

Chapter 1

An introduction to Python programming

Applied objectives

1. Use IDLE to test Python expressions and statements in the interactive shell.
2. Use IDLE to open, compile, and run a Python source file.

Knowledge objectives

1. List three reasons why Python is a good first language for new programmers.
2. Describe a console program.
3. Explain how Python compiles and runs a program in terms of source code, bytecode, and the virtual machine.
4. Explain how main memory and disk storage work together when a program is running.
5. Distinguish between systems software and application software.
6. Distinguish between testing and debugging.
7. Distinguish between syntax errors and runtime errors.
8. Describe an exception.

Four general-purpose programming languages

- C++
- Java
- C#
- Python

The Python timeline

Year	Month	Release
2000	October	2.0
2008	December	3.0
2010	July	2.7
2020	October	3.9

Syntax differences between Python and Java

Some Java code

```
private static double calculateFutureValue(  
    double monthlyInvestment, double monthlyRate, int months) {  
    double futureValue = 0.0;  
    for (int i = 1; i <= months; i++) {  
        futureValue = (  
            futureValue + monthlyInvestment) * (1 + monthlyRate);  
    }  
    return futureValue;  
}
```

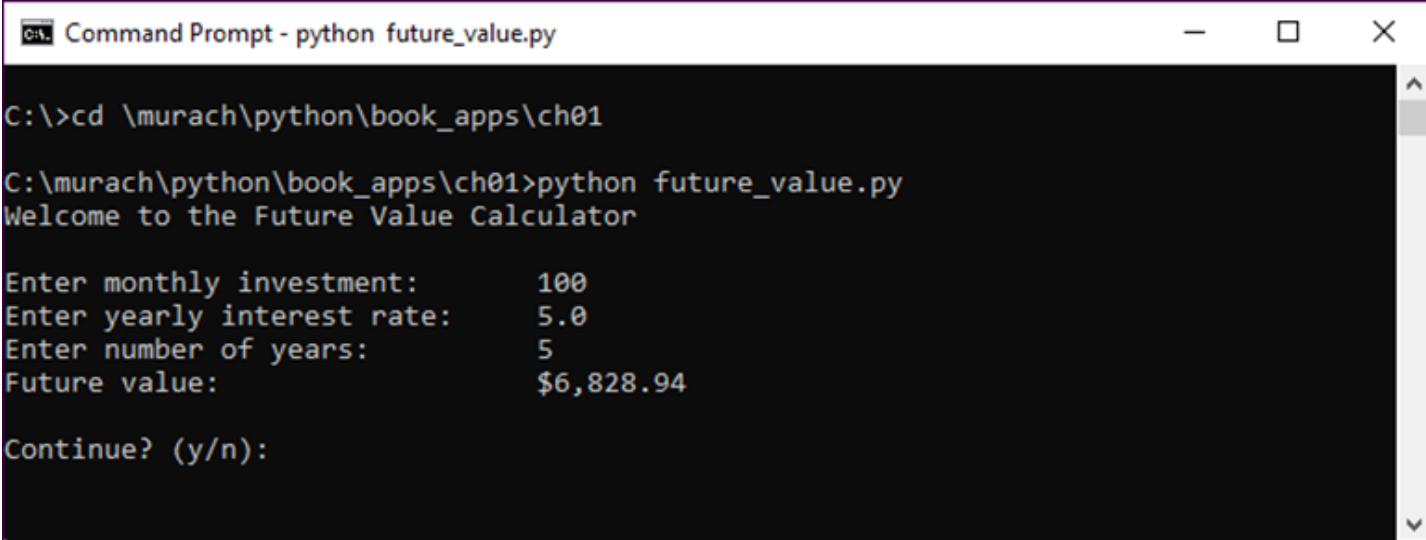
Python code that works the same

```
def calculateFutureValue(monthlyInvestment, monthlyRate, months):  
    futureValue = 0.0  
    for i in range(months):  
        futureValue = (  
            futureValue + monthlyInvestment) * (1 + monthlyRate)  
    return futureValue
```

Why Python is a great first language

- Python has a simple syntax that's easier to read and use than most other languages.
- Python has most of the features of traditional programming languages. As a result, you can use Python to learn concepts and skills that apply to those languages too.
- Python supports the development of a wide range of programs, including games, web applications, and system administration.
- Python is used by many successful companies, including Google, IBM, Disney, and EA Games. As a result, knowing Python is a valuable skill.
- Python is *open source*. There are many advantages to being open source.

