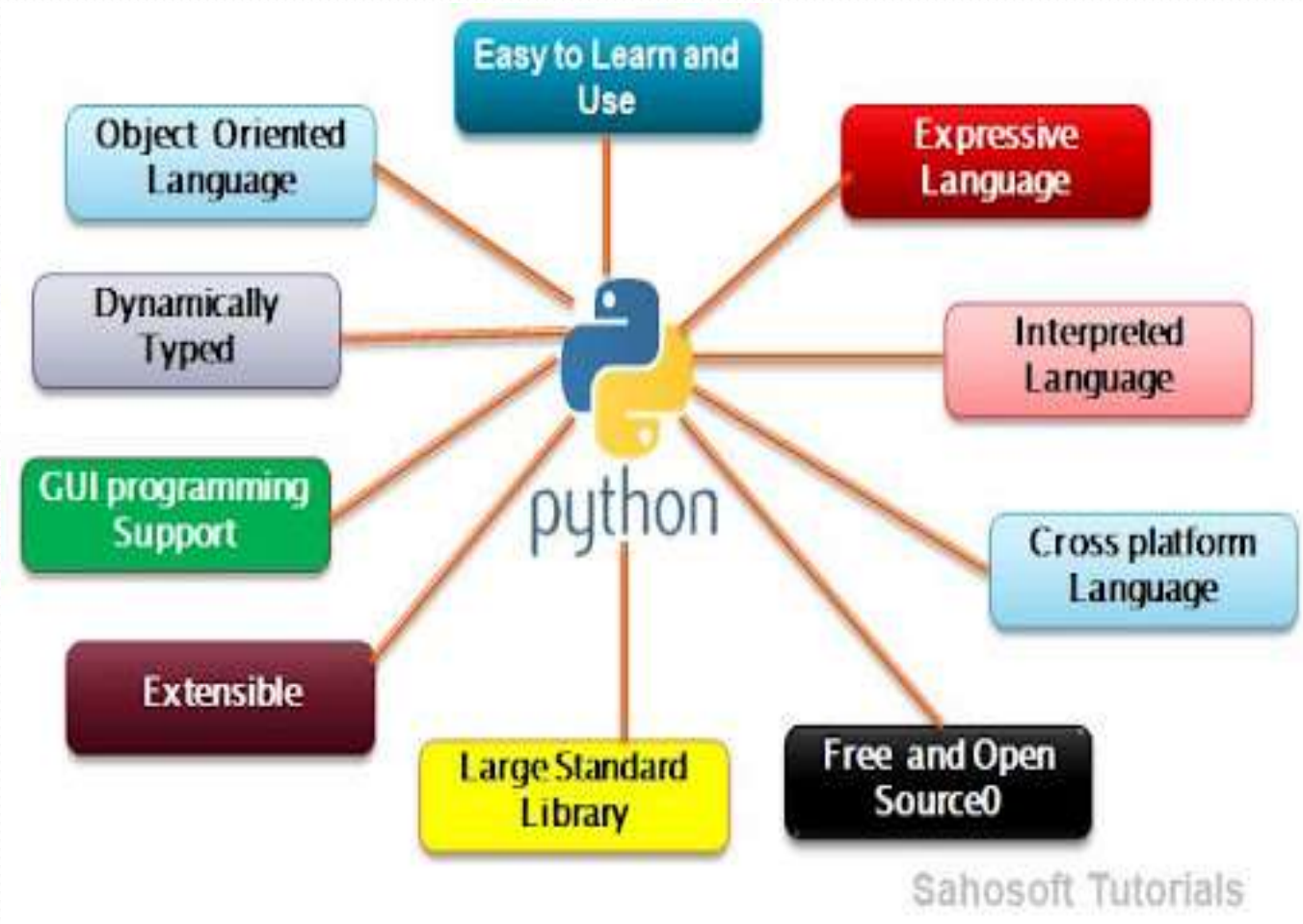
Python can be treated in a procedural way, an object-oriented way or a functional way.

