

# UE18CS101: INTRODUCTION TO COMPUTING USING



Department of Computer Science and  
Engineering  
PES UNIVERSITY

# Lecture 4

- First program in Python
- Program Structure
- Running a program

# The Python Programming Language



The Python Programming Language was created by **Guido van Rossum**. It was first released in the early 1990s.

Its name comes from a 1970s British comedy sketch show called *Monty Python's Flying Circus* ([The Argument Clinic](#)).

Companies and organizations that use Python include YouTube, Google, Yahoo and NASA.

# Python Features

- **Simple Syntax**  
Python programs are clear and easy to read
- **Interpreted Language**  
Python instructions can be executed interactively
- **Powerful Programming Features**  
Can accomplish significant computation with few instructions
- **Numerous Python Modules Provide Additional Capabilities**  
Capabilities that can be incorporated into a Python program

# Program Structure

Distinguishes upper case and lower case

Case sensitive

- consists of # of statements
  - # a statement has to be on a line
  - # may use some symbol with its corresponding closing symbol
  - # to write a statement on multiple lines
- # source code - program in a high level language
- # format - way of presentation
- # not a free format source code
- # all statements in the beginning should start from the first column

# Program Structure

```
print("one")  
# Print("two") # upper case and lower case are distinguished  
print("three")print  
# fn name => expr => stmt; no call !!  
("no output")  
# string within parentheses is an expr ;  
no action  
3 + 4  
# forgot to ask this to be displayed!!  
"four"
```

# Program Structure

# continue statement on multiple lines

# 1. use constructs which has beginning and ending markers - like a pair of parentheses

```
print("five")
```

# 2. enter \ (backslash) before pressing <Enter> key - this is called escaping.

```
print \
```

```
(
```

```
"six"
```

```
)
```

# multiple statements on a single line

```
print("seven"); print("eight")
```

# 3 + 4 is an expression

# "four" is also an expression

# an expression has value

# function name is also an expression

# any expr is a statement

# The IDLE Python Development Environment

---

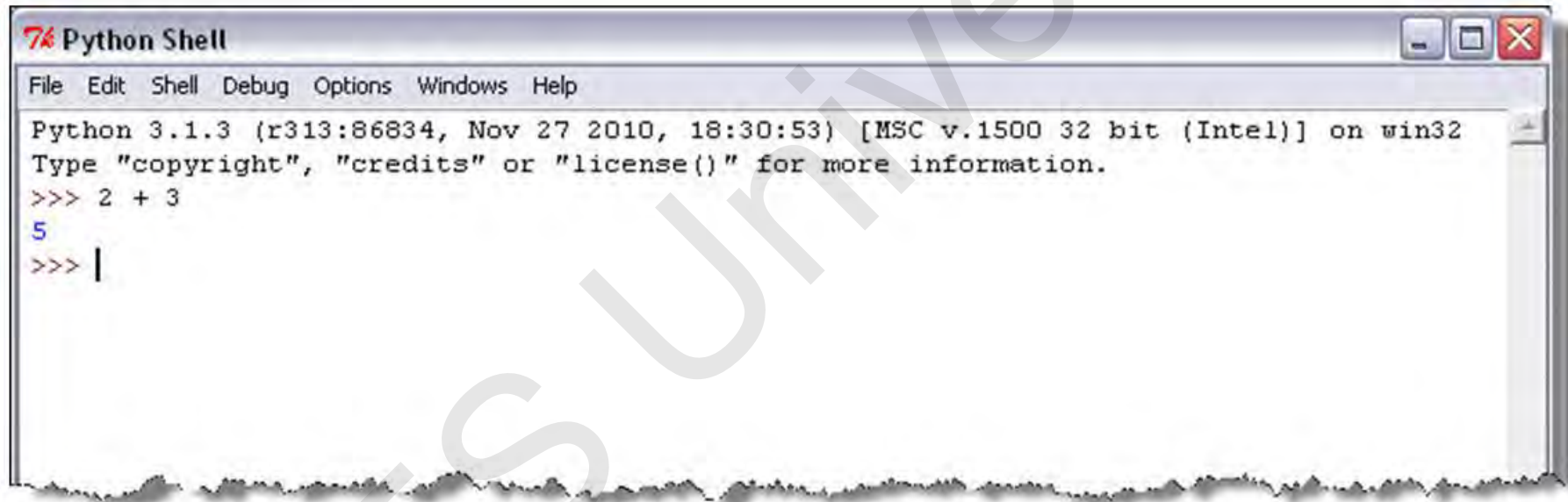
**IDLE** is an **integrated development environment (IDE)**. An **IDE** is a bundled set of software tools for program development. This typically includes,

