**HAUB SCHOOL OF BUSINESS**

**SAINT JOSEPH’S UNIVERSITY**

**DSS 615: Python Programming**

**Instructor: Michael Ghen**

**Assignment 2**

By:

Vinayak Suresh Tayshetye (10673718)

**Section 2.1**

**pp 31-35 Exercises 1-30 odd, 68, 69, 72, 73, 77, 78**

In Exercises 1 through 12, evaluate the numeric expression without the computer, and then use Python to check your answer.

**1.** 3\*4

[evaluation]

12

3. 1/(2\*\*3)

[evaluation]

0.125

5. (5-3)\*4

[evaluation]

8

7. 7//3

[evaluation]

2

9. 7%3

[evaluation]

1

11. 5//5

[evaluation]

1

**In Exercises 13 through 18, determine whether the name is a valid variable name.**

13. sales.2008

[evaluation]

Not Valid

15. fOrM\_1040

[evaluation]

Valid

17. expenses?

[evaluation]

Not Valid

**In Exercises 19 through 24, evaluate the numeric expression where a = 2, b = 3, and c = 4.**

19. (a \* b) + c

[evaluation]

10

21. (1 + b) \* c

[evaluation]

16

23. b \*\* (c - a)

[evaluation]

9

**In Exercises 25 through 30, write lines of code to calculate and display the values.**

25. 7.8 + 5

[CODE]

a = 7.8

b = 5

c = a + b

print (c)

[RUN]

12.8

27. 5.5% of 20

[CODE]

a = 20

b = 5.5

c = a \* (b/100)

print (c)

[RUN]

1.1

29. 17(3 + 162)

[CODE]

a = 17

b = 3

c = 162

d = a \* ( b + c )

print (d)

[RUN]

2805

68.

Profit from Stock The following steps calculate the percentage profit from the sale of a stock.

(a) Create the variable purchasePrice and assign it the value 10.

(b) Create the variable sellingPrice and assign it the value 15.

(c) Create the variable percentProfit and assign it 100 times the value of the difference between sellingPrice and purchasePrice divided by purchasePrice.

(d) Display the value of the variable percentProfit.

[CODE]

purchasePrice = 10

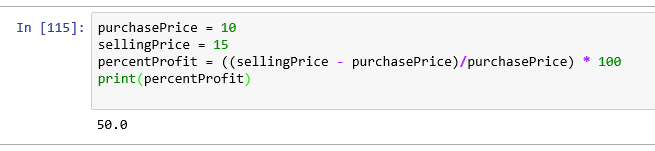
sellingPrice = 15

percentProfit = ((sellingPrice - purchasePrice)/purchasePrice) \* 100

print(percentProfit)

[RUN]

50.0



**In Exercises 69 through 78, write a program to solve the problem and display the answer. The program should use variables for each of the quantities.**

69.

Corn Production Suppose each acre of farmland produces 18 tons of corn. How many tons of corn can be grown on a 30-acre farm?

[CODE]

# Corn production per acre of farmland in tons

production = 18

# Farmland available for corn production in acre

farm = 30

# Calculating total corn production in tons

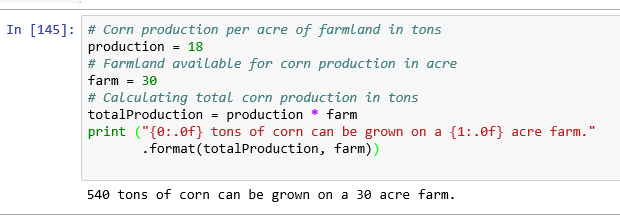
totalProduction = production \* farm

print ("{0:.0f} tons of corn can be grown on a {1:.0f} acre farm."

.format(totalProduction, farm))

[RUN]

540 tons of corn can be grown on a 30 acre farm.



72.

Gas Mileage A motorist wants to determine her gas mileage. At 23,352 miles (on the odometer) the tank is filled. At 23,695 miles the tank is filled again with 14 gallons. How many miles per gallon did the car average between the two fillings?

reading1 = 23352

reading2 = 23695

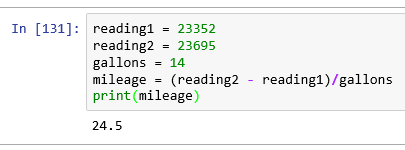
gallons = 14

mileage = (reading2 – reading1)/gallons

print(mileage)

[RUN]

24.5



73.

