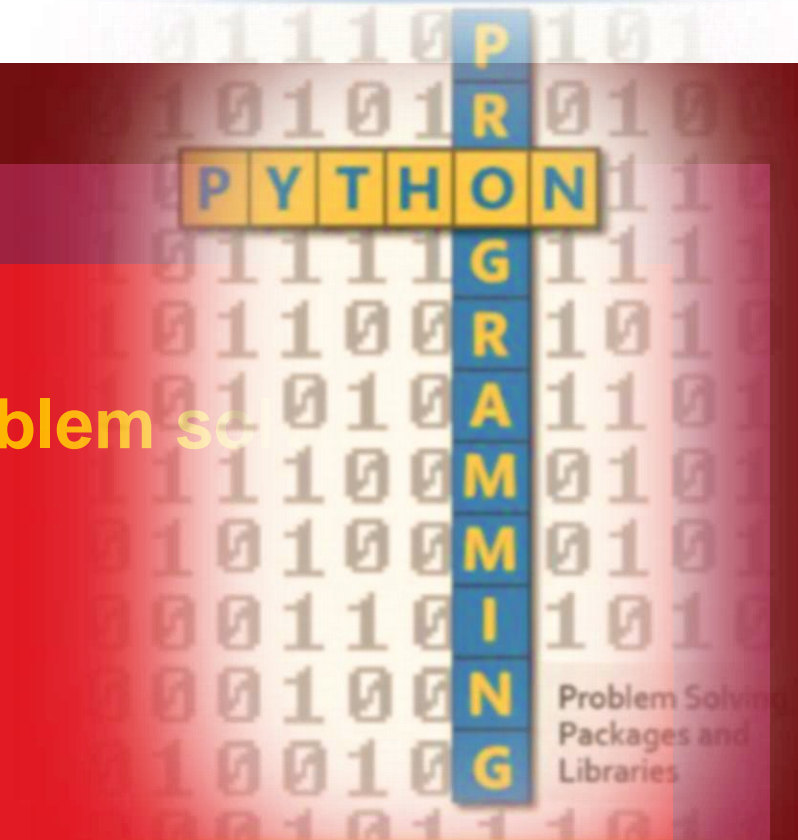


Python Programming: Problem Solving Packages and Libraries

Chapter 1



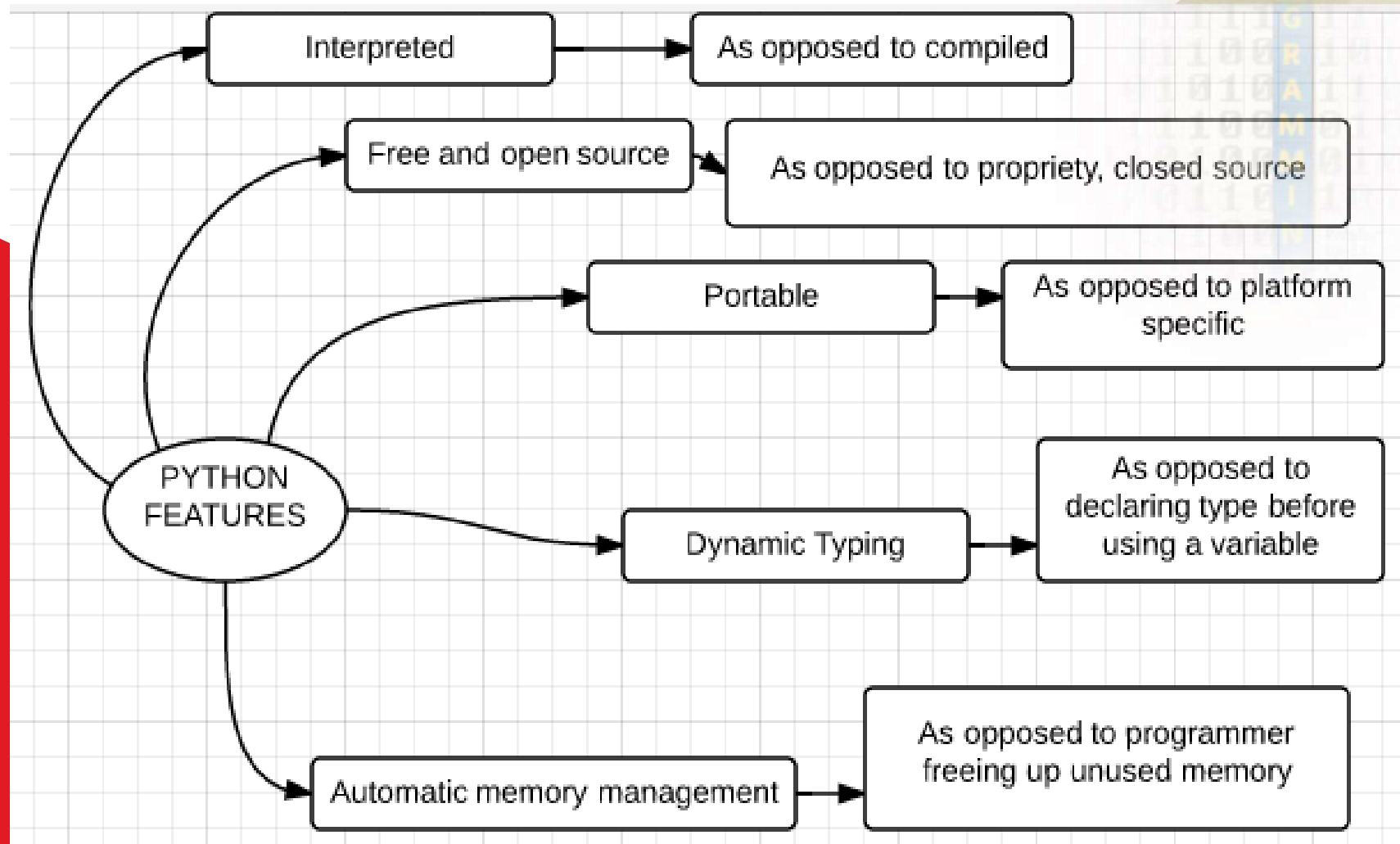
ANURAG GUPTA

AND P. BISWAS

Prerequisites:-

- This course book is largely centred around the Windows OS. But the instructions for using this book on Mac/ Linux are very similar.
- You may install python as the traditional IDLE or the more recent Anaconda distribution (Or both).
- Use of Anaconda appears to be the preferred way of learning/ using python.
- Python comes in 2 flavours ie 2.x and 3.x. We will use Python 3.x, but major differences with Python 2.x (Wherever they exist) will be pointed out.