Python’s Benevolent Dictator For Life

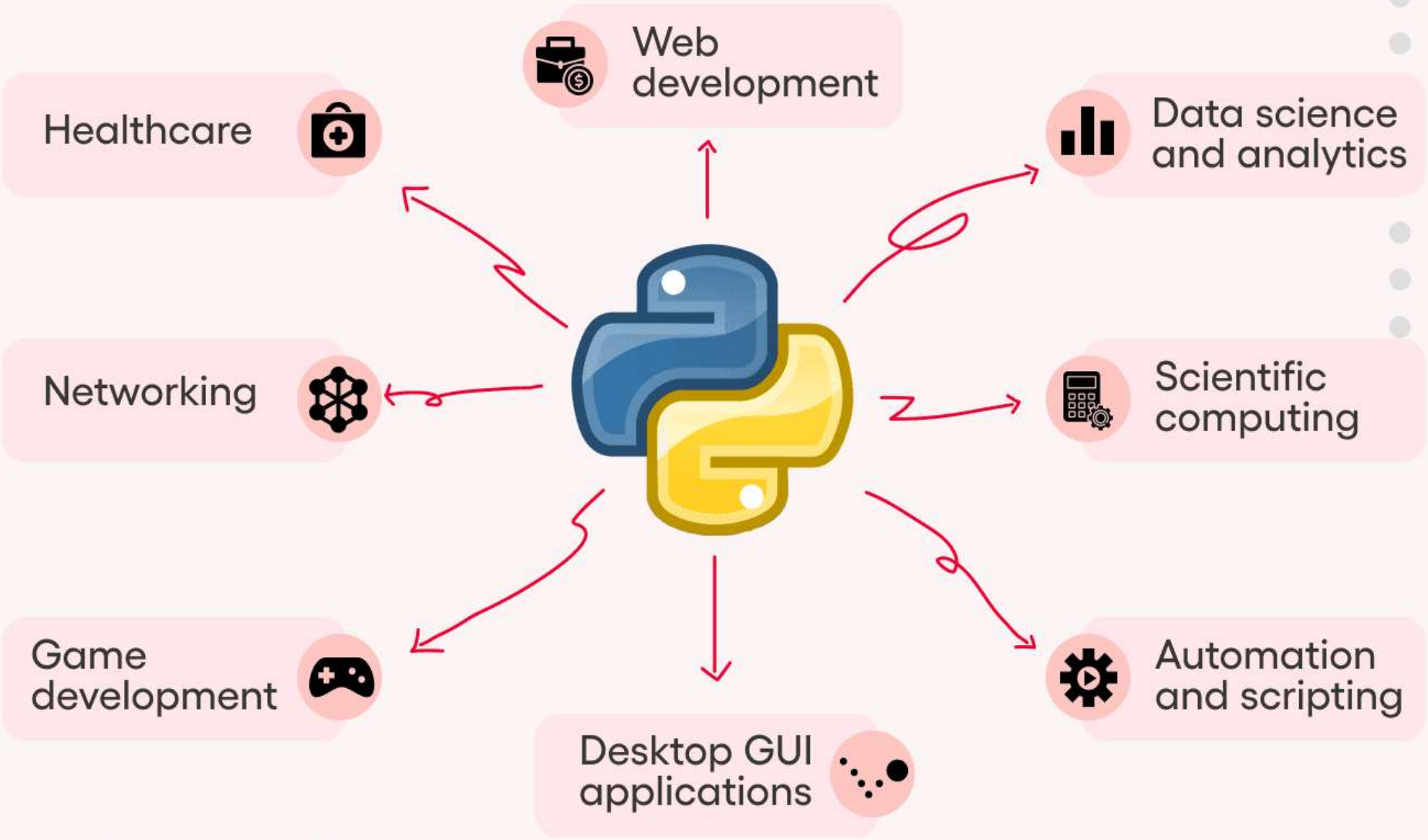
“Python is an experiment in how much freedom program-mers need. Too much freedom and nobody can read another's code; too little and expressive-ness is endangered.”