Water Usage A survey showed that Americans use an average of 1,600 gallons of water per person per day, including industrial use. How many gallons of water are used each year in the United States? Note: The current population of the United States is about 315 million people.

[CODE]

waterUsed = 1600

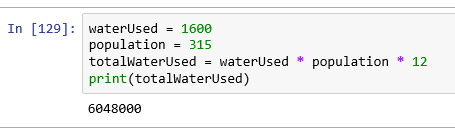
population = 315

totalWaterUsed = waterUsed \* population \* 12

print(totalWaterUsed)

[RUN]

6048000 millions



77.

U.S. National Debt Suppose the U.S. national debt is 1.68\*1013 dollars and the U.S. population is 3.1588\*108. Calculate the per capita U.S. national debt. Display the answer rounded to the nearest whole number.

[CODE]

usDebt = 1.68 \* (10 \*\* 13)

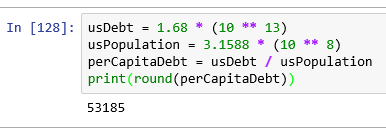
usPopulation = 3.1588 \* (10 \*\* 8)

perCapitaDebt = usDebt / usPopulation

print(round(perCapitaDebt))

[RUN]

53185



78.

Calories Estimate the number of calories in one cubic mile of chocolate ice cream. Note: There are 5,280 feet in a mile and one cubic foot of chocolate ice cream contains about 48,600 calories.

[CODE]

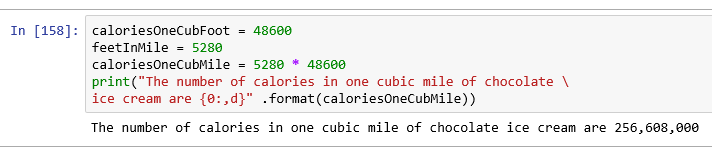
caloriesOneCubFoot = 48600

feetInMile = 5280

caloriesOneCubMile = 5280 \* 48600

print("The number of calories in one cubic mile of chocolate \

ice cream are {0:,d}" .format(caloriesOneCubMile))



**Section 2.2**

pp 43-49 Exercises 46-92 odd, 97, 100, 102, 107, 110, 111

**In Exercises 47 through 70, determine the output displayed by the lines of code.**

47.

a = 4

b = 6

c = "Municipality"

d = "pal"

print(len(c))

print(c.upper())

print(c[a:b] + c[b + 4:])

print(c.find(d))

[RUN]

12

MUNICIPALITY

city

6

**49.**

**print("f" + "lute")**

[RUN]

flute

**51.**

**print("Your age is " + str(21) + ".")**

[RUN]

Your age is 21.

**53.**

**r = "A ROSE"**

**b = " IS "**

**print(r + b + r + b + r)**

[RUN]

A ROSE IS A ROSE IS A ROSE

**55.**

**var = "WALLA"**

**var += var**

**print(var)**

[RUN]

WALLAWALLA

57. **str1 = "good"**

**str1 += "bye"**

**print(str1)**

[RUN]

goodbye

59. **print('M' + ('m' \* 6) + '.')**

[RUN]

Mmmmmmm.

61. **print('a' + (" " \* 5) + 'b')**

[RUN]

a b

63.

**s = "trombones"**

**n = 76**

**print(n, s)**

[RUN]

76 trombones

65.

**num = input("Enter an integer: ")**

**print('1' + str(num))**

(Assume the response is 7.)

[RUN]

Enter an integer: 7

17

67.

**num = float(input("Enter a number: "))**

**print(1 + num)**

(Assume the response is *7*.)

[RUN]

Enter a number: 7

8.0

69.

**film = "the great gatsby".title()[:10].rstrip()**

**print(film, len(film))**

[RUN]

The Great 9

71.

Give a simple expression that lops off the last character of a string.

Print(Vinayak[:-1])

[RUN]

Vinaya

73.

What is the negative index of the first character in a string of eight characters?

[EXPLANATION]

-8

75.

(True or false) If *n* is the length of *str1*, then **str1[n – 1:]** is the string consisting of the last character of *str1*.

