Programming Lab

Autumn Semester

Course code: PC503

I can toast!
Let me rephrase that
I can do ANYTHING you tell me to...
... provided you speak my language

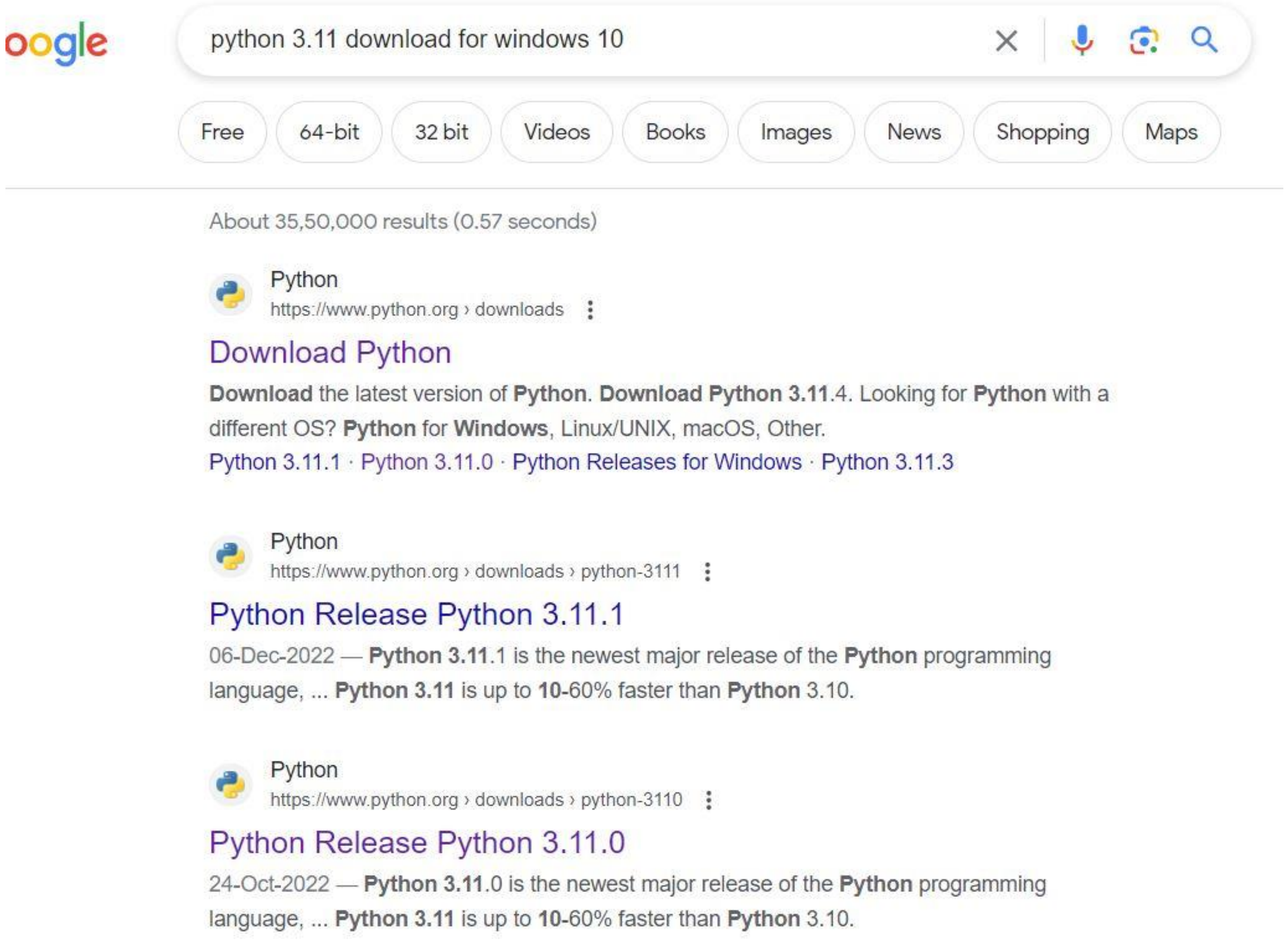
Dr. Rahul Mishra
Assistant Professor
DA-IICT, Gandhinagar

Images are collected from the web and may be subject to copyright

Lecture 1

Installation and Basics

Installation



python 3.11 download for windows 10

About 35,50,000 results (0.57 seconds)

Python
https://www.python.org › downloads

Download Python

Download the latest version of **Python**. **Download Python 3.11**.4. Looking for **Python** with a different OS? **Python** for **Windows**, Linux/UNIX, macOS, Other.