A console application



```
Command Prompt - python future_value.py

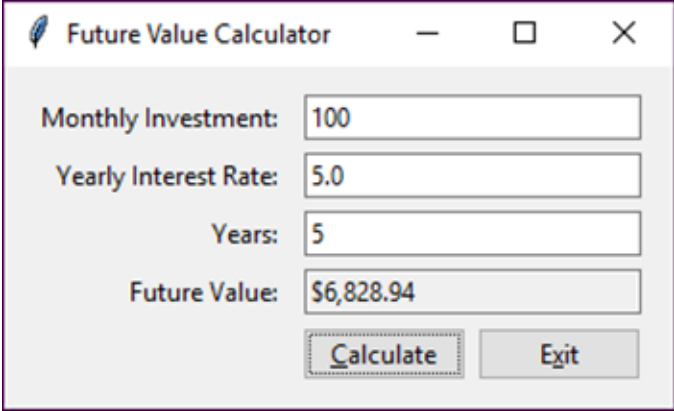
C:\>cd \murach\python\book_apps\ch01

C:\murach\python\book_apps\ch01>python future_value.py
Welcome to the Future Value Calculator

Enter monthly investment:      100
Enter yearly interest rate:    5.0
Enter number of years:        5
Future value:                  $6,828.94

Continue? (y/n):
```


A GUI application

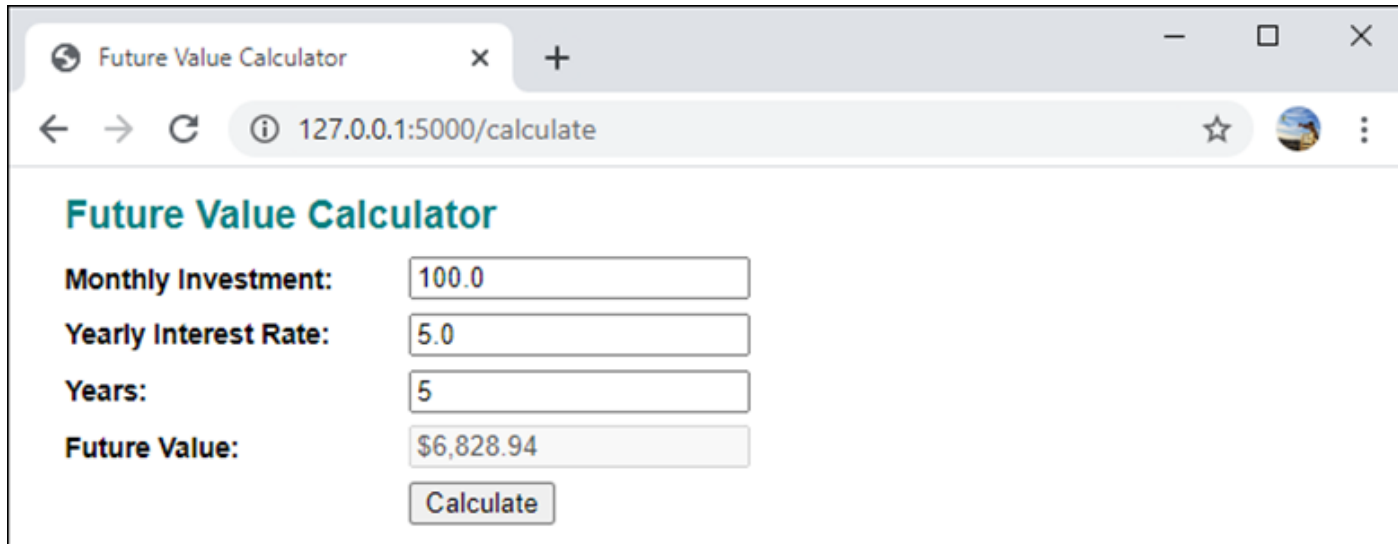


The image shows a screenshot of a graphical user interface (GUI) application titled "Future Value Calculator". The window has a standard title bar with a feather icon, a minus button, a maximize button, and a close button. The main area of the window contains four input fields and one output field, each with a label to its left:

- "Monthly Investment:" followed by a text box containing the value "100".
- "Yearly Interest Rate:" followed by a text box containing the value "5.0".
- "Years:" followed by a text box containing the value "5".
- "Future Value:" followed by a text box containing the calculated value "\$6,828.94".

Below the input fields are two buttons: "Calculate" and "Exit". The "Calculate" button is highlighted with a dashed border, indicating it is the active or default button.

A web application



The screenshot shows a web browser window with a single tab titled "Future Value Calculator". The address bar displays the URL "127.0.0.1:5000/calculate". The page content features the title "Future Value Calculator" in a teal font. Below the title, there are four input fields and one output field, each with a label to its left: "Monthly Investment:" with a value of "100.0", "Yearly Interest Rate:" with a value of "5.0", "Years:" with a value of "5", and "Future Value:" with a value of "\$6,828.94". A "Calculate" button is positioned below the "Future Value:" field.