- **an editor**  
for creating and modifying programs
- **a translator**  
for executing programs, and
- **a program debugger**  
for taking control of the execution of a program  
to aid in finding program errors



## The Python Shell

Python can be executed interactively in the **Python shell**. In this mode, **executing Python is similar to using a calculator**.



The **>>>** symbol is the shell prompt. Here, typing `2 + 3` at prompt outputs the result `5`, again displaying the prompt in wait of another instruction.

## The Python Standard Library

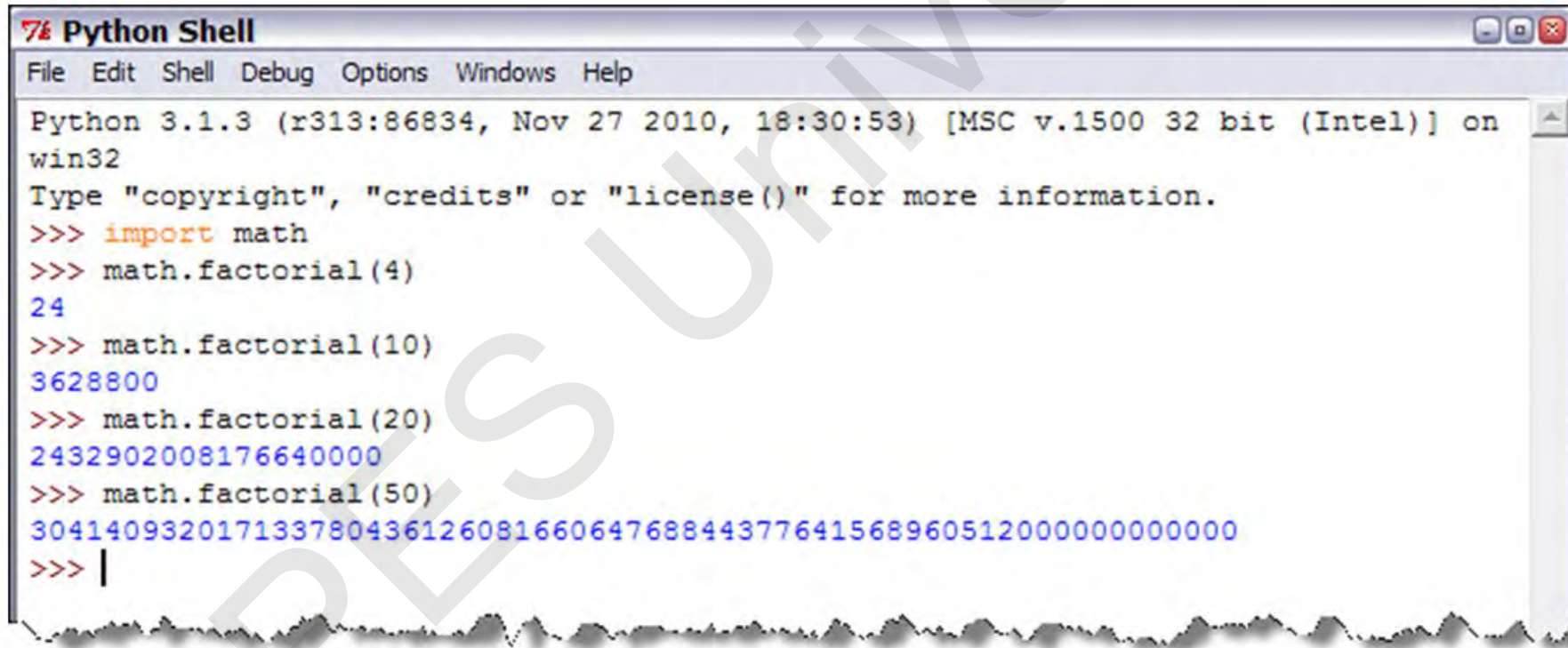
The **Python Standard Library** is a collection of **built-in modules**, *each providing specific functionality* beyond what is included in the “core” part of Python.