Python 3.11.1 · Python 3.11.0 · Python Releases for Windows · Python 3.11.3

Python
https://www.python.org › downloads › python-3111

Python Release Python 3.11.1

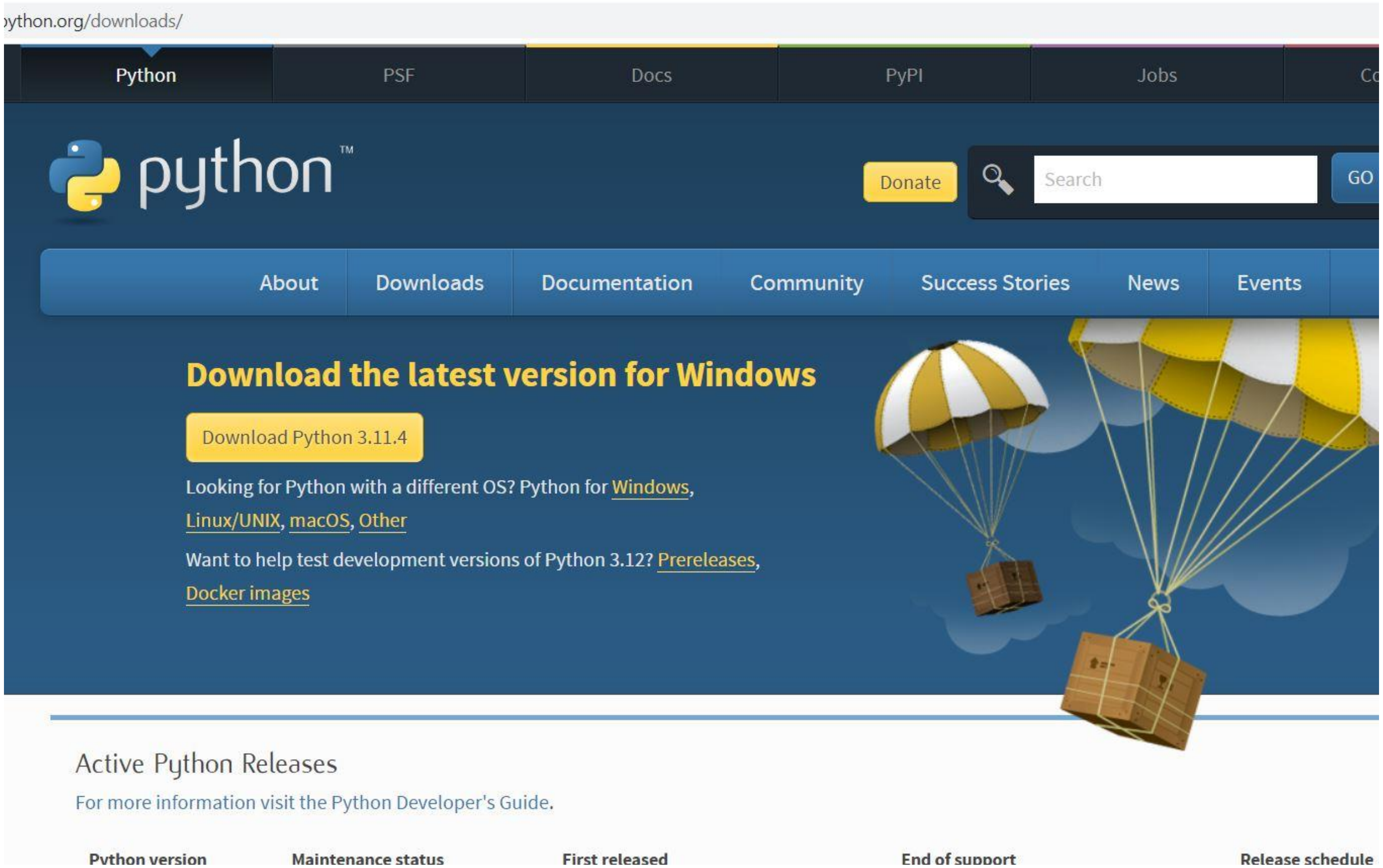
06-Dec-2022 — **Python 3.11**.1 is the newest major release of the **Python** programming language, ... **Python 3.11** is up to **10**-60% faster than **Python** 3.10.

Python
https://www.python.org › downloads › python-3110

Python Release Python 3.11.0

24-Oct-2022 — **Python 3.11**.0 is the newest major release of the **Python** programming language, ... **Python 3.11** is up to **10**-60% faster than **Python** 3.10.