Label	Value
Monthly Investment:	100.0
Yearly Interest Rate:	5.0
Years:	5
Future Value:	\$6,828.94

Calculate

The source code for a console application (part 1)

```
#!/usr/bin/env python3

import locale

# set the locale for use in currency formatting
locale.setlocale(locale.LC_ALL, 'en_US')

# display a welcome message
print("Welcome to the Future Value Calculator")
print()

choice = "y"
while choice.lower() == "y":
    # get input from the user
    monthly_investment = float(input("Enter monthly investment:\t"))
    yearly_interest_rate = float(input("Enter yearly interest rate:\t"))
    years = int(input("Enter number of years:\t\t"))

    # convert yearly values to monthly values
    monthly_interest_rate = yearly_interest_rate / 12 / 100
    months = years * 12
```

The source code for a console application (part 2)

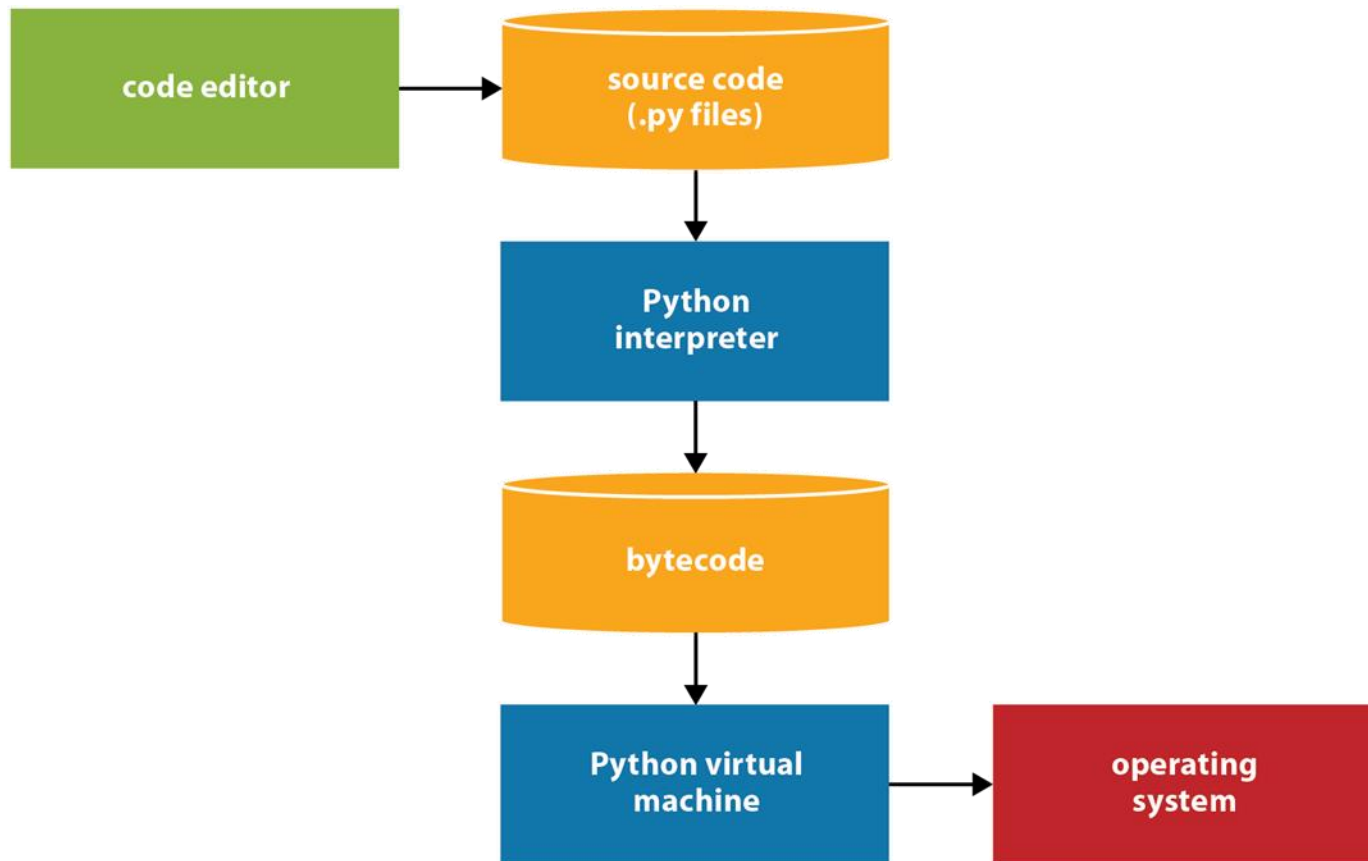
```
# calculate the future value
future_value = 0
for i in range(months):
    future_value = future_value + monthly_investment
    monthly_interest_amount = future_value * monthly_interest_rate
    future_value = future_value + monthly_interest_amount

# format and display the result
print("Future value:\t\t" + locale.currency(
    future_value, grouping=True))
print()

# see if the user wants to continue
choice = input("Continue? (y/n): ")
print()

print("Bye!")
```