[EXPLANATION]

TRUE

77.

(True or false) **str1[:n]** consists of the first *n* characters of *str1*.

[EXPLANATION]

TRUE

**In Exercises 79 through 92, identify all errors.**

79. phoneNumber = 234-5678

print("My phone number is " + phoneNumber)

[EXPLANATION]

TypeError: Traceback error.

The datatype of phoneNumber must be str, not int

81.

for = "happily ever after."

print("They lived " + for)

[EXPLANATION]

SyntaxError: invalid syntax

Because for is a reserved keyword

83.

print('Say it ain't so.')

[EXPLANATION]

SyntaxError: invalid syntax

Because ain’t should be return like this ain\’t

85.

print("Python".Upper())

[EXPLANATION]

AttributeError: 'str' object has no attribute 'Upper'

It is ‘upper’

87.

age = 19

print("Age: " + age)

[EXPLANATION]

TypeError: must be str, not int

91.

language = "Python"

print(language[8])

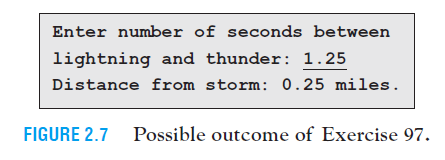
[EXPLANATION]

IndexError: string index out of range

The range of string is from 0 to 5.

97.

**Distance from a Storm** If *n* is the number of seconds between lightning and thunder, the storm is *n*/5 miles away. Write a program that requests the number of seconds between lightning and thunder and reports the distance from the storm rounded to two decimal places. See Fig. 2.7.



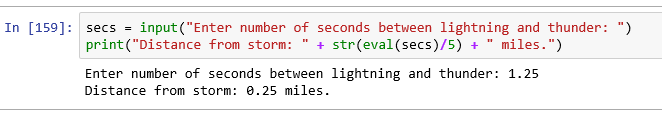
secs = input("Enter number of seconds between lightning and thunder: ")

print("Distance from storm: " + str(eval(secs)/5) + " miles.")

[RUN]

Enter number of seconds between lightning and thunder: 1.25

Distance from storm: 0.25 miles.



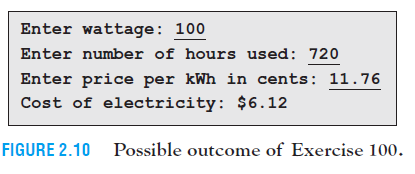
100.

**Cost of Electricity** The cost of the electricity used by a device is given by the formula

cost of electricity (in dollars) = wattage of device . hours used

1,000 **.** cost per kWh (in cents)

where kWh is an abbreviation for “kilowatt hour.” The cost per kWh of electricity varies with locality. Suppose the current average cost of electricity for a residential customer in the United States is 11.76¢ per kWh. Write a program that allows the user to calculate the cost of operating an electrical device. Figure 2.10 calculates the cost of keeping a light bulb turned on for an entire month.



wattageofDevice = eval(input("Enter wattage: "))

hoursUsed = eval(input("Enter number of hours used: "))

costperKWh = eval(input("Enter price per kWh in cents: "))

cost = (wattageofDevice\*hoursUsed)/(1000\*costperKWh)

print("Cost of electricity: $ {0:.2f}".format(cost))

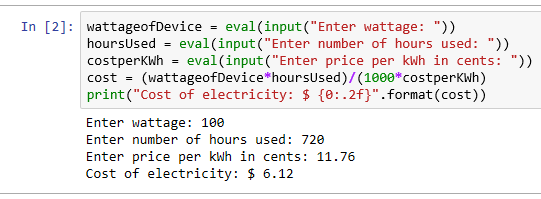
[RUN]

Enter wattage: 100

Enter number of hours used: 720

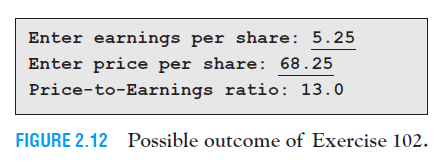
Enter price per kWh in cents: 11.76

Cost of electricity: $ 6.12



102.