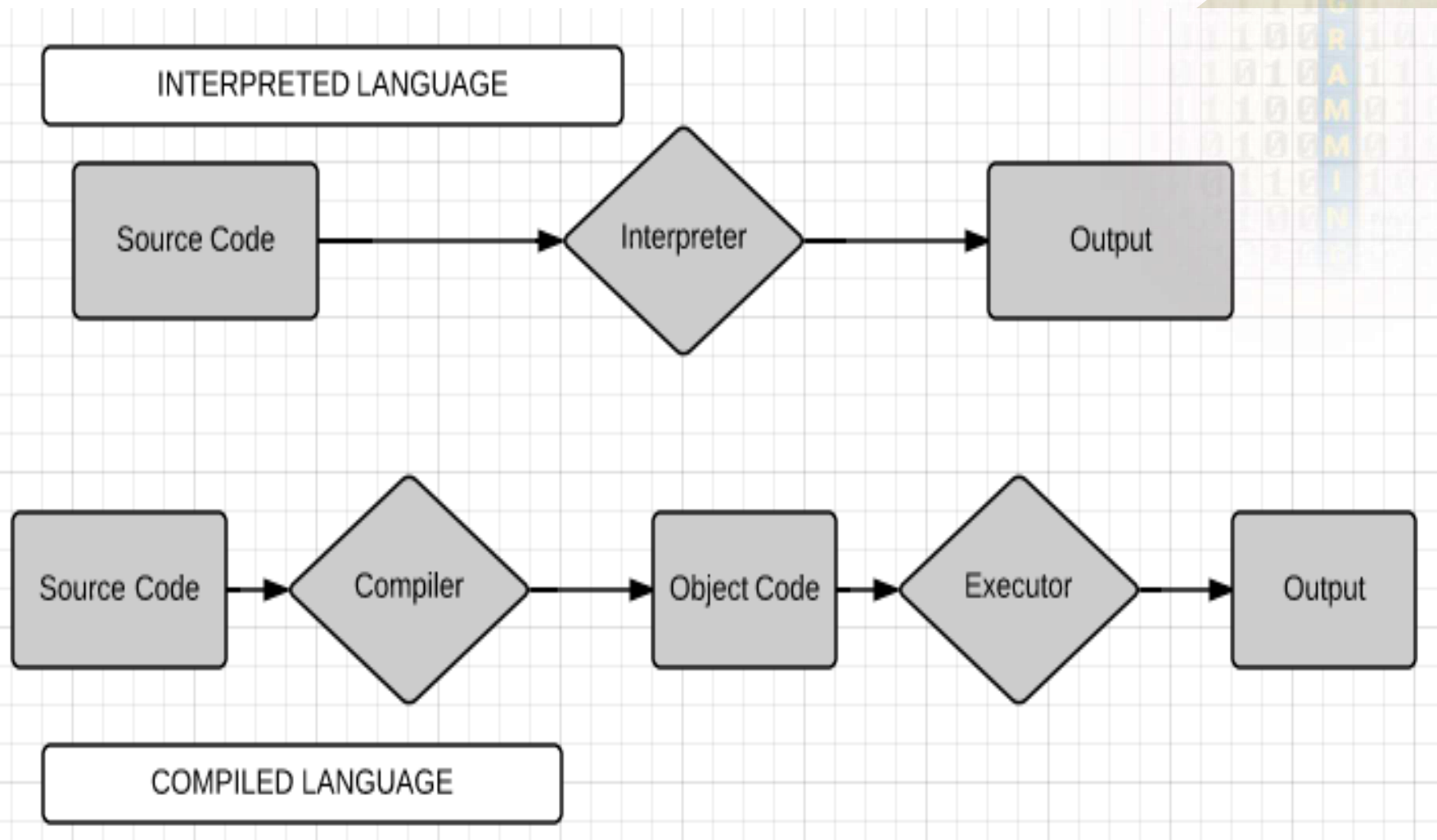
Python major features



Compiler versus interpreter

- Python is a “high level” language.
- To translate these “high level” programs into machine readable form, two types of programs are used. They are:- interpreters and compilers:-
- An interpreter reads a high-level program and executes it, one statement at a time.
- A compiler on the other hand reads the entire program (Also called “source code”) and first converts into a “machine readable” form (Also called “object code”). It then “executes” the entire translated program ie the object code in one go.

Difference between an interpreted and a compiled language



Open source versus propriety (Also free vs. paid)

- The cake analogy

- The recipe of the cake is analogous to the source code of an application.**
- If someone gives you the cake but not its recipe, then you have the cake but do not know how to make it; you cannot modify it or change its recipe. This is equivalent to a closed source system.**
- However if you are given the recipe of the cake, it is an open source system. You can now make your own cakes or even the modify the recipe to serve your own needs.**

Portable:-

- **Portability, in relation to software, indicates how easily an application can be transferred from one computer environment to another.**
- **Standard Python is written in portable ANSI C. It can compile and runs on all major operating systems like Windows, Linux, Unix and Mac.**

Dynamic typing or dynamic binding:-

- The concept of dynamic typing is explained later.
- For the present it is sufficient to know that in Python the type of a variable need not be declared before it is used.
- This is different from statically typed languages like C++ or Java, where the “type” of a variable has to be declared before it can be used

Automatic memory management:-

- The Python interpreter does automatic memory management by allocating memory when objects are created and reclaiming it when objects are no longer in use.
- It does this through “garbage collection”. This concept will be explained later.
- Java also has garbage collector
- But C does not have garbage collector. So in C, you have to explicitly free the memory yourself, or else there will be “memory leak”

Command line mode versus script mode

- **Python is an interpreted language. So there are two ways to use it:-**
- **First one can type a command on an input straight into a Python interpreter and get the output (The command line mode).**
- **The second method is to write the entire program or script in one file and then run it (The script mode).**

The traditional method to learn Python was to use the IDLE python interpreter.

However in 2001, an important variation of Python interactive shell was developed called IPython.

In 2014, a project called Jupyter notebook was started. This project provides a way of running python scripts in a web browser using a server on local machine.

Why use Anaconda?

- The anaconda package is very easy to install. You can install the entire Python interpreter along with more than 200 libraries for scientific computing with a single click.
- The user interface of IPython and Jupyter notebook is much better than of IDLE.
- Jupyter and IPython have become the virtual standards for teaching machine learning in almost all major universities all over the world.
- Jupyter also provides integration with many other programming languages (Like language R for statistics).