How Python compiles and runs source code

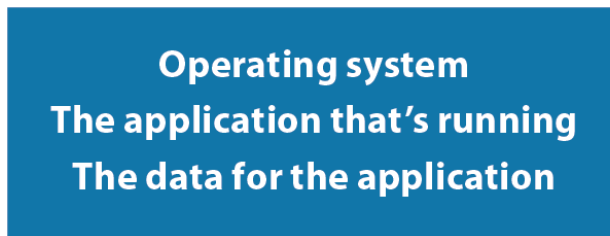


Procedure

- Step 1** The programmer uses a *text editor* or *IDE* to enter and edit the *source code*. Then, the programmer saves the source code to a file with a `.py` extension.
- Step 2** The source code is *compiled* by the Python *interpreter* into *bytecode*.
- Step 3** The bytecode is translated by the Python *virtual machine* into instructions that can interact with the operating system of the computer.

Main memory and disk storage as an application runs

Main memory



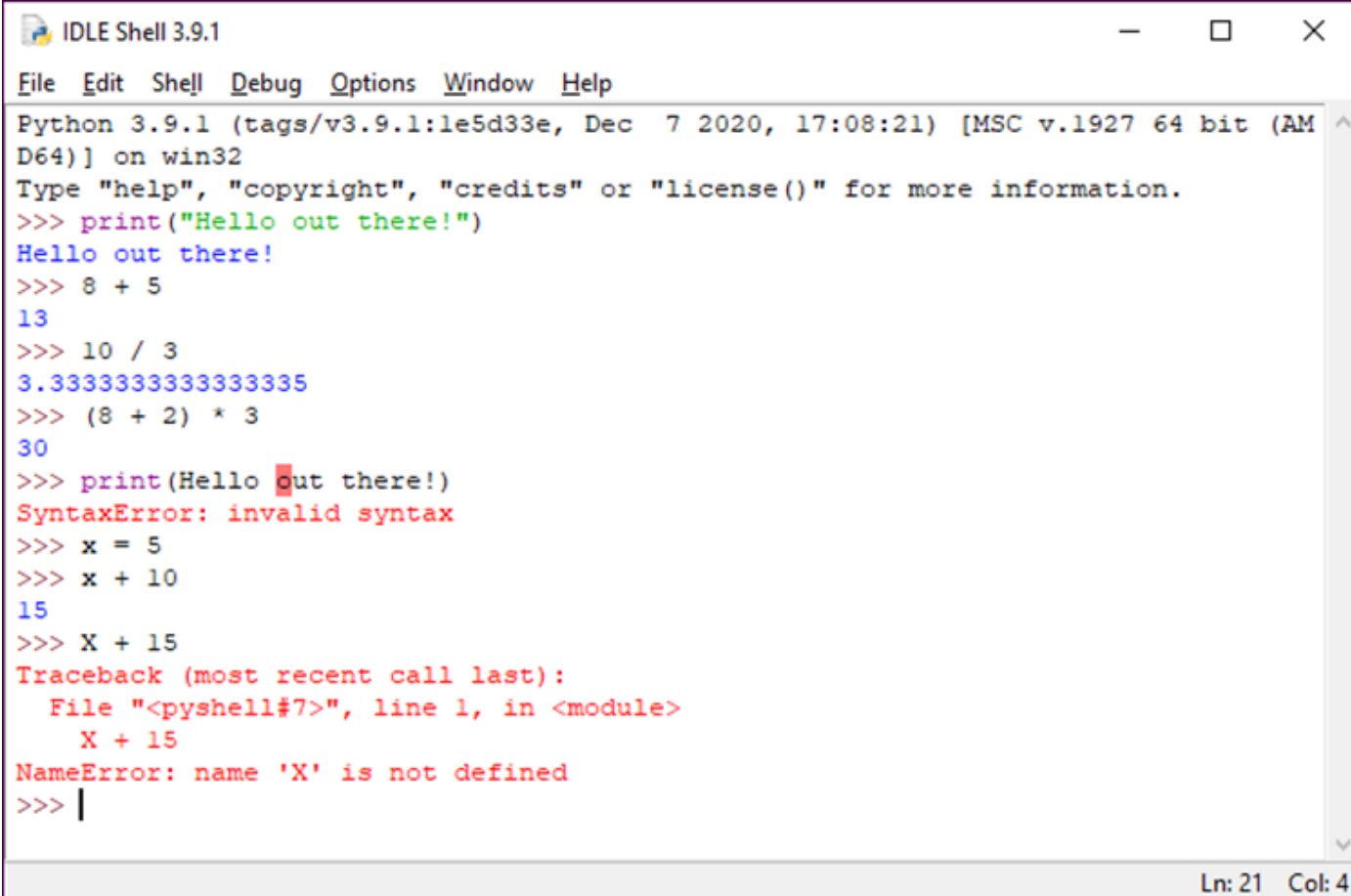
Disk storage



How disk storage and main memory work together

- When you start the computer, it loads the operating system into main memory. Then, you use the features of the operating system to start an application.
- When you start an application, the operating system loads it into main memory. Then, it runs the application.
- As the application runs, it may read data from disk storage into main memory or write data from main memory to disk storage.

IDLE's interactive shell