Installation



Download the latest version for Windows

Download Python 3.11.4

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python 3.12? Prereleases, Docker images

Active Python Releases

For more information visit the Python Developer's Guide.

Python version	Maintenance status	First released	End of support	Release schedule

Basic

- Python is an easy-to-learn, powerful programming language. It has efficient high-level *data structures* and a simple but effective approach to *object-oriented programming.*

- Python's elegant (easier) syntax and dynamic typing, together with ***its interpreted nature,*** make it an ideal language for scripting and rapid application development in many areas on most platforms.

- The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python website, https://www.python.org/, and may be freely distributed.

- The same site also contains distributions of and pointers to many free third-party Python modules, programs and tools, and additional documentation.

Basic

- The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C).

- Python is also suitable as an extension language for customizable applications.

If you do much work on computers, eventually you find that there's some task you'd like to automate.

For example, you may wish to perform a search-and-replace over a large number of text files or rename and rearrange a bunch of photo files in a complicated way. Perhaps you'd like to write a small custom database, or a specialized GUI application, or a simple game.

Basic

- Python is simple to use, but it is a real programming language, offering much more structure and support for large programs than shell scripts or batch files can offer.

- On the other hand, Python also offers much more error checking than C, and, being a *very high-level language*, it has high-level data types built in, such as flexible arrays and dictionaries.

- Because of its more general data types Python is applicable to a much larger problem domain.

Basic

- Python allows you to split your program into modules that can be reused in other Python programs.

- It comes with a large collection of standard modules that you can use as the basis of your programs — or as examples to start learning to program in Python.

- Some of these modules provide things like file I/O, system calls, sockets, and even interfaces to graphical user interface toolkits.

- Python is an interpreted language, which can save you considerable time during program development because no compilation and linking is necessary.

- The interpreter can be used interactively, which makes it easy to experiment with features of the language, to write throw-away programs, or to test functions during bottom-up program development. It is also a handy desk calculator.

Using the Python Interpreter

Invoking the Interpreter

```
python3.11
```

Typing an end-of-file character *(Control-D on Unix, Control-Z on Windows)* at the primary prompt causes the interpreter to exit with a zero exit status.

If that doesn’t work, you can exit the interpreter by typing the following command: quit().

A second way of starting the interpreter is the *python -c command [arg] ...*, which executes the statement(s) in the *command*, analogous to the shell’s -c option.

Since Python statements often contain spaces or other characters that are special to the shell, it is usually advised to quote the *command* in its entirety.

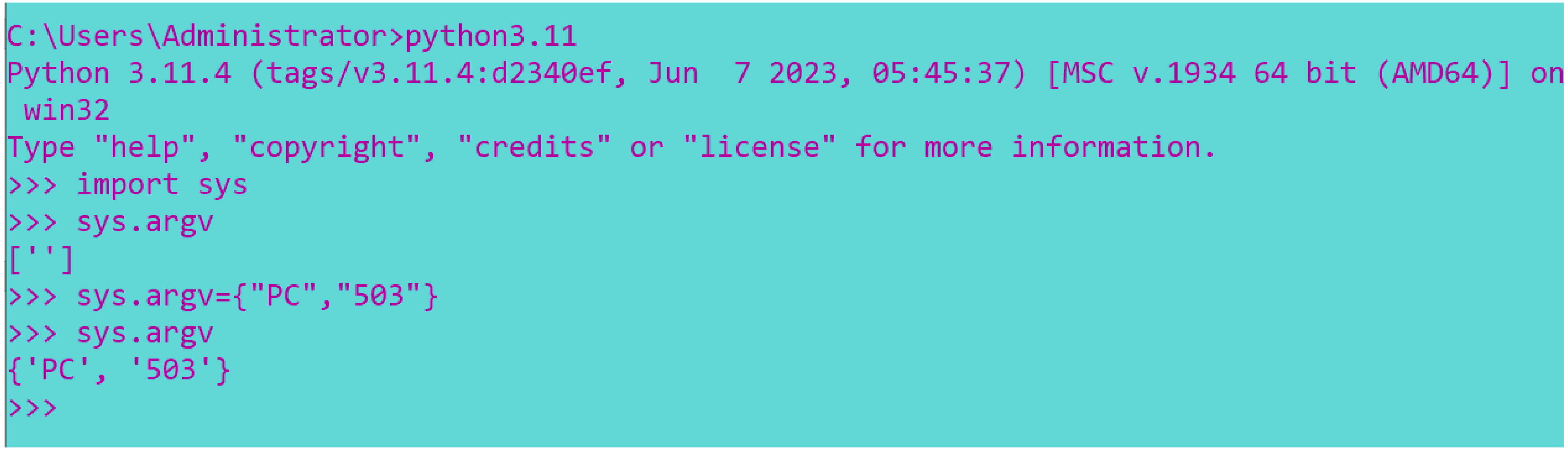
Using the Python Interpreter

Argument Passing

- When known to the interpreter, the script name and additional arguments thereafter are turned into a list of strings and assigned to the ***argv*** variable in the ***sys module***.