For example, the **math module** provides additional mathematical functions. The random module provides the ability to generate random numbers, useful in programming, as we shall see.

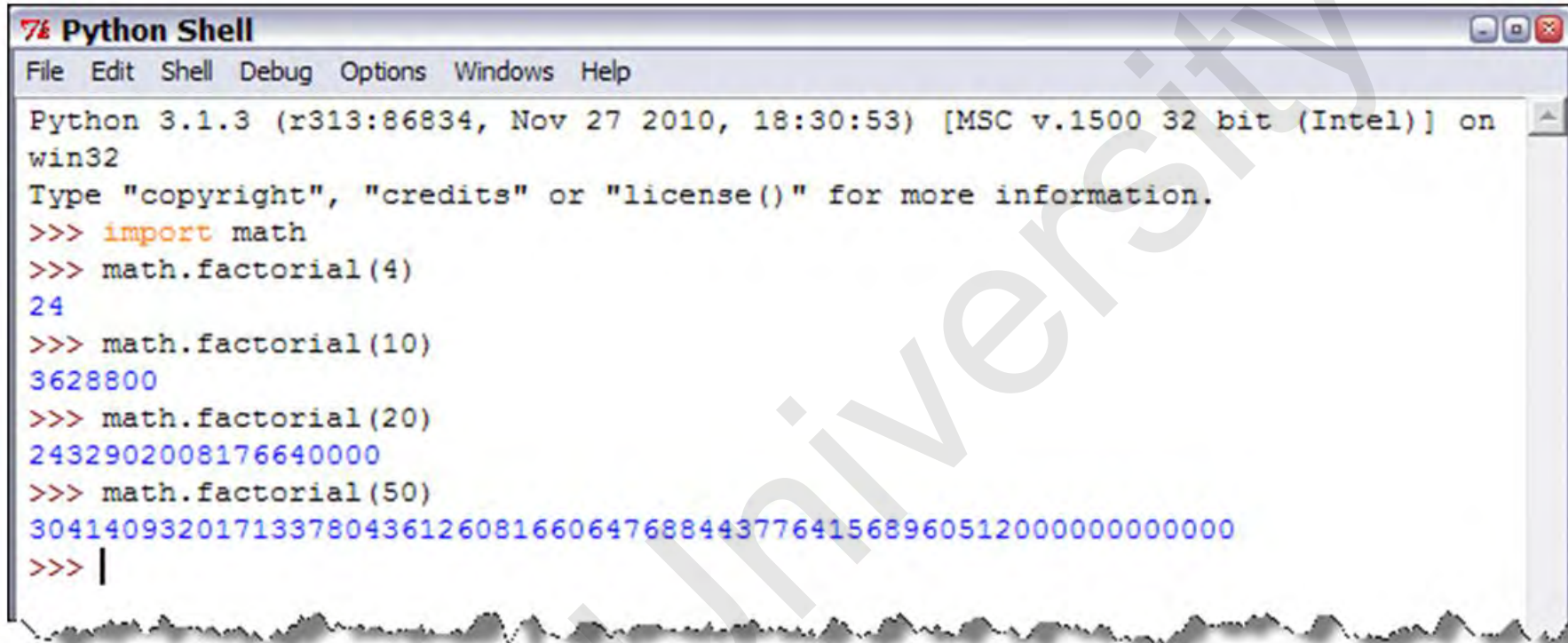
Other Python modules are described in the [Python 3 Programmers' Reference](#).

## Importing a Library Module

In order to utilize the capabilities of modules in a specific program, an **import statement** is used as shown.