The screenshot shows the IDLE Shell 3.9.1 window. The title bar reads 'IDLE Shell 3.9.1'. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area contains the following text:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello out there!")
Hello out there!
>>> 8 + 5
13
>>> 10 / 3
3.3333333333333335
>>> (8 + 2) * 3
30
>>> print(Hello out there!)
SyntaxError: invalid syntax
>>> x = 5
>>> x + 10
15
>>> X + 15
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    X + 15
NameError: name 'X' is not defined
>>> |
```

The status bar at the bottom right indicates 'Ln: 21 Col: 4'.

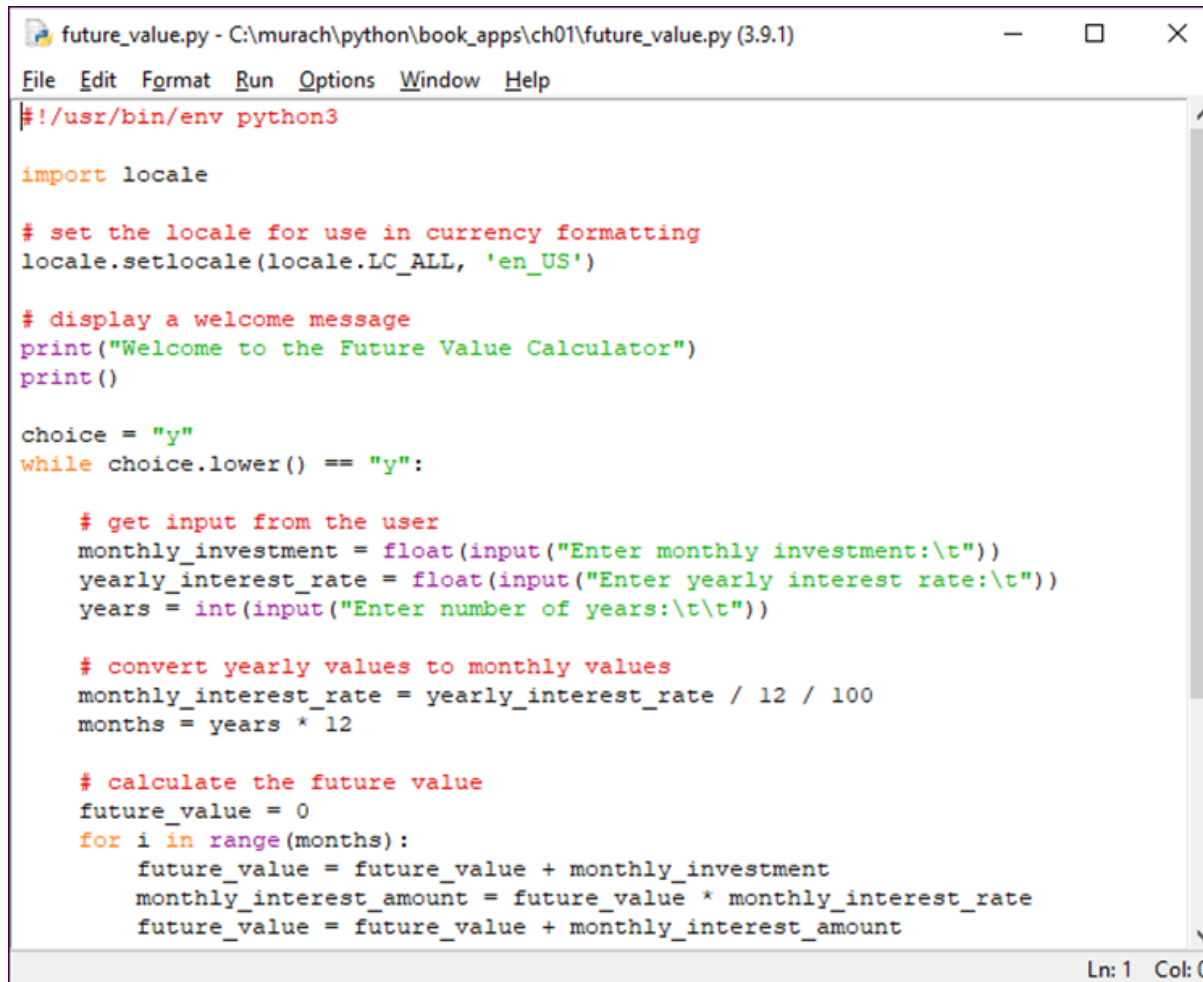
How to open, close, and restart the shell

- To start IDLE, use the features of your operating system. This opens an interactive shell.
- To close the interactive shell, click on its close button or select File→Close.
- To restart an interactive shell, select Run→Python Shell from an interactive shell window.

How to use the interactive shell

- Enter Python code after the `>>>` prompt. Then, press Enter.
- If you enter valid code that produces a result, the shell displays the result.
- If you enter an invalid code, the shell displays an error message.

IDLE's editor with a source file displayed