- Guido van Rossum

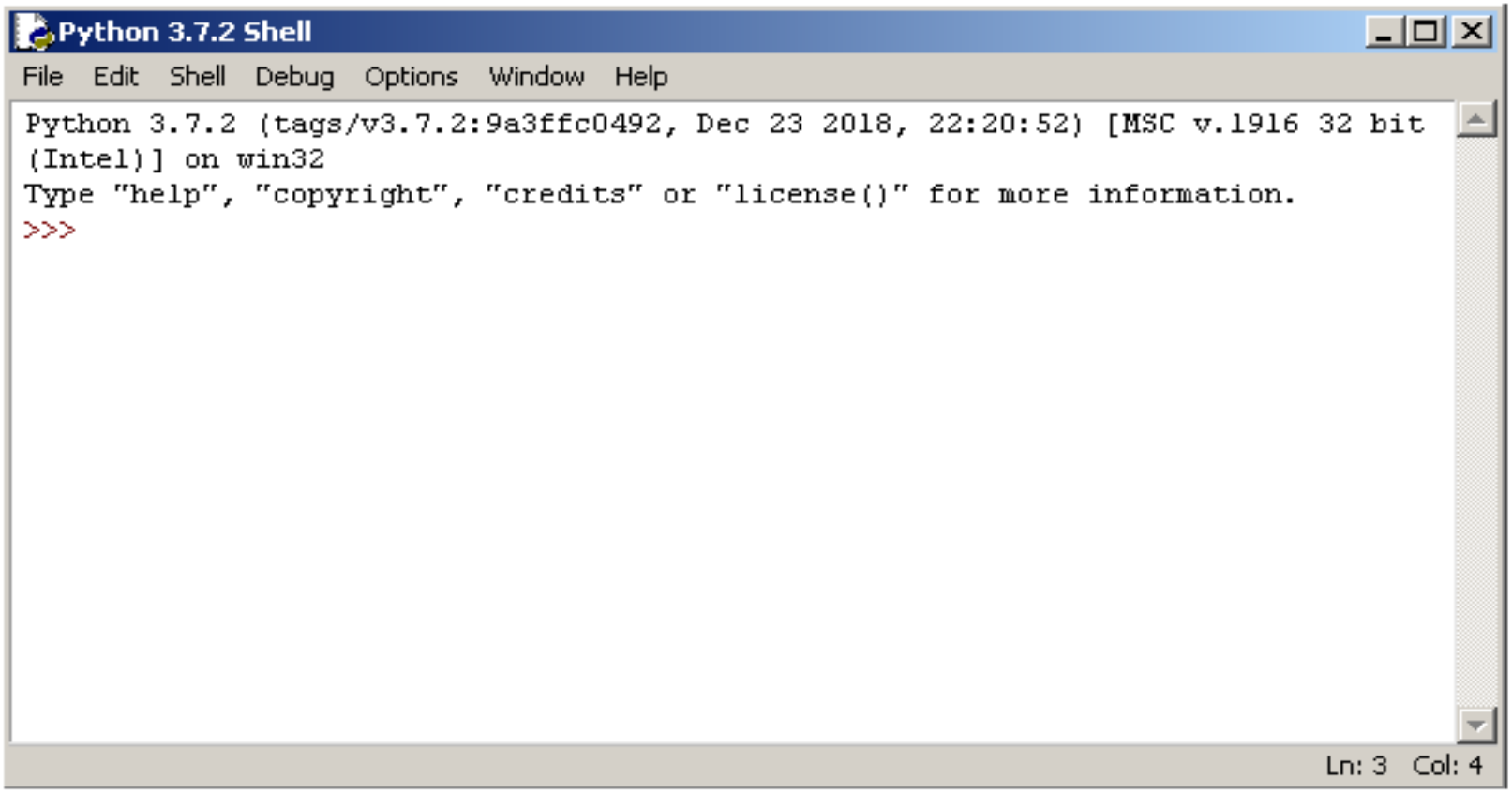
Features of python:



Applications of Python



Shell mode in IDLE:



Script mode in IDLE:

untitled

File Edit Format Run Options Window Help

```
# Taking two variables from user
a = int(input("Enter a number of integer value: "))
b = int(input("Enter second integer number: "))
# Addition of two input variables
c = a + b
# Printing addition result in the output
print("The addition of two numbers given by you is: ", c)
```

Variable Definitions in Python:

The most basic building-block of any programming language is the concept of a variable, a name and place in memory that we reserve for a value.