Y

X

Jupyter Notebook as a client-server application

- One needs to run a server(Jupyter server in this case).
- The server may be run either on the local machine or on remote machine.
- Once the server is started, it will in turn open a web browser page which will be the client of the server.
- So the Jupyter notebook client will run in a web browser like Chrome or firefox.
- The code that you enter in the web client will be sent by the web client to the server and will be executed by the server and the result displayed on the client (ie the browser).

- You can use the Python interpreter as a calculator.
- As you type in mathematical expressions, results are displayed on pressing “return”.
- The operators `+`, `-`, `*` and `/` (For addition, subtraction, multiplication and division) work just like in many popular programming languages.
- You can use parenthesis ie `()` for grouping.
- Hash ie `#` is used for single line comments and totally ignored by the Python interpreter.

print statement in Python 2.x and print() function in 3.x

- **You can give a print command with text to IDLE and the Python interpreter will output the text.**
- **In Python 2.x print was a statement**
- **while in Python 3.x print is a function.**
- **Functions are dealt with later but for the present it is sufficient to know that in Python, a function call is followed by parenthesis ie ().**
- **In Python 3.x you have to wrap the object that you want to print in parenthesis:-**

Explicit vs Implicit line continuation

- In Python, end of a statement is marked by a newline character. But you can make a statement extend over multiple lines with the line continuation character (`\`). This is explicit line continuation.
- Line continuation in Python, is implicit (ie implied) inside brackets `[]`, parentheses `()` and braces `{ }`.

Python keywords

and	del	from	not	while
as	elif	global	or	with
assert	else	if	pass	yield
break	except	import	print	
class	exec	in	raise	
continue	finally	is	return	
def	for	lambda	try	

“Seeing” the keyword list on Jupyter. (In the book it is shown on interpreter

You can get the entire keyword list on Jupyter by typing in the following 2 commands:-

```
import keyword  
keyword.kwlist
```

```
In [1]: 1 import keyword  
        2 keyword.kwlist
```

```
Out[1]: ['False',  
         'None',  
         'True',  
         'and',  
         'as',  
         'assert',  
         'break',  
         'class',
```

Rules for writing identifiers in Python

- **An identifier can be a combination of lowercase letters ie (a to z), uppercase letters ie (A to Z) digits ie (0 to 9) or an underscore (_).**
- **An identifier must not start with a digit. 1stName is invalid, but Name1 is ok.**
- **Special symbols like @, \$,!, #, % etc. cannot be used in identifier.**
- **You cannot use Python keywords as identifiers. Python is a case-sensitive language. Therefore myVariable and Myvariable are not the same.**

Some good identifier naming conventions:

- One should name identifiers that make sense. While, `n = 10` is valid. Writing `number = 10` would make for more readable code and easier to figure out on a later date.
- 2. It is good naming convention to separate multiple words with underscores. For example `my_age` is a valid identifier.
- 3. You can also use camel-case style of writing, i.e., capitalize every first letter of the word except the initial word. Thus `my_age` in camel case would be `myAge`. Come to think of it, `myAge` does look like a camel with the capital A representing the hump of the camel

A “variable” as a “box” vs a variable as a “tag” analogy

A “variable” is like the “name of a box” which contains something.

A variable can also be seen as a tag. The “type” of the variable is the kind of data it stores.

In strongly typed languages like C++, each variable also has a type which has to be declared before the variable name can be used.

This is in contrast to variable name in Python. In python, a variable name can be compared to a “tag”.

Assignment

- **Note that the assignment of a data (like string, integer, float etc) to a variable is different in C++ and in Python.**
- **In C++, the assignment of a data to a variable first creates a place in memory, which you can think of as a box. Once such a “box” is created, then data may be placed in the box. However assignment of a value to a variable in Python is different**