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.1.3 (r313:86834, Nov 27 2010, 18:30:53) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import math
>>> math.factorial(4)
24
>>> math.factorial(10)
3628800
>>> math.factorial(20)
2432902008176640000
>>> math.factorial(50)
304140932017133780436126081660647688443776415689605120000000000000
>>> |
```



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.1.3 (r313:86834, Nov 27 2010, 18:30:53) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import math
>>> math.factorial(4)
24
>>> math.factorial(10)
3628800
>>> math.factorial(20)
2432902008176640000
>>> math.factorial(50)
30414093201713378043612608166064768844377641568960512000000000000
>>> |
```

Because the factorial function is from the math module, the function is called with the name of the module prepended:

e.g., `math.factorial(20)`

## A Bit of Python

We introduce a bit of Python, just enough to begin writing some simple programs.

Since all computer programs,

- input data
- process the data
- output results

we look at the notion of a **variable**, how to perform some simple arithmetic calculations, and how to do simple input and output.

## Variables

One of the most fundamental concepts in programming is that of a **variable**.

A variable is “**a name that is assigned to a value**,” as shown below,

$n = 5$                       variable  $n$  is assigned the value 5

Thus, whenever variable  $n$  appears in a calculation, it is the current value of  $n$  that is used, as in the following,

$n + 20$                        $(5 + 20)$

If variable  $n$  is assigned a new value, then the same expression will produce a different result,

$n = 10$   
 $n + 20$                        $(10 + 20)$

## Some Basic Arithmetic Operators

The **common arithmetic operators** in Python are,

<b>+</b> (addition)	<b>*</b> (multiplication)	<b>**</b> (exponentiation)
<b>-</b> (subtraction)	<b>/</b> (division)	

Addition, subtraction, and division use standard mathematical notation,

$10 + 20$        $25 - 15$        $20 / 10$   
(Also, **//** for truncated division, discussed later)

For multiplication and exponentiation, the asterisk (\*) is used,

$5 * 10$  (5 times 10)       $2 ** 4$  (2 to the 4th power)

Multiplication is never denoted by the use of parentheses,

$10 * (20 + 5)$  **CORRECT**       $10(20 + 5)$  **INCORRECT**

Note that parentheses may be used to denote subexpressions.

## Basic Input

**The programs that we will write request and get information from the user.**  
In Python, the **input function** is used for this purpose,

```
name = input('What is your name?: ')
```

**Characters within quotes are called strings.** This particular use of a string, for requesting input from the user, is called a **prompt**.

The input function displays the string on the screen to prompt the user for input,

```
What is your name?: Charles
```

The underline is used here to indicate the user's input.



## Basic Output

The **print function** is used to display information on the screen in Python. This may be used to display a message (string),

```
>>> print('Welcome to My First Program!')  
Welcome to My First Program!
```

or used to output the value of a variable,

```
>>> n = 10  
>>> print(n)  
10
```

Can also display a combination of strings and variables,

```
>>> name = input('What is your name?: ')\nWhat is your name?: Charles
```

```
>>> print('Hello', name)\nHello Charles
```

**Note that a comma is used to separate the individual items being printed, which causes a space to appear between each when displayed.** Thus, the output of the print function in this case is

```
Hello Charles
```

and not

```
HelloCharles
```

We will soon learn more about variables, operators, and input/output in Python.