In Python, we use this syntax to create a variable and assign a value to this variable:

```
<var_name> = <value>
```

For example:

```
age = 56
```

You just need to call the print() function and write "Hello, World!" within parentheses:

```
print("Hello, World!")
```

You will see this message after running the program:

```
"Hello, World!"
```

Numeric Data Types in Python:

Integers, Floats, and Complex

These are the numeric types that you can work with in Python:

Integers:

Integers are numbers without decimals. You can check if a number is an integer with the type() function.

If the output is <class 'int'>, then the number is an integer.

For example:

```
>>> type(1)
<class 'int'>
 >>> type(15)
<class 'int'> >>>
 type(0)
<class 'int'>
 >>> type(-46)
 <class 'int'>
```

Float data type:

Floats are numbers with decimals. You can detect them visually by locating the decimal point. If we call type() to check the data type of these values, we will see this as the output:

<class 'float'>Here we have some examples:

```
>>> type(4.5)
<class 'float'>
 >>> type(5.8)
<class 'float'>
>>> type(2342423424.3)
<class 'float'>
>>> type(4.0) <class 'float'>
 >>> type(0.0)
<class 'float'>
>>> type(-23.5)
 <class 'float'>
```

➢Complex Datatype:

Complex numbers have a real part and an imaginary part denoted with j.
You can create complex numbers in Python with complex().
The first argument will be the real part and the second argument will be the imaginary part.

These are some examples:

```
>>> complex(4, 5)
 (4+5j)
 >>> complex(6, 8)
 (6+8j)
>>> complex(3.4, 3.4)
(3.4+3.4j)
 >>> complex(0, 0) 0j
 >>> complex(5) (5+0j)
 >>> complex(0, 4) 4j
```

Strings in Python:

Strings incredibly helpful in Python. They contain a sequence of characters and they are usually used to represent text in the code.

For example:

"Hello, World!"'Hello, World!'

String Indexing:

We can use indices to access the characters of a string in our Python program. An index is an integer that represents a specific position in the string. They are associated to the character at that position.
This is a diagram of the string "Hello":
String: H e l l o Index: 0 1 2 3 4

Data types Example:

Python 3.4.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.4.4 (v3.4.4:737efcadf5a6, Dec 20 2015, 19:28:18) [MSC v.1600 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:/Python34/python_datatypes.py ==================
The type of variable having value 100  is  <class 'int'>
The type of variable having value 10.2345  is  <class 'float'>
The type of variable having value (100+3j)  is  <class 'complex'>
>>>
```

How to import packages:

Import a Python packageIn the script we import the entire os module and make use of a piece of its functionality by calling os. getcwd() .

We import third-party Python packages into Python files in the same way as we import modules from the Python Standard Library into Python files.

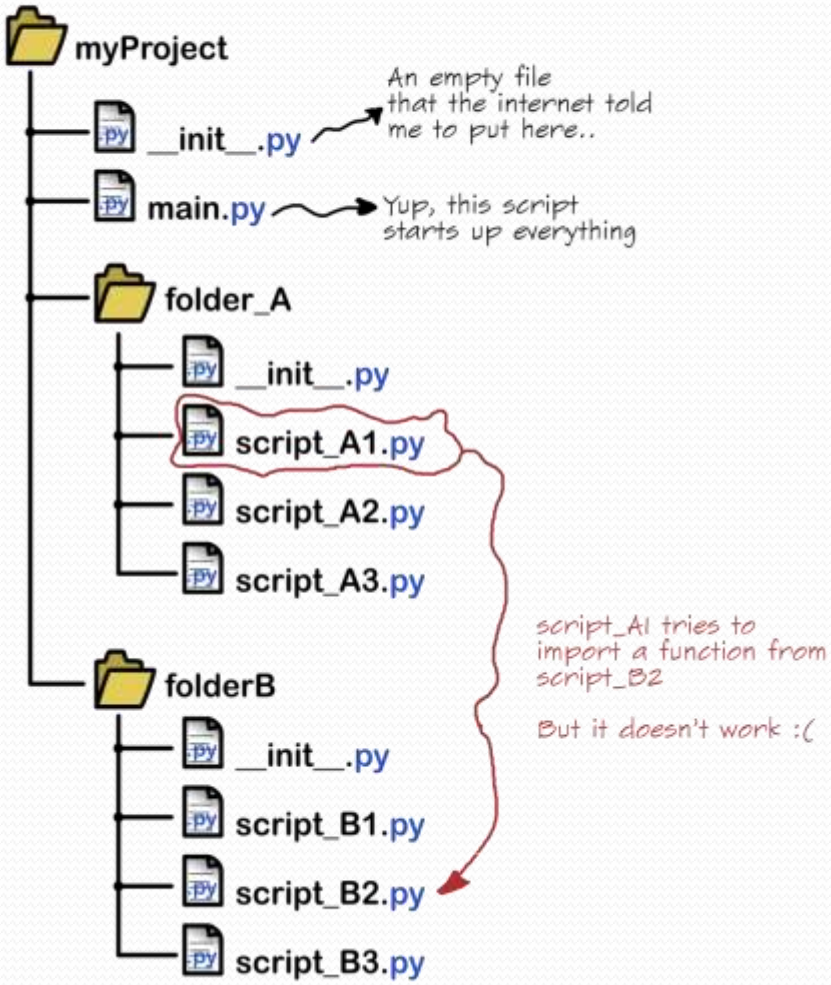
What is import from in Python?

The Python import statement lets you import a module into your code. A module is a file that contains functions and values that you can reference from your program. The import statement syntax is: import modulename

Working of modules:

Q: Importing modules from a neighbouring folder in Python

I have the following file structure for my Python project:



This is the burning question. How do I make script_A1 import a function from script_B2 ?

Similar questions have been asked before, but most answers suggest to add the module/script/package (whatever) to the PATH variable. For example:

```
sys.path.append('...')
```

But adding the module to the PATH variable just feels so wrong. I do not want to alter my system in any way. When my application closes, I want my Python environment to be clean and 'untouched'.
I'm afraid that adding uncontrolled modules to the PATH variables on my system will cause headaches later on.

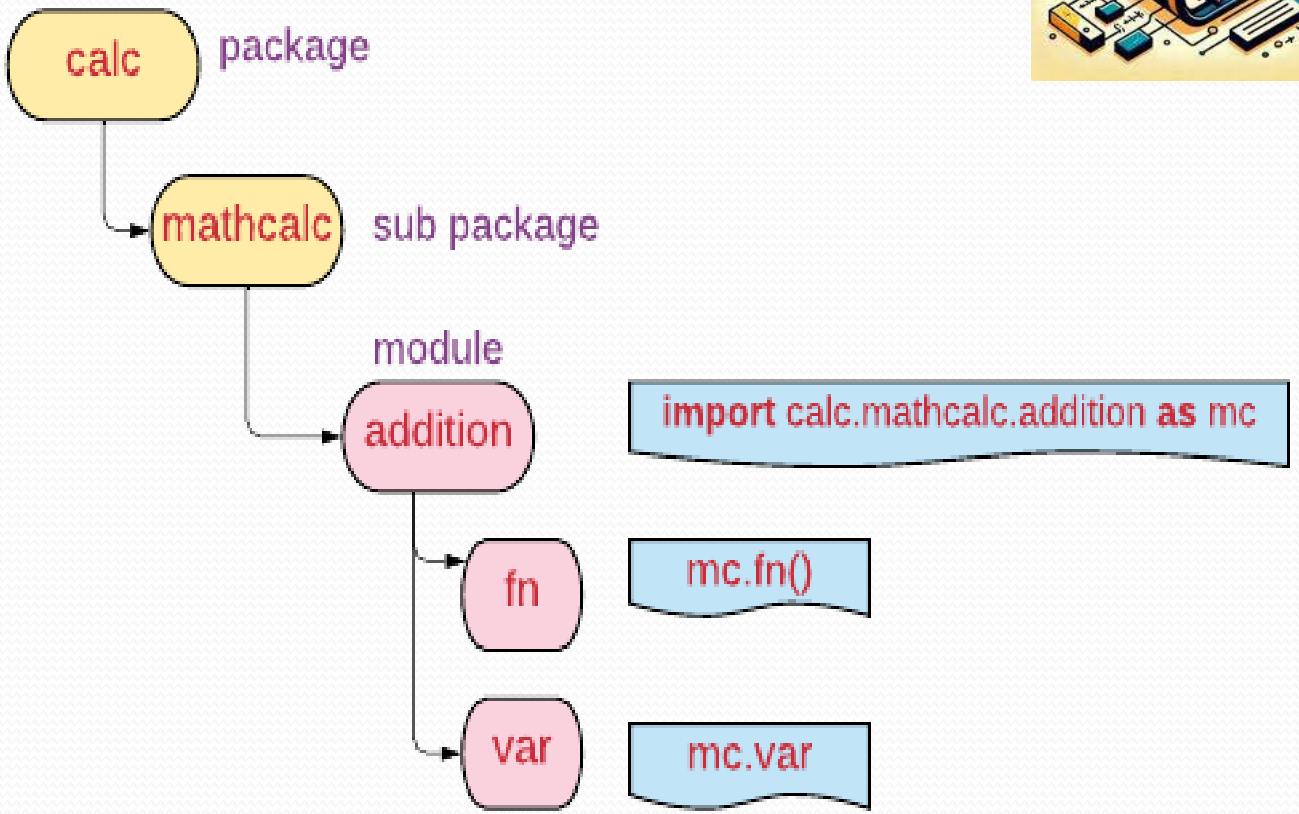
Thank you for helping me out ☺

(Please don't mark this question as duplicate. I've really done my best to make it different from the related ones).

Import packages:

```
1   import os
2   import socket
3   import warnings
4   from functools import update_wrapper
5   from threading import RLock
6
7   import werkzeug.utils
8   from werkzeug.exceptions import NotFound
9   from werkzeug.routing import BuildError
10  from werkzeug.urls import url_quote
11
12  from .globals import _app_ctx_stack
13  from .globals import _request_ctx_stack
14  from .globals import current_app
15  from .globals import request
16  from .globals import session
17  from .signals import message_flashed
18
```

How to work packages:



Python Modules

MODULE

```
def square(x):
    print(x*x)
def hello(x):
    print("Hello ",x)
```

OUTPUT

```
25
Hello Gokhan
```

CODE

```
import ourmodule
ourmodule.square(5)
ourmodule.hello("Gokhan")
```


python

Enough to Understand the Code

✓Indentation matters to code meaning

Block structure indicated by indentation

✓First assignment to a variable creates it

Variable types don't need to be declared.

Python figures out the variable types on its own.

✓Assignment is = and comparison is ==

For numbers + - * / % are as expected

Special use of + for string concatenation and % for string formatting (as in C's printf)

✓Logical operators are words (`and, or, not`) *not* symbols

✓The basic printing command is `print`

Sequence types: Tuples, Lists, and Strings

Sequence Types:

1. Tuple: ('john', 32, [CMSC])
   - A simple *immutable* ordered sequence of items
   - Items can be of mixed types, including collection types
2. Strings: "John Smith"
   Immutable
   Conceptually very much like a tuple
3. List: [1, 2, 'john', ('up', 'down')]
   - *Mutable* ordered sequence of items of mixed types

untitled

File Edit Format Run Options Window Help

```
# Taking two variables from user
a = int(input("Enter a number of integer value: "))
b = int(input("Enter second integer number: "))
# Addition of two input variables
c = a + b
# Printing addition result in the output
print("The addition of two numbers given by you is: ", c)
```

CONCLUSION:

- In conclusion, Python is a popular programming language.
- It's used for many things like web development, data science and scientific computing.
- It's easy to learn and has many resources available.
- Taking a Python course can help you learn to program or get a job that uses Python.