- Access this list by executing ***import sys.***

- The length of the list is at least one; when no script and no arguments are given, ***sys.argv[0]*** is an empty string.



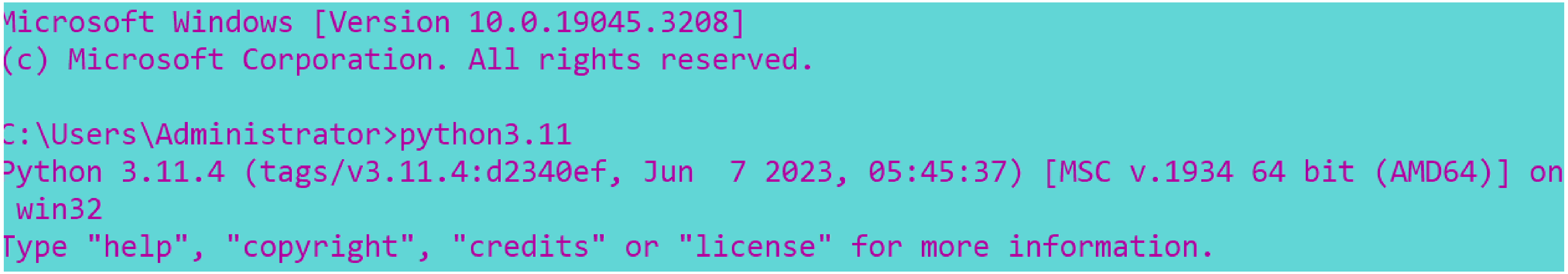
```
C:\Users\Administrator>python3.11
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on
 win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> sys.argv
['']
>>> sys.argv={"PC","503"}
>>> sys.argv
{'PC', '503'}
>>>
```

Using the Python Interpreter

Interactive Mode

- In this mode, it prompts for the next command with the primary prompt, usually three ***greater-than signs (>>>);*** **for continuation lines**, it prompts with the secondary prompt, ***by default three dots (...)***

- The interpreter prints a welcome message stating its version number and a copyright notice before printing the first prompt:



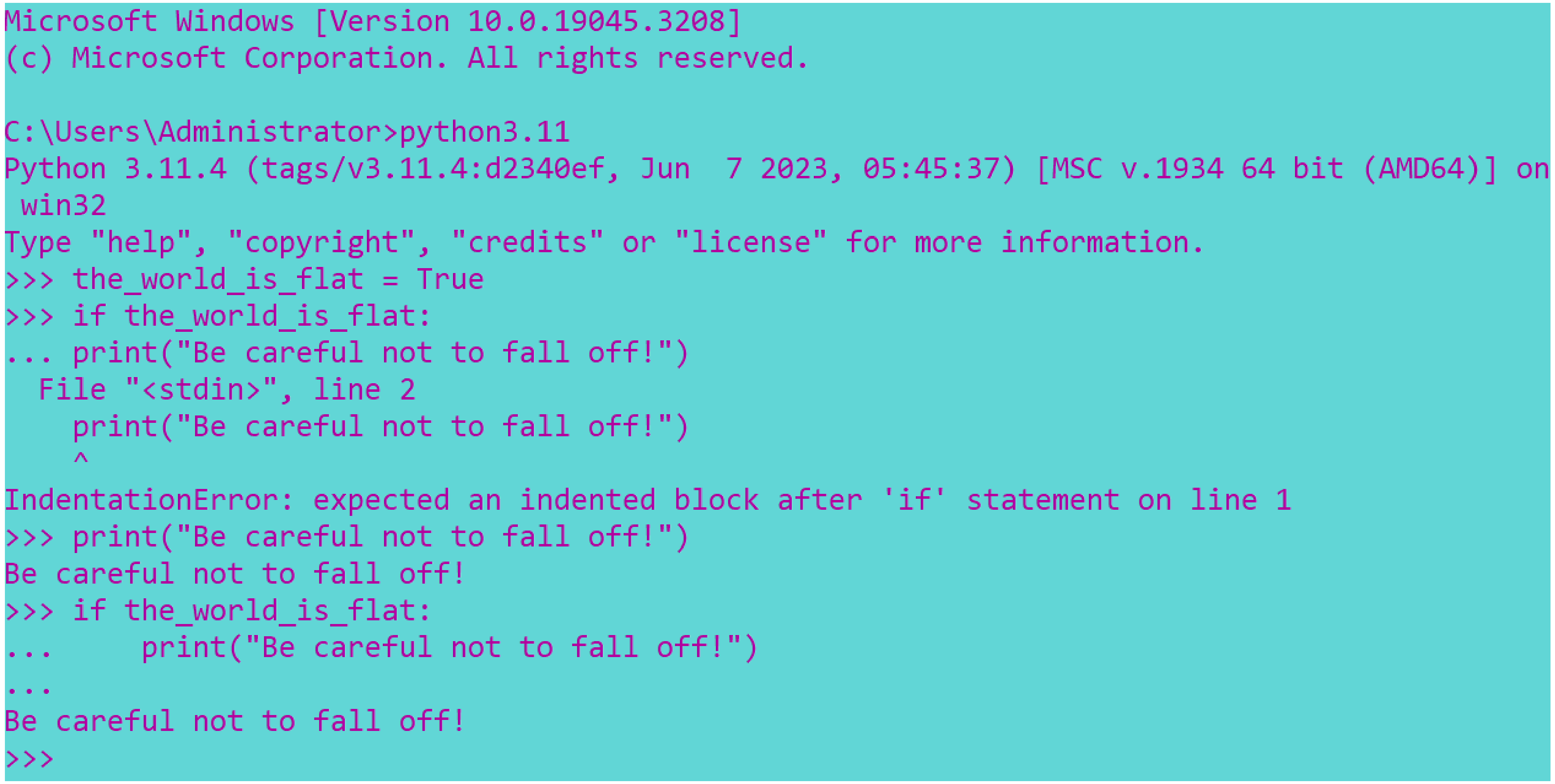
```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>python3.11
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on
 win32
Type "help", "copyright", "credits" or "license" for more information.
```

Using the Python Interpreter

Interactive Mode

Continuation lines are needed when entering a multi-line construct . As an example, take a look at this if statement:



```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>python3.11
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on
 win32
Type "help", "copyright", "credits" or "license" for more information.
>>> the_world_is_flat = True
>>> if the_world_is_flat:
... print("Be careful not to fall off!")
  File "<stdin>", line 2
    print("Be careful not to fall off!")
    ^
IndentationError: expected an indented block after 'if' statement on line 1
>>> print("Be careful not to fall off!")
Be careful not to fall off!
>>> if the_world_is_flat:
...     print("Be careful not to fall off!")
...
Be careful not to fall off!
>>>
```

The Interpreter and Its Environment

Source Code Encoding

- By default, Python source files are treated as encoded in UTF-8.

- In that encoding, characters of most languages in the world can be used simultaneously in string literals, identifiers, and comments— although the standard library only uses ASCII characters for identifiers, a convention that any portable code should follow.

- To display all these characters properly, your editor must recognize that the file is UTF-8, and it must use a font that supports all the characters in the file.

The Interpreter and Its Environment

Source Code Encoding

To declare an encoding other than the default one, a special comment line should be added as the *first* line of the file. The syntax is as follows:

```
# -*- coding: encoding -*-
```

where *encoding* is one of the valid `codecs` supported by Python.

For example, to declare that Windows-1252 encoding is to be used, the first line of your source code file should be:

```
# -*- coding: cp1252 -*-
```

One exception to the *first line* rule is when the source code starts with a UNIX “shebang” line. In this case, the encoding declaration should be added as the second line of the file. For example:

```
#!/usr/bin/env python3
# -*- coding: cp1252 -*-
```

An Informal Introduction to Python

- Comments in Python start with the hash character, #, and extend to the end of the physical line.
- A comment may appear at the start of a line or following whitespace or code, but not within a string literal.
- A hash character within a string literal is just a hash character.
- Since comments are to clarify code and are not interpreted by Python, they may be omitted when typing in examples.

```
>>> #this is the first comment
>>> spam =1 #and this is the second comment
>>>                      # ... and now a third
>>> text = '# This is not a comment because its inside quotes'
>>> text
'# This is not a comment because its inside quotes'
>>> spam =1
>>> spam
1
>>>
```

An Informal Introduction to Python

Using Python as a Calculator

- The interpreter acts as a simple calculator: you can type an expression
- at it and it will write the value. Expression syntax is straightforward:
- the operators +, -, *, and / work just like in most other languages (for example, C);
- parentheses (( )) can be used for grouping.