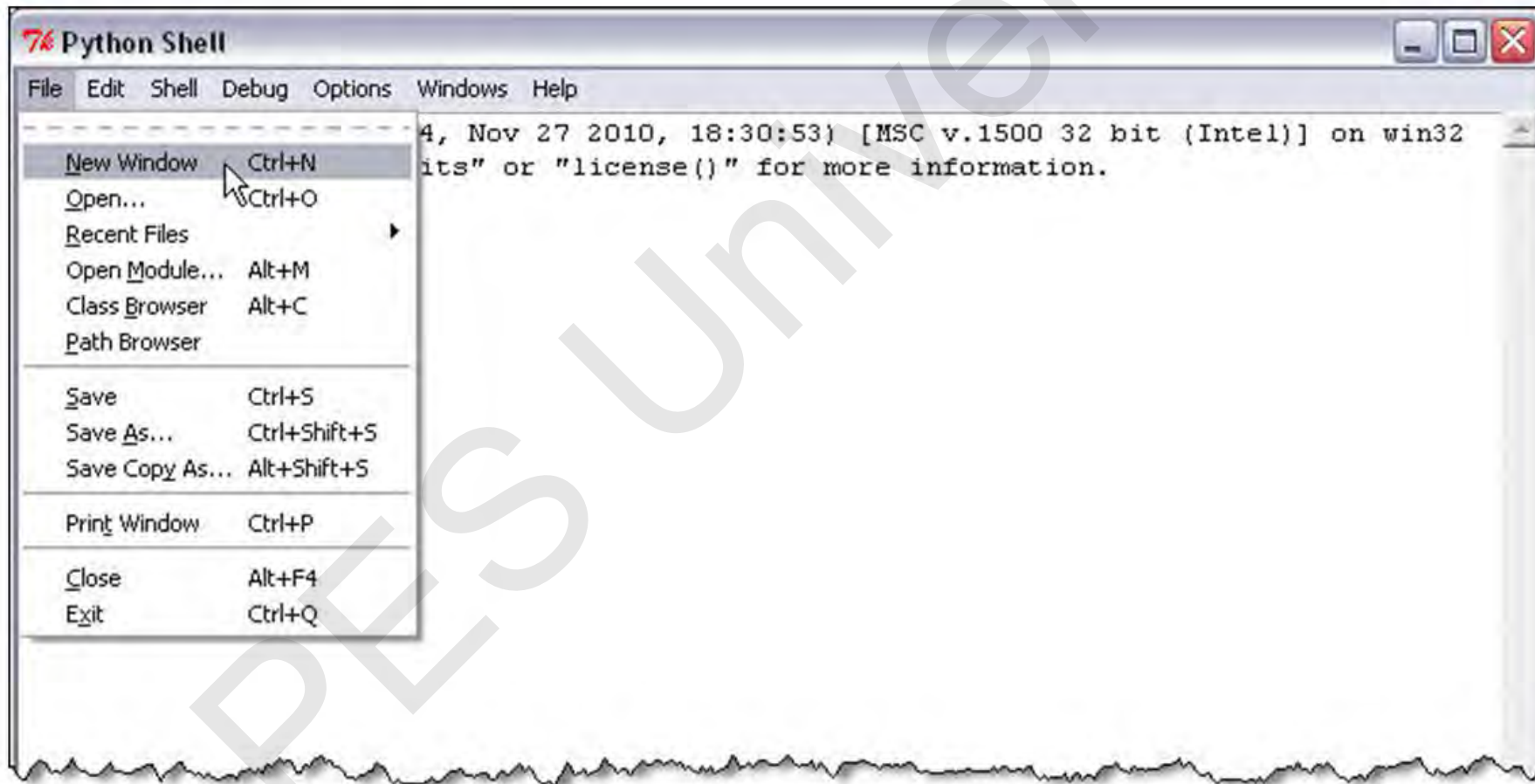
## Using IDLE

In order to become familiar with writing your own Python programs using IDLE, we will create a simple program that asks the user for their name and responds with a greeting. This program utilizes the following concepts:

- **Creating and executing Python programs**
- **Input and print functions**

## Creating a New Python Program


To create a Python program file, select **New Window** from the **File** menu in the Python shell,



A new, untitled window will appear,



Now can **begin entering lines** of a program without them being immediately executed, as in the Python shell.