**Price-to-EarningsRatio** Write a program that requests a company’s earnings-per-share for the year and the price of one share of stock as input, and then displays the company’s price-to-earnings ratio (that is, price ÷ earnings). See Fig. 2.12.



earningsPerShare = eval(input("Enter earnings per share: "))

pricePerShare = eval(input("Enter price per share: "))

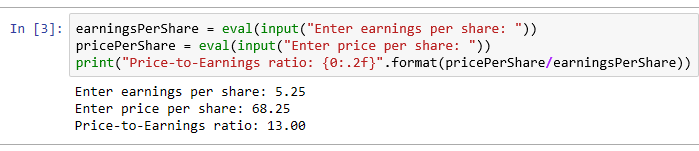
print("Price-to-Earnings ratio: {0:.2f}".format(pricePerShare/earningsPerShare))

[RUN]

Enter earnings per share: 5.25

Enter price per share: 68.25

Price-to-Earnings ratio: 13.00



107.

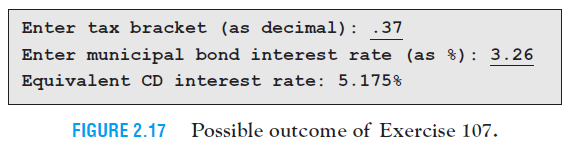
**Equivalent Interest Rates** Interest earned on municipal bonds from an investor’s home state is not taxed, whereas interest earned on CDs *is* taxed. Therefore, in order for a CD to earn as much as a municipal bond, the CD must pay a higher interest rate. How much higher the interest rate must be depends on the investor’s tax bracket. Write a program that requests a tax bracket and a municipal bond interest rate as input, and then displays the CD interest rate having the same yield. See Fig. 2.17.

***Note:*** If the tax bracket is expressed as a decimal, then

***Note:*** If the tax bracket is expressed as a decimal, then

CD interest rate = municipal bond interest rate

(1 - tax bracket)



taxBracket = eval(input("Enter tax bracket (as decimal): "))

municipalIntRate = eval(input("Enter municipal bond interest rate (as %) : "))

CDintRate = municipalIntRate/(1 - taxBracket)

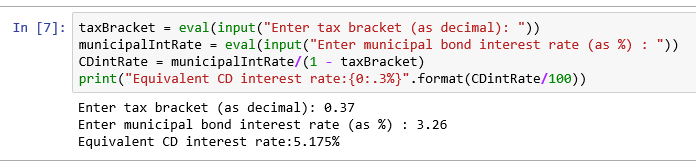
print("Equivalent CD interest rate:{0:.3%}".format(CDintRate/100))

[RUN]

Enter tax bracket (as decimal): 0.37

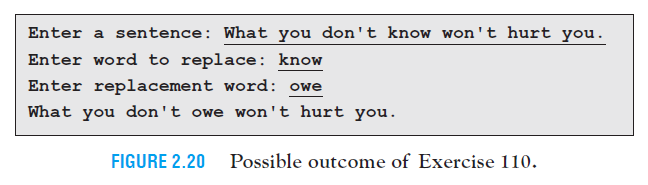
Enter municipal bond interest rate (as %) : 3.26

Equivalent CD interest rate:5.175%



110.

**Word Replacement** Write a program that requests a sentence, a word in the sentence, and another word and then displays the sentence with the first word replaced by the second. See Fig. 2.20.



sentence = input("Enter a sentence: ")

word = input("Enter word to replace: ")

wordToReplace = input("Enter replacement word: ")

wordIndex = sentance.find(word)

wordLength = len(word)

newSentaence = sentance[:wordIndex] + wordToReplace + sentence[wordIndex + wordLength:]

print(newSentence)

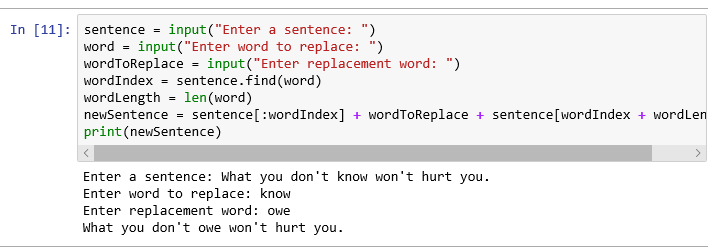
[RUN]

Enter a sentence: What you don't know won't hurt you.