```
future_value.py - C:\murach\python\book_apps\ch01\future_value.py (3.9.1)
File Edit Format Run Options Window Help
#!/usr/bin/env python3

import locale

# set the locale for use in currency formatting
locale.setlocale(locale.LC_ALL, 'en_US')

# display a welcome message
print("Welcome to the Future Value Calculator")
print()

choice = "y"
while choice.lower() == "y":

    # get input from the user
    monthly_investment = float(input("Enter monthly investment:\t"))
    yearly_interest_rate = float(input("Enter yearly interest rate:\t"))
    years = int(input("Enter number of years:\t\t"))

    # convert yearly values to monthly values
    monthly_interest_rate = yearly_interest_rate / 12 / 100
    months = years * 12

    # calculate the future value
    future_value = 0
    for i in range(months):
        future_value = future_value + monthly_investment
        monthly_interest_amount = future_value * monthly_interest_rate
        future_value = future_value + monthly_interest_amount
```

Ln: 1 Col: 0

How to create, open, save, and close source files

- Use IDLE's File menu and common techniques for your operating system.

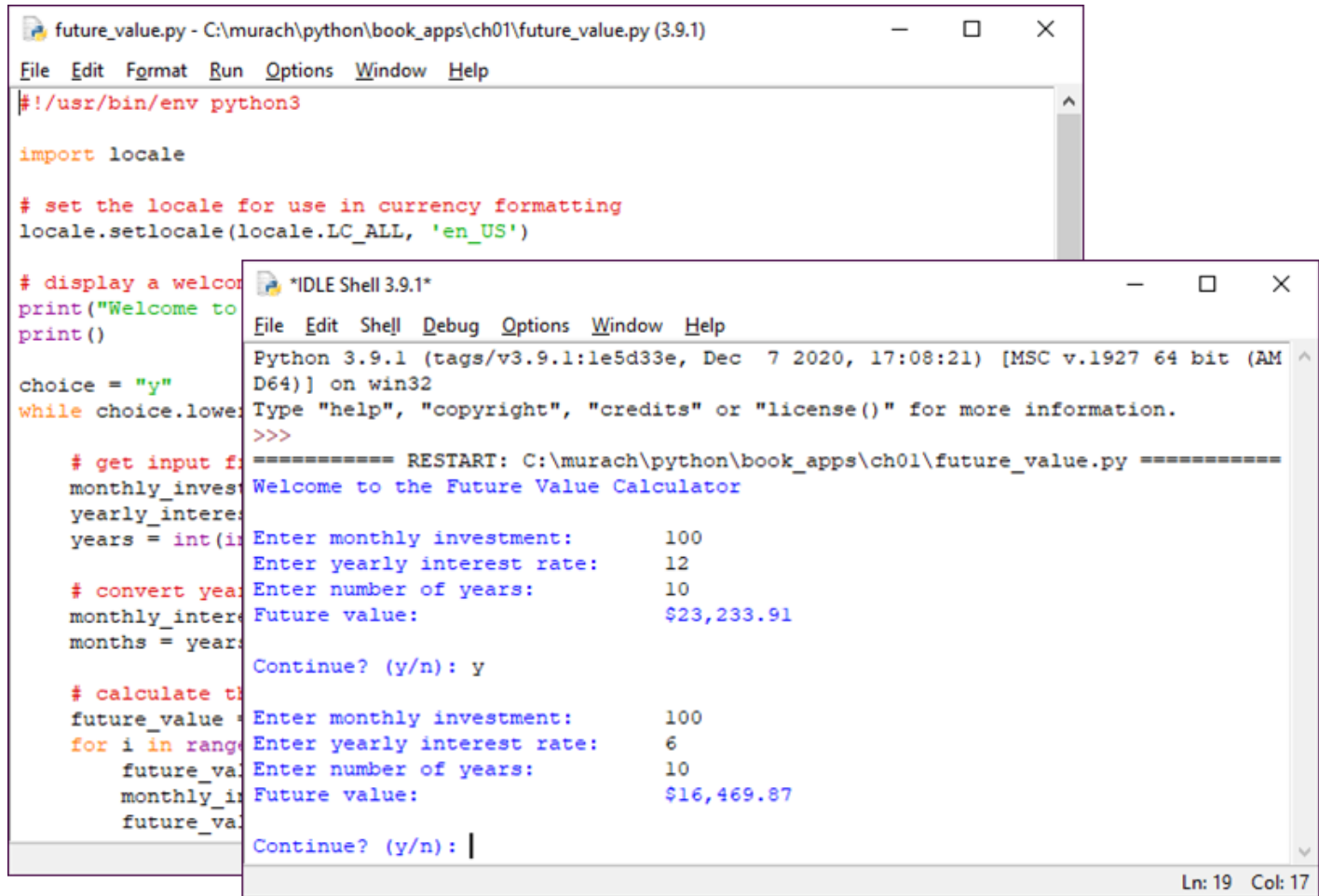
How to switch between source and shell windows

- Use IDLE's Window menu.

How to enter and edit Python code

- Use IDLE's Edit and Format menus, common editing keystrokes, and context menus.

A console application that's being run in the shell



The image shows a Python IDE window titled "future_value.py - C:\murach\python\book_apps\ch01\future_value.py (3.9.1)". The code in the editor includes a shebang line, locale settings, a welcome message, and a loop for a future value calculator. An "IDLE Shell 3.9.1" window is open, showing the execution of the script. It displays the restart command, the welcome message, and two iterations of the calculator with user input and calculated future values.