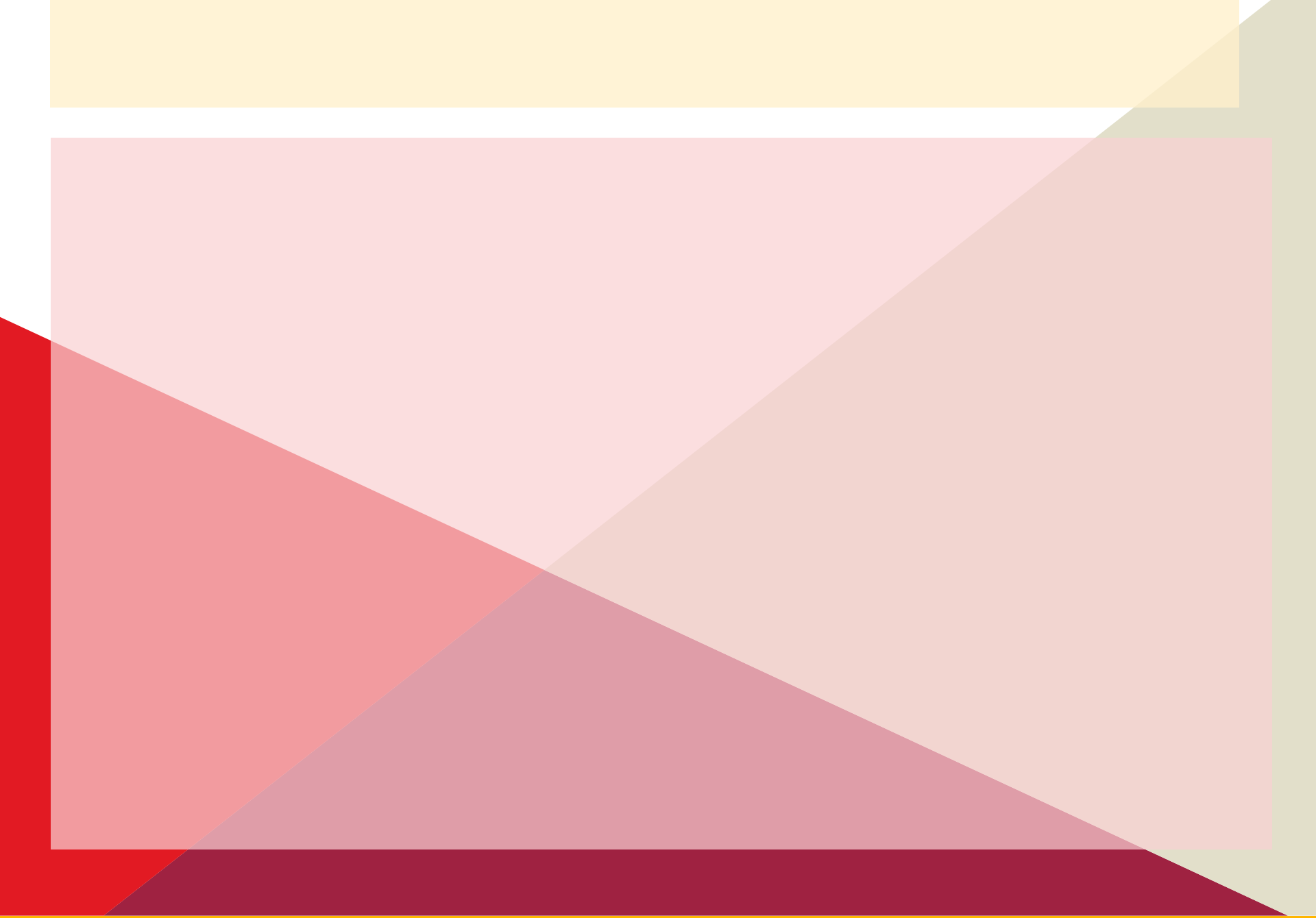


Variable vs. identifier:-

- The terms “variables” and “identifiers”, appear synonymous.
- However an identifier is “more than” a variable, meaning that all variables are identifiers but all identifiers may not be variables.
- So you can say that variables are a sub-set of identifiers.
- An “identifier” can be used not only for a “variable” but even other entities like “labels”, “functions”, “modules” etc.
- The name of the variable is the identifier for that variable.
- Variables must be “assigned a value” before they can be used, or an error will occur.

Why style guide and naming conventions are important

- One of Guido's key insights is that code is read much more often than it is written.
- So you must write your code in a manner that others can understand.
- For this you must follow the Python “style guide”.
- Why? Because style guide is about “consistency”.
- The 2 major most popular style guides for Python are PEP 8 and PEP 257.
- PEP 8 – Style Guide for Python Code.
- PEP 257 – Docstring Conventions. (Docstrings are discussed later)






Because learning changes everything.®

Thank You!

For any queries or feedback contact us at:

 support.india@mheducation.com

 1800-103-5875

 www.mheducation.co.in

in