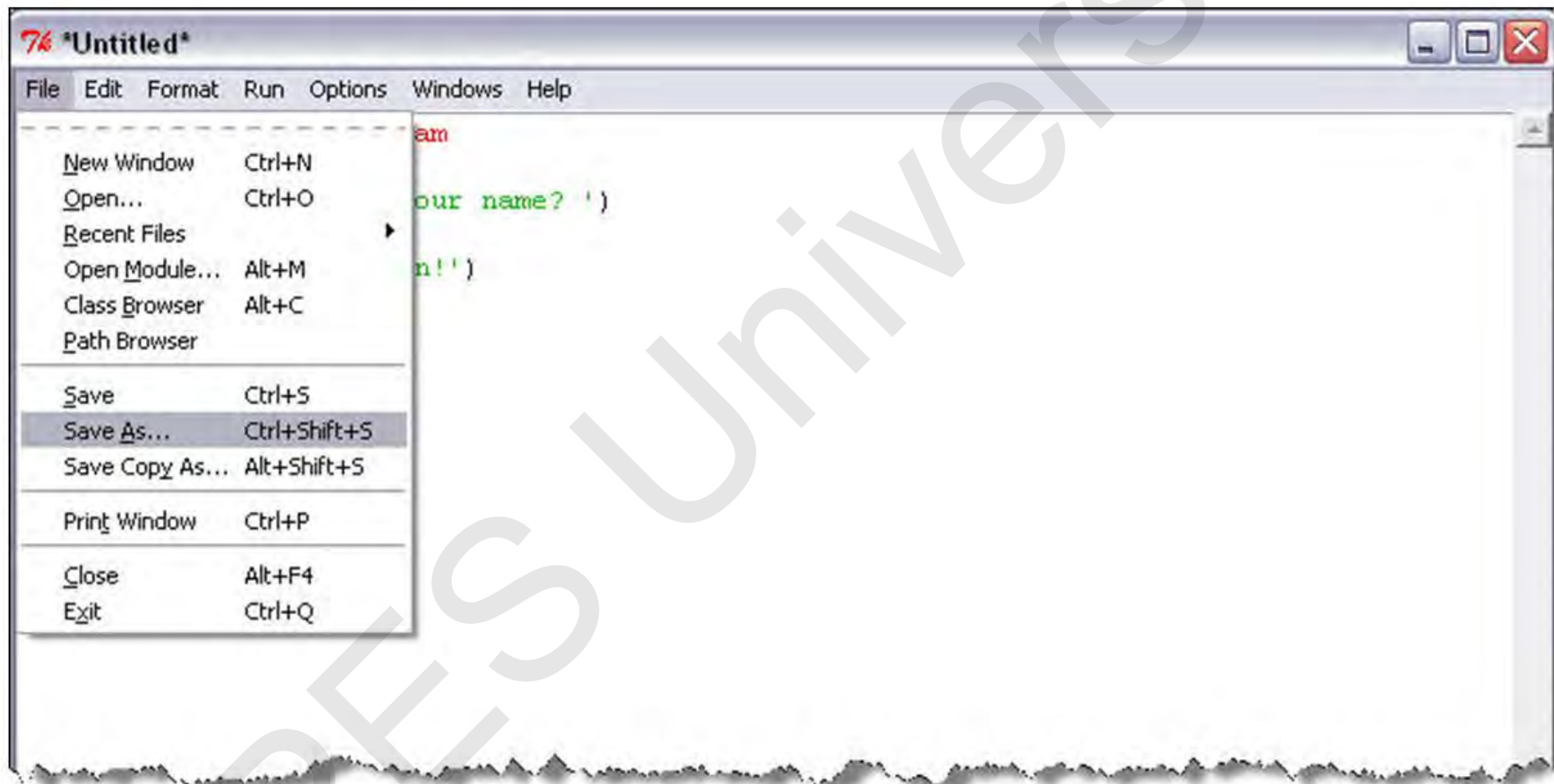


```
7k *Untitled*
File Edit Format Run Options Windows Help
# My First Python Program

name = input('What is your name? ')
print('Hello', name)
print('Welcome to Python!')
```