Enter word to replace: know

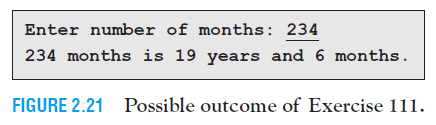
Enter replacement word: owe

What you don't owe won't hurt you.



111.

**Convert Months** Write a program that asks the user to enter a whole number of months as input and then converts that amount of time to years and months. See Fig. 2.21. The program should use both integer division and the modulus operator.



numberOfMonths = eval(input("Enter number of months: "))

years = numberOfMonths // 12

months = numberOfMonths % 12

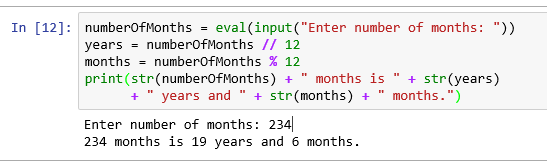
print(str(numberOfMonths) + " months is " + str(years)

+ " years and " + str(months) + " months.")

[RUN]

Enter number of months: 234

234 months is 19 years and 6 months.



**Section 2.3**

pp 54-56 Exercises 30-53 odd, 55, 57, 58

31.

print("{0:10s}{1:^16s} {2:s}".format("Language", "Native speakers",

"% of World Pop."))

print("{0:10s}{1:^16,d}{2:10.2%}".format("Mandarin", 935000000,.141))

print("{0:10s}{1:^16,d}{2:10.2%}".format("Spanish", 387000000,.0585))

print("{0:10s}{1:^16,d}{2:10.2%}".format("English", 365000000,.0552))

[RUN]

Language Native speakers % of World Pop.

Mandarin 935,000,000 14.10%

Spanish 387,000,000 5.85%

English 365,000,000 5.52%

33.

print("Be {0:s} – {1:s} else is taken.".format("yourself", "everyone"))

[RUN]

Be yourself – everyone else is taken.

35.

print("Always {0:s} on the bright side of {1:s}.".format("look", "life"))

[RUN]

Always look on the bright side of life.

37.

x = 3

y = 4

print("The product of {0:d} and {1:d} is {2:d}.".format(x, y, x \* y))

[RUN]

The product of 3 and 4 is 12..

39.

x = 2 # square root of 2 is 1.414213562 to 9 decimal places

print("The square root of {0:n} is about {1:.4f}.".format(x, x \*\* .5))

[RUN]

The square root of 2 is about 1.4142.

41.

str1 = "In a randomly selected group of {0:d} people, the " + \

"probability\nis {1:.2f} that 2 people have the same birthday."

print(str1.format(23, .507397))

[RUN]

In a randomly selected group of 23 people, the probability

is 0.51 that 2 people have the same birthday.

43.

str1 = "You miss {0:.0%} of the shots you never take. - Wayne Gretsky"

print(str1.format(1))

[RUN]

You miss 100% of the shots you never take. - Wayne Gretsky

45.

# 43/193 is .2227979275 to 10 decimal places

print("{0:.2%} of the UN nations are in {1:s}.".format(43/193, "Europe"))

[RUN]

22.28% of the UN nations are in Europe.

47.

print("{0:s}{1:s}{0:s}".format("abra", "cad"))

[RUN]

abracadabra

49.

str1 = "Be {0:s} whenever {1:s}. It is always {1:s}. - Dalai Lama"

print(str1.format("kind", "possible"))

[RUN]

Be kind whenever possible. It is always possible. - Dalai Lama

51.

Do print("Hello") and print("Hello", end="\n") produce the same output?

[ANS]

YES.

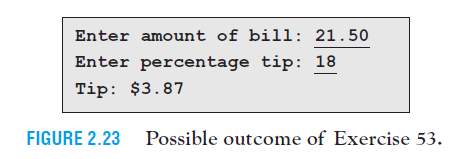


In Exercises 53 through 58, write a program to carry out the stated task.2

53.