```
future_value.py - C:\murach\python\book_apps\ch01\future_value.py (3.9.1)
File Edit Format Run Options Window Help
#!/usr/bin/env python3

import locale

# set the locale for use in currency formatting
locale.setlocale(locale.LC_ALL, 'en_US')

# display a welcome message
print("Welcome to the Future Value Calculator")
print()

choice = "y"
while choice.lower() in ("y", "n"):

    # get input from user
    monthly_investment = int(input("Enter monthly investment: "))
    yearly_interest_rate = float(input("Enter yearly interest rate: "))
    years = int(input("Enter number of years: "))

    # convert years to months
    monthly_interest_rate = yearly_interest_rate / 12
    months = years * 12

    # calculate the future value
    future_value = 0
    for i in range(months):
        future_value = (future_value * (1 + monthly_interest_rate) + monthly_investment)

    # display the future value
    print(f"Future value: {future_value}")

    # ask if the user wants to continue
    choice = input("Continue? (y/n): ")

IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\murach\python\book_apps\ch01\future_value.py =====
Welcome to the Future Value Calculator

Enter monthly investment: 100
Enter yearly interest rate: 12
Enter number of years: 10
Future value: $23,233.91

Continue? (y/n): y

Enter monthly investment: 100
Enter yearly interest rate: 6
Enter number of years: 10
Future value: $16,469.87