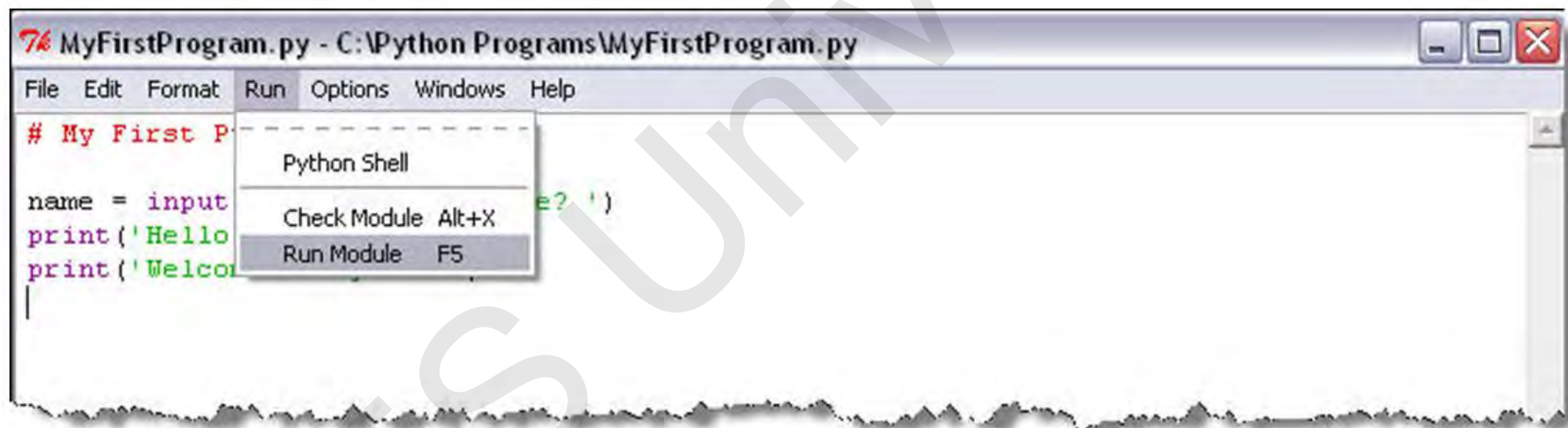
**Note that parts of the program lines are displayed in a certain color.** Since **print** and **input** are predefined function names in Python, they are colored purple. The **strings** in the program are colored green. The statement in red is a **comment statement**. **Comment statements are for those reading the program, and are ignored when the program is executed.**

When finished, **save the program file** by selecting **Save As** under the **File menu**, and save in the appropriate folder with the name **MyFirstProgram.py**.



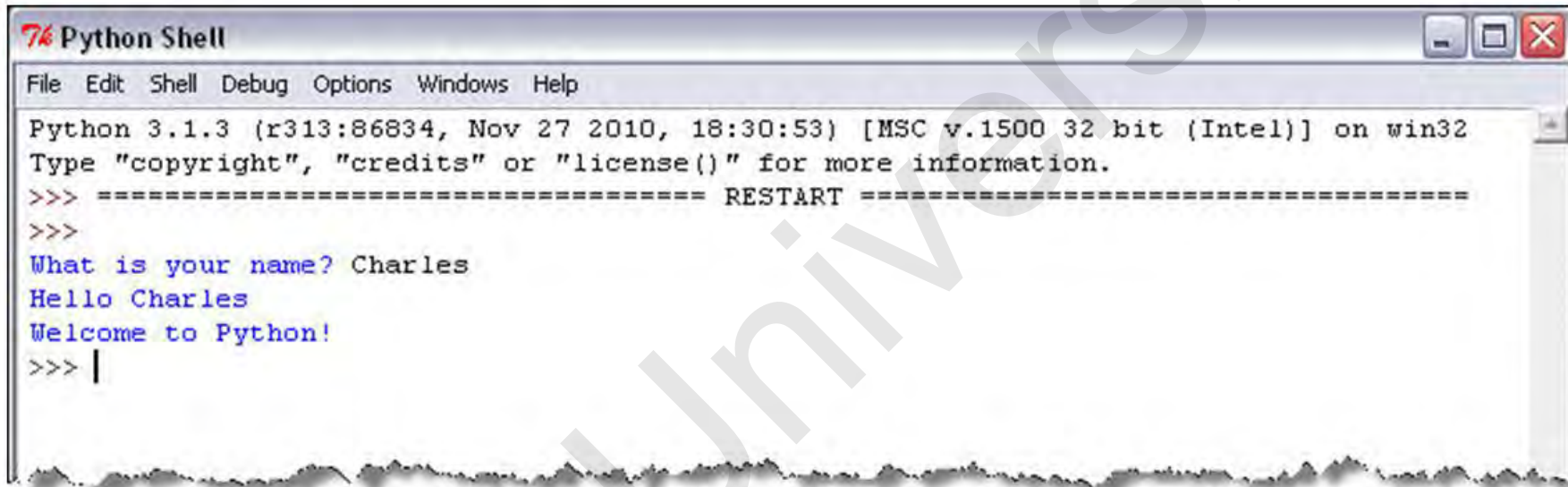
## Executing a Python Program

To run a Python program, select **Run Module** from the Run menu (or simply hit **function key F5**).