Server’s Tip Calculate the amount of a server’s tip, given the amount of the bill

and the percentage tip as input. See Fig. 2.23.



billAmount = eval(input("Enter amount of bill: "))

percentageTip = eval(input("Enter percentage tip: "))

tip = (billAmount \* percentageTip)/100

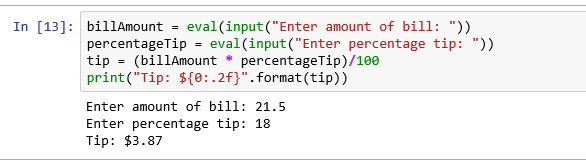
print("Tip: ${0:.2f}".format(tip))

[run]

Enter amount of bill: 21.5

Enter percentage tip: 18

Tip: $3.87



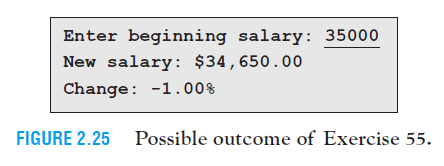
55.

Change in Salary A common misconception is that if you receive a 10% pay raise

and later a 10% pay cut, your salary will be unchanged. Request a salary as input

and then display the salary after receiving a 10% pay raise followed by a 10% pay

cut. The program also should display the percentage change in salary. See Fig. 2.25.



beginingSalary = eval(input("Enter beginning salary: "))

newSalary = (beginingSalary \* 1.1) \* 0.9

change = (newSalary - beginingSalary)/beginingSalary

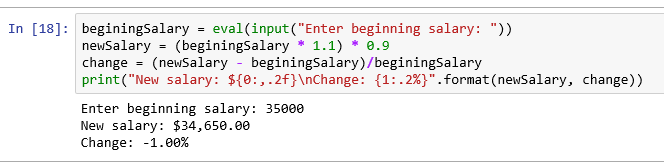
print("New salary: ${0:,.2f}\nChange: {1:.2%}".format(newSalary, change))

[run]

Enter beginning salary: 35000

New salary: $34,650.00

Change: -1.00%



57.

Future Value If P dollars (called the principal) is invested at r% interest compounded

annually, then the future value of the investment after n years is given by

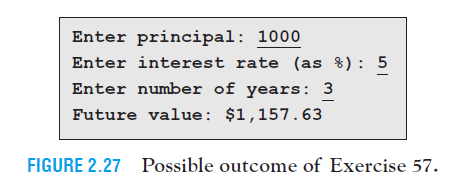
the formula

future value = P (1 + r/100 )n .

Calculate the future value of an investment after the user enters the principal, interest

rate, and number of years. Figure 2.27 shows that $1,000 invested at 5% interest

will grow to $1,157.63 in 3 years.



principal = float(input("Enter principal: "))

intRate = float(input("Enter interest rate (as %): "))

years = int(input("Enter number of years: "))

futureValue = principal \* ((1 + (intRate / 100)) \*\* years)

print("Future value: ${:,.2f}".format(futureValue))

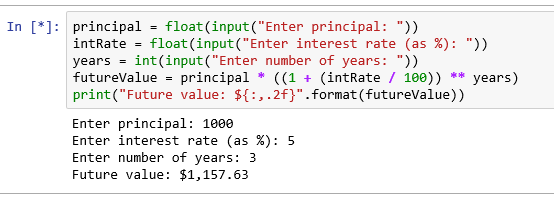
[run]

Enter principal: 1000

Enter interest rate (as %): 5

Enter number of years: 3

Future value: $1,157.63



58.

Present Value The present value of f dollars at interest rate r% compounded annually

for n years is the amount of money that must be invested now in order to

grow to f dollars (called the future value) in n years where the interest rate is r% per

year. The formula for present value is

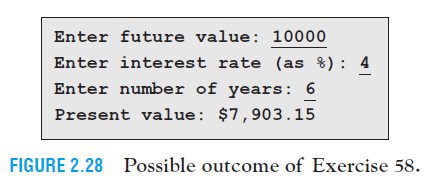
future value = f .

(1 + r/100 )n .