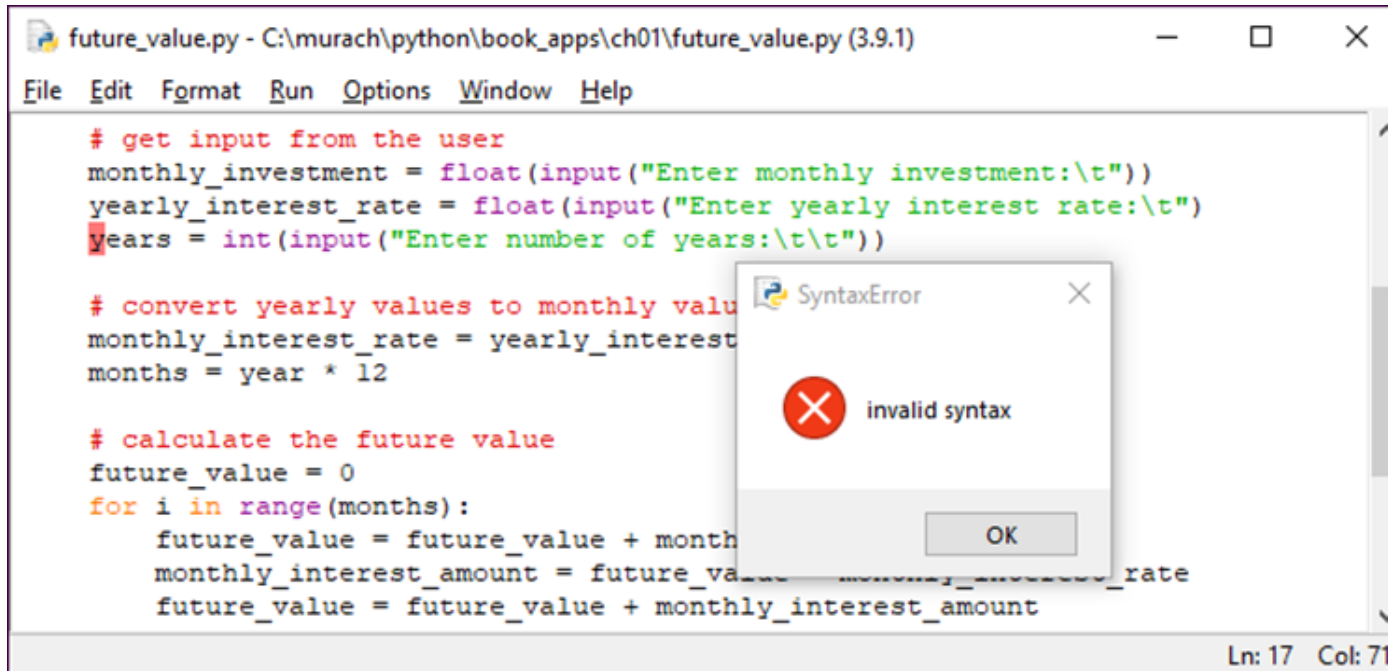
Continue? (y/n): |
```

Ln: 19 Col: 17

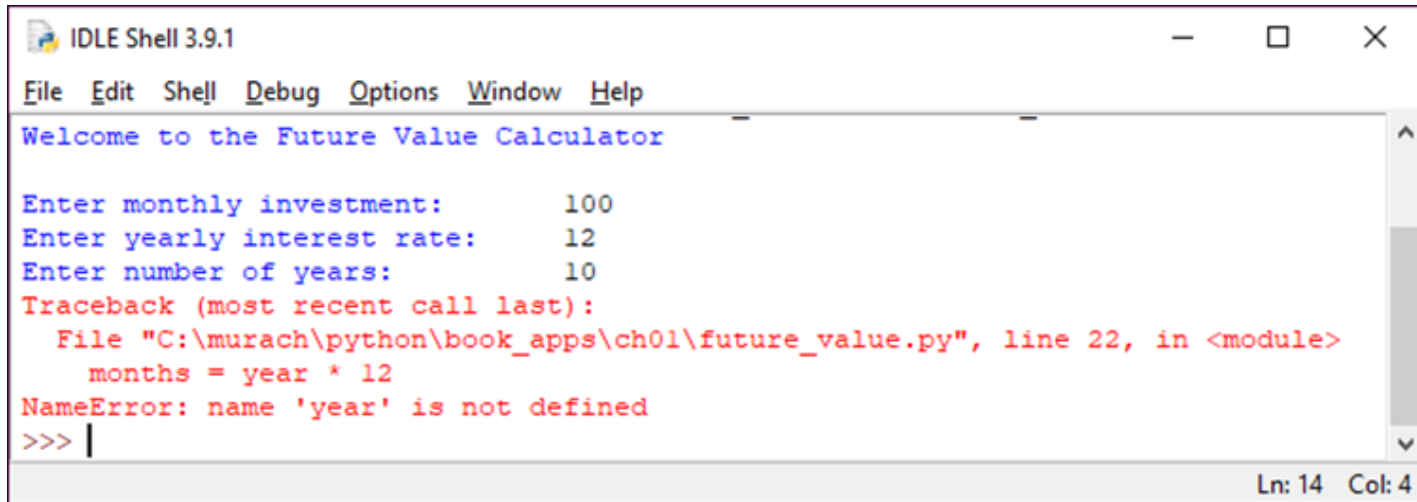
How to compile and run a Python program

- From the editor window, press the F5 key or select Run→Run Module.
- If IDLE displays a dialog box that indicates that you must save the program first, press the Enter key or click Yes to save it. Then, if the program doesn't have any errors, IDLE runs the program in the interactive shell.

A dialog box for a syntax error



A message that's displayed for a runtime error



The screenshot shows the IDLE Shell 3.9.1 window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area displays the following content:

```
Welcome to the Future Value Calculator

Enter monthly investment:      100
Enter yearly interest rate:    12
Enter number of years:        10
Traceback (most recent call last):
  File "C:\murach\python\book_apps\ch01\future_value.py", line 22, in <module>
    months = year * 12
NameError: name 'year' is not defined
>>> |
```

The status bar at the bottom right indicates "Ln: 14 Col: 4".