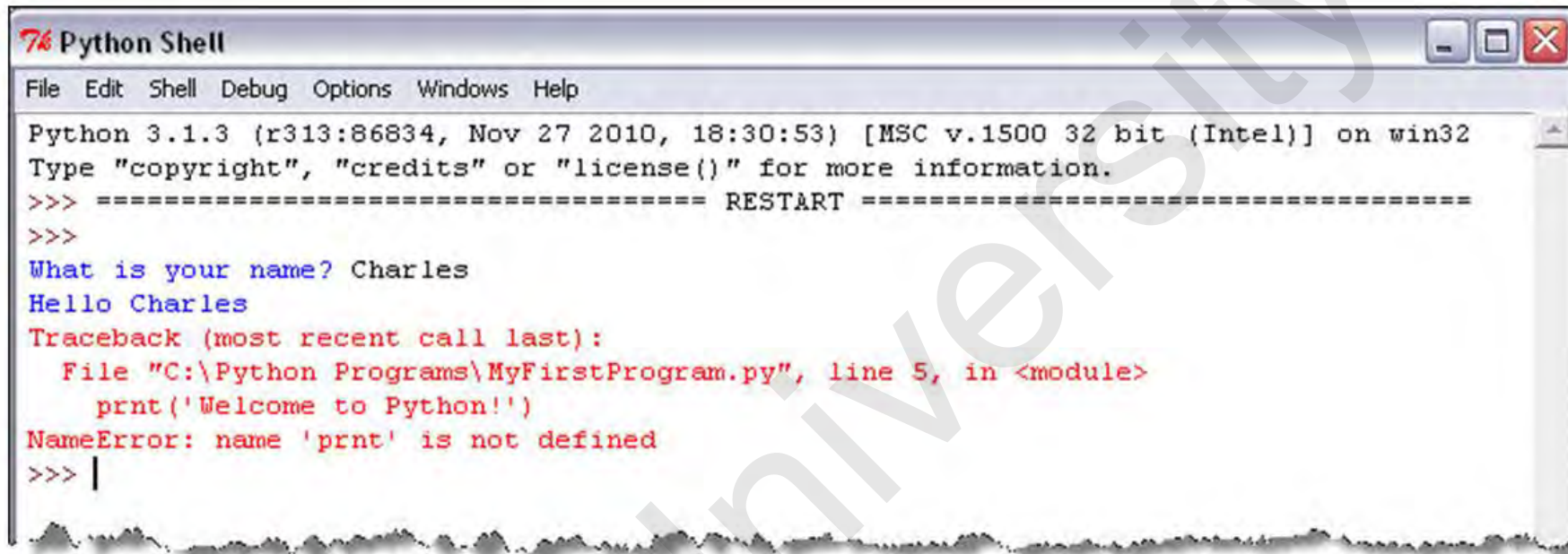


If you have entered the program code correctly, the program should execute as shown



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.1.3 (r313:86834, Nov 27 2010, 18:30:53) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
What is your name? Charles
Hello Charles
Welcome to Python!
>>> |
```

If, however, you have mistyped part of the program resulting in a **syntax error** (such as mistyping `print`), you will get an error message.



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.1.3 (r313:86834, Nov 27 2010, 18:30:53) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
What is your name? Charles
Hello Charles
Traceback (most recent call last):
  File "C:\Python Programs\MyFirstProgram.py", line 5, in <module>
    print('Welcome to Python!')
NameError: name 'print' is not defined
>>> |
```

In such instances, you need to go back to the program window and make the needed corrections, the re-save and re-execute the program. **You may need to go through this process a number of times until all the syntax errors have been corrected.**