Calculate the present value of an investment after the user enters the future value,

interest rate, and number of years. Figure 2.28 shows that at 4% interest per year,

$7,903.15 must be invested now in order to have $10,000 after 6 years.



futureValue = float(input("Enter future value: "))

intRate = float(input("Enter interest rate (as %): "))

years = int(input("Enter number of years: "))

presentValue = futureValue / ((1 + (intRate / 100)) \*\* years)

print("Present value: ${:,.2f}".format(presentValue))

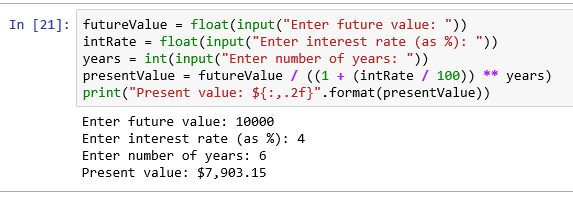
[run]

Enter future value: 10000

Enter interest rate (as %): 4

Enter number of years: 6

Present value: $7,903.15



**Section 2.4**

pp 66-71 Exercises 42-100 odd, 101, 102, 103

**In Exercises 1 through 48, assume that the list states contains the names of the fifty U.S. states in the order they joined the union, and determine the output displayed by the lines of code.**

**(The output sequence to the below questions may vary as EVEN questions were not considered.)**

43.

states[1] = "Commonwealth of Pennsylvania"

print(states[:3])

[RUN]

['United States', 'Commonwealth of Pennsylvania', 'Pennsylvania']

45.

print(states[-4].split())

print(states[2].split())

[RUN]

["Seward's", 'Folly'] ['Pennsylvania']

47.

print((',').join(states[1:4]))

[RUN]

Commonwealth of Pennsylvania,Pennsylvania,New Jersey

In Exercises 49 through 54, assume that list1 contains 100 items. Determine the number

of items in each of the slices.

49.

list1[-8:]

[]

8 ITEMS

51.

list1[:]

[]

100 ITEMS

53.

list1[8:8]

[]

0 ITEMS

In Exercises 55 through 58, assume that the list nums = [6, 2, 8, 0], and determine the

output displayed by the line of code.

55.

print("Largest Number:", max(nums))

[RUN]

Largest Number: 8

57.

print("Total:", sum(nums))

[RUN]

Total: 16

In Exercises 59 through 94, determine the output displayed by the lines of code.

59.

L= ["sentence", "contains", "five", "words."]

L.insert(0, "This")

print(" ".join (L))

del L[3]

L.insert(3, "six")

L.insert(4, "different")

print(" ".join (L))

[RUN]

This sentence contains five words.

This sentence contains six different words.

61.

name = input("Enter name with two parts: ")

L = name.split()

print("{0:s}, {1:s}".format(L[1], L[0]))

(Assume the name entered is Charles Babbage.)

[RUN]

Enter name with two parts: Charles Babbage

Babbage, Charles

63.

name = input("Enter name with three parts: ")

L = name.split()

print("Middle Name:", L[1])

(Assume the name entered is Guido van Rossum.)

[RUN]

Enter name with three parts: Guido van Rossum

Middle Name: van

65.

tuple1 = ("course", "of", "human", "events", "When", "in", "the")

tuple2 = tuple1[4:] + tuple1[:4]

print((" ".join(tuple2)))

[RUN]

When in the course of human events

67.

headEditor = ["editor", "in", "chief"]

print(('-').join(headEditor))

[RUN]

editor-in-chief

69.

motto = ["e", "pluribus", "unum"]

print(("\*\*").join(motto))

[RUN]

e\*\*pluribus\*\*unum

71.

state = "New York,NY,Empire State,Albany"

stateFacts = state.split(',')

print(stateFacts)

[RUN]

['New York', 'NY', 'Empire State', 'Albany']

73.

nations = "France\nEngland\nSpain\n"

countries = nations.split()

print(countries)

[RUN]

['France', 'England', 'Spain']

75.

# The three lines of Dev.txt contain mer, gram, pro

infile = open("Dev.txt", 'r')

dev = [line.rstrip() for line in infile]

infile.close()

dev[0], dev[-1] = dev[-1], dev[0]

word = ("").join(dev)

print(word)

[RUN]

grampromer

77.

# The three lines of Star.txt contain your, own, star.

infile = open("Star.txt", 'r')

words = [line.rstrip() for line in infile]

infile.close()

words.insert(0, "Follow")

quote = (" ").join(words)

print(quote)

[RUN]

Follow your own star.