An Informal Introduction to Python

Using Python as a Calculator

The integer numbers (e.g. 2, 4, 20) have type ***int***, and the ones with a fractional part (e.g. 5.0, 1.6) have type float.

Division (/) always returns a float.

To do floor division and get an integer result you can use the ***// operator***;

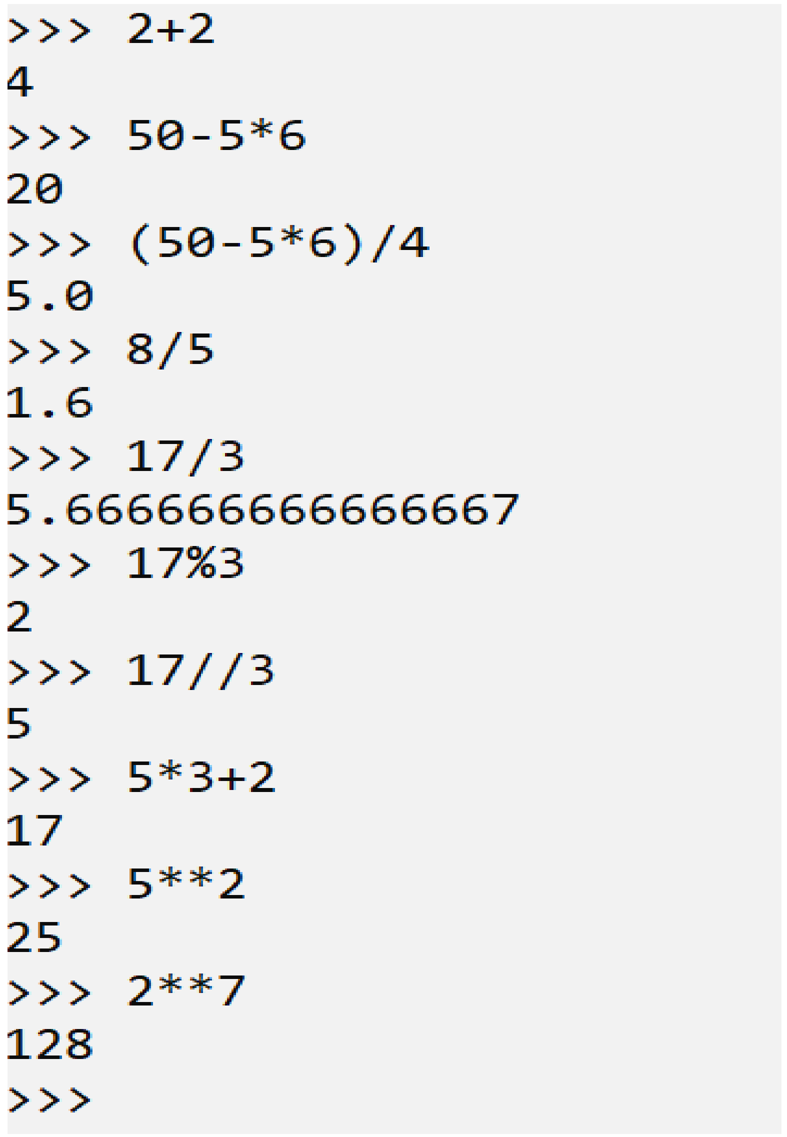
to calculate the remainder ***we can use %:***

With Python, it is possible to use the ***** operator*** *to calculate powers*

An Informal Introduction to Python

Using Python as a Calculator

Try 10 examples by your self.

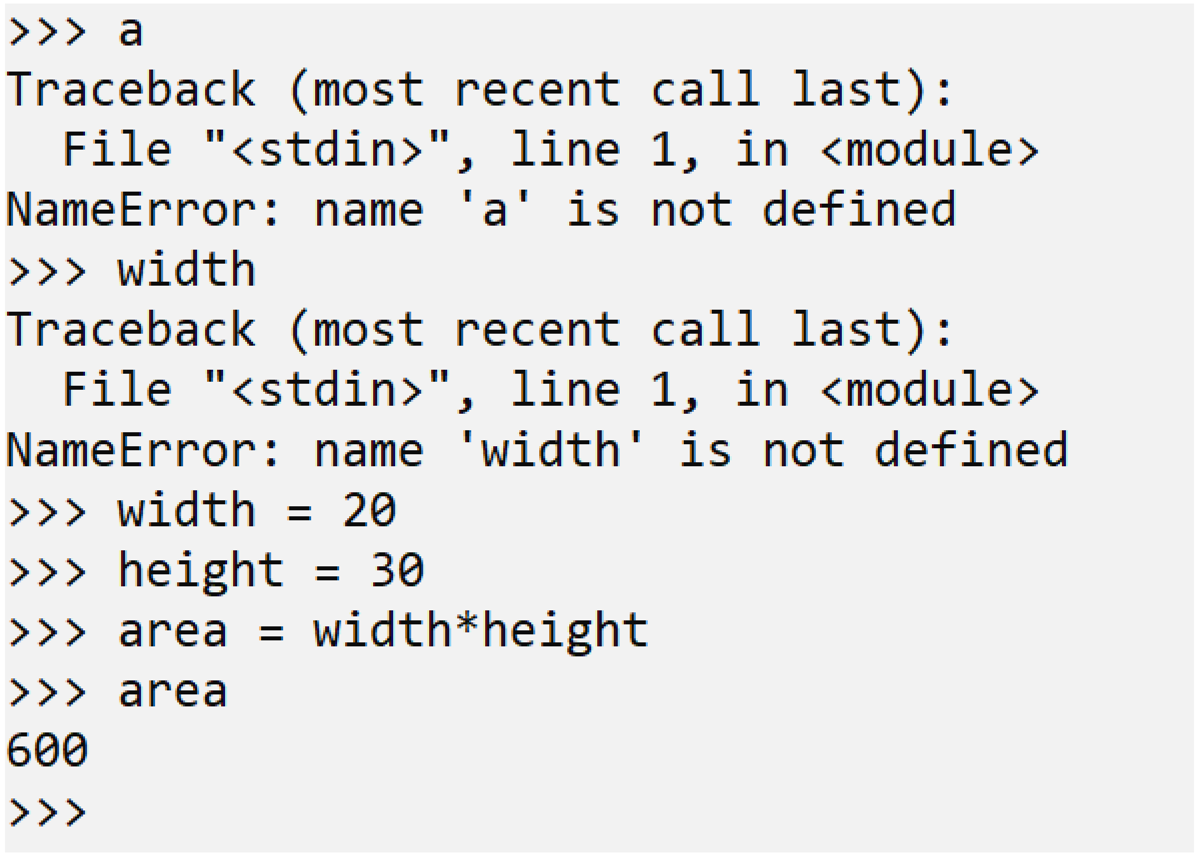


```
>>> 2+2
4
>>> 50-5*6
20
>>> (50-5*6)/4
5.0
>>> 8/5
1.6
>>> 17/3
5.666666666666667
>>> 17%3
2
>>> 17//3
5
>>> 5*3+2
17
>>> 5**2
25
>>> 2**7
128
>>>
```

An Informal Introduction to Python

Using Python as a Calculator

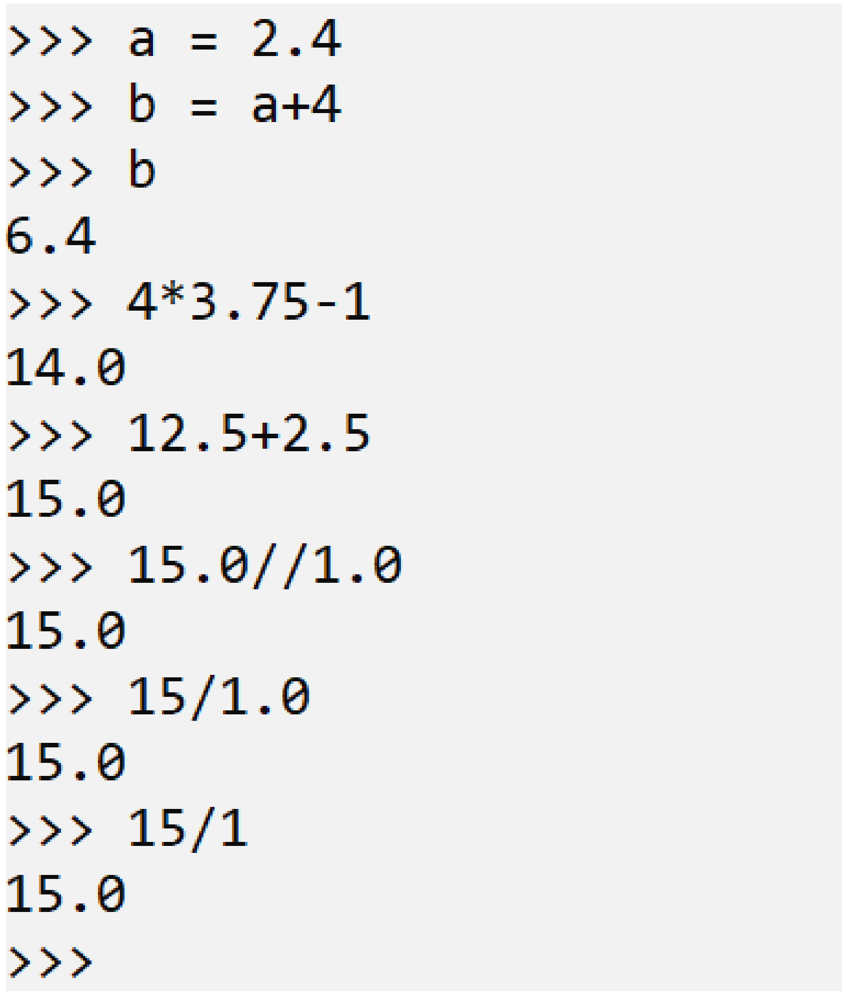
- The equal sign (=) is used to assign a value to a variable.
- Afterward, no result is displayed before the next interactive prompt:



```
>>> a
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'a' is not defined
>>> width
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'width' is not defined
>>> width = 20
>>> height = 30
>>> area = width*height
>>> area
600
>>>
```