79.

phoneNumber = "9876543219"

list1 = list(phoneNumber)

list1.insert(3, '-')

list1.insert(7, '-')

phoneNumber = "".join(list1)

print(phoneNumber)

[RUN]

987-654-3219

81.

nums = (3, 9, 6)

print(list(nums))

[RUN]

[3, 9, 6]

83.

word = "etch"

L = list(word)

L[1] = "a"

print("".join(L))

[RUN]

each

85.

list1 = ["soprano", "tenor"]

list2 = ["alto", "bass"]

list1.extend(list2)

print(list1)

[RUN]

['soprano', 'tenor', 'alto', 'bass']

87.

list1 = ["gold"]

list2 = ["silver", "bronze"]

print(list1 + list2)

[RUN]

['gold', 'silver', 'bronze']

89.

list1 = ["mur"] \* 2

print("".join(list1))

[RUN]

murmur

91.

t = ("Dopey", "Sleepy", "Doc", "Grumpy", "Happy", "Sneezy", "Bashful")

print(t[4:20])

[RUN]

('Happy', 'Sneezy', 'Bashful')

93.

answer = ["Yes!", "No!", "Yes!", "No!", "Maybe."]

num = answer.index("No!")

print(num)

[RUN]

1

In Exercises 95 through 100, identify all errors.

95.

threeRs = ["reading", "riting", "rithmetic"]

print(threeRs[3])

[RUN]

IndexError: list index out of range

Because the list ranges from 0 to 2.

97.

list1 = [1, "two", "three", 4]

print(" ".join(list1))

[RUN]

TypeError: sequence item 0: expected str instance, int found

99.

title = ("The", "Call", "of", "the", "Wild")

title[1] = "Calm"

print(" ".join(title))

[RUN]

TypeError: 'tuple' object does not support item assignment

100.

words = ("Keep", "cool", "but", "don't")

words.append("freeze.")

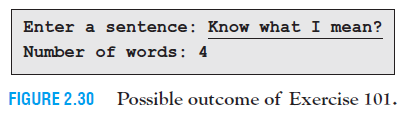
print(words)

[RUN]

AttributeError: 'tuple' object has no attribute 'append'

101.

Analyze a Sentence Write a program that counts the number of words in a sentence input by the user. See Fig. 2.30.



sentence = input("Enter a sentence: ")

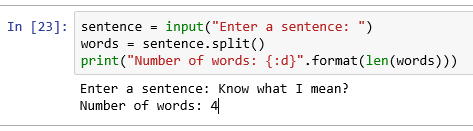
words = sentence.split()

print("Number of words: {:d}".format(len(words)))

[RUN]

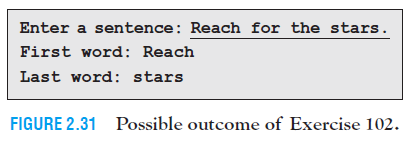
Enter a sentence: Know what I mean?

Number of words: 4



102.

Analyze a Sentence Write a program that displays the first and last words of a sentence input by the user. See Fig. 2.31. Assume that the only punctuation is a period at the end of the sentence.



sentence = input("Enter a sentence: ")

words = sentence.split()

#Strip last word from the period.

lastWord = words[-1].split(".")

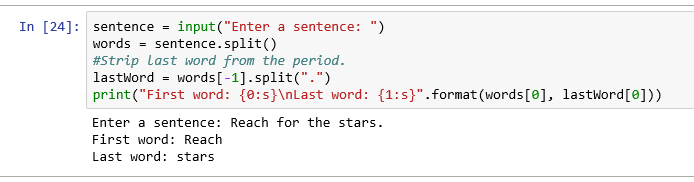
print("First word: {0:s}\nLast word: {1:s}".format(words[0], lastWord[0]))

[RUN]

Enter a sentence: Reach for the stars.

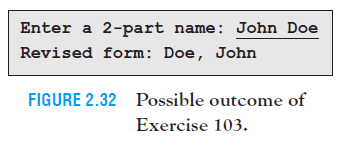
First word: Reach

Last word: stars



103.

Name Write a program that requests a two-part name and then displays the name in the form "lastName, firstName". See Fig. 2.32.



name = input("Enter a 2-part name: ")

words = name.split()

print("Revised form: {0:s}, {1:s}".format(words[1], words[0]))

[RUN]

Enter a 2-part name: John Doe

Revised form: Doe, John