An Informal Introduction to Python

The default data type is Python is float rather than integer in c.

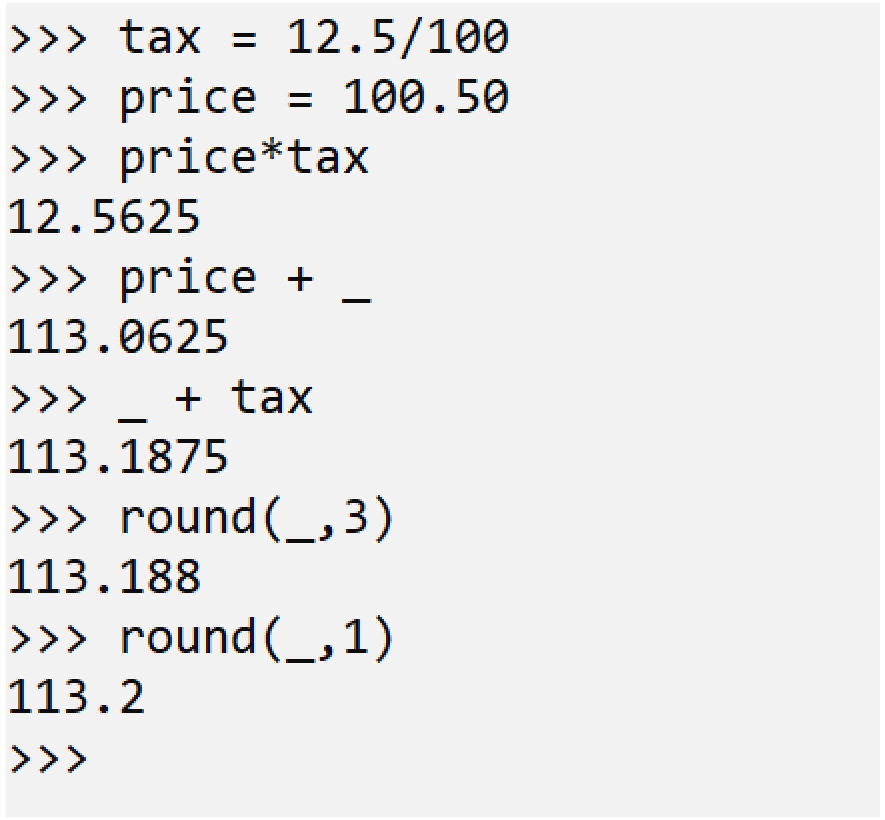


```
>>> a = 2.4
>>> b = a+4
>>> b
6.4
>>> 4*3.75-1
14.0
>>> 12.5+2.5
15.0
>>> 15.0//1.0
15.0
>>> 15/1.0
15.0
>>> 15/1
15.0
>>>
```

An Informal Introduction to Python

- In interactive mode, the last printed expression is assigned to the variable _.
- This means that when you are using Python as a desk calculator,
- it is somewhat easier to continue calculations, for example:

Using similar technique write a program to calculate area of a equilateral triangle and round it by 3 digits.



```
>>> tax = 12.5/100
>>> price = 100.50
>>> price*tax
12.5625
>>> price + _
113.0625
>>> _ + tax
113.1875
>>> round(_,3)
113.188
>>> round(_,1)
113.2
>>>
```