5602201

2. Review: Python Programming and Syntax Accessing data with Python

CHUTIPORN ANUTARIYA


LET'S
START

What is Python?

Python ...

Python was developed in 1990s by Guido van Rossum, Netherlands.

Python has a simple syntax, easier to read and use than most other languages.

Python has most of the features of traditional programming languages, can be used to develop a wide range of programs, including web apps, games, etc.

Python is popular and used widely!!!

Python is open source.

Tools to be used

- Python
- Visual Studio Code
- MySQL Server
- MySQL Workbench
- MongoDB

Python At First Glance

```
import math

def showArea(shape):
    print "Area = %d" % shape.area()

def widthOfSquare(area):
    return math.sqrt(area)

class Rectangle(object):

    def __init__(self, width, height):
        self.width = width
        self.height = height

    def area(self):
        return self.width * self.height

###### Main Program ######
r = Rectangle(10, 20)

showArea(r)
```

Import a library module

Function definition

Class definition

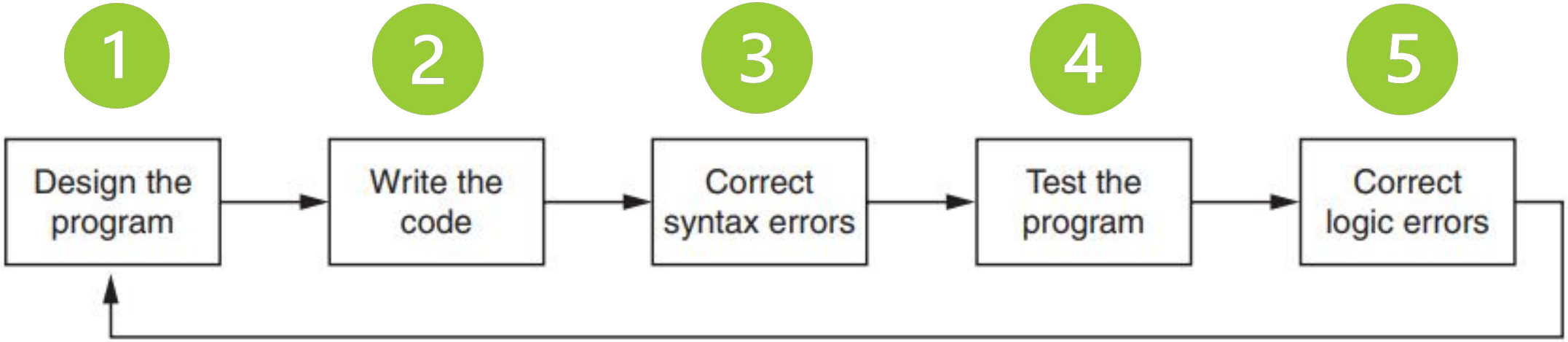
Comment

Object instantiation

Calling a function

Program Development

Program Development Cycle

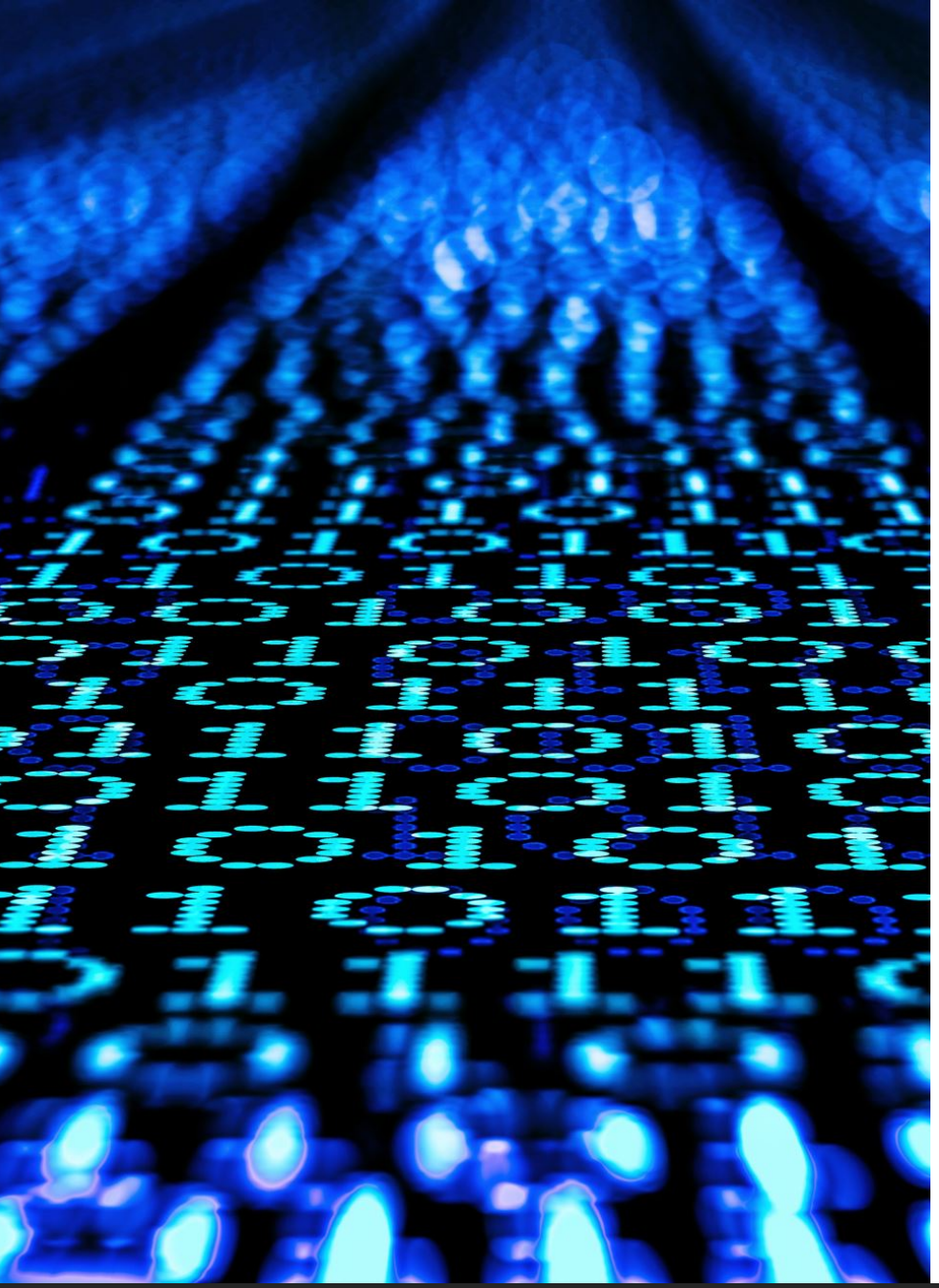


1.1 Understand the task that the program is to perform.

1.2. Determine the steps that must be taken to perform the task.

- Algorithms
- Pseudocode
- Flowcharts

Python Syntax Review!



Variables

Variables

A variable is a name that represents a storage location in the computer's memory.

Naming Rules:

- A variable name must start with a letter or the underscore character.
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive

Valid Variable Names???

units_per_day

dayOfWeek

3dGraph

June1997

Mixture#3

Variable Assignment Statement

Syntax:

```
variable = expression
```

Example:

```
nickname = 'Kitty'
age = 32
years_in_service = 17
dollarToThai = 32.80
```

Consider different data types assigned to different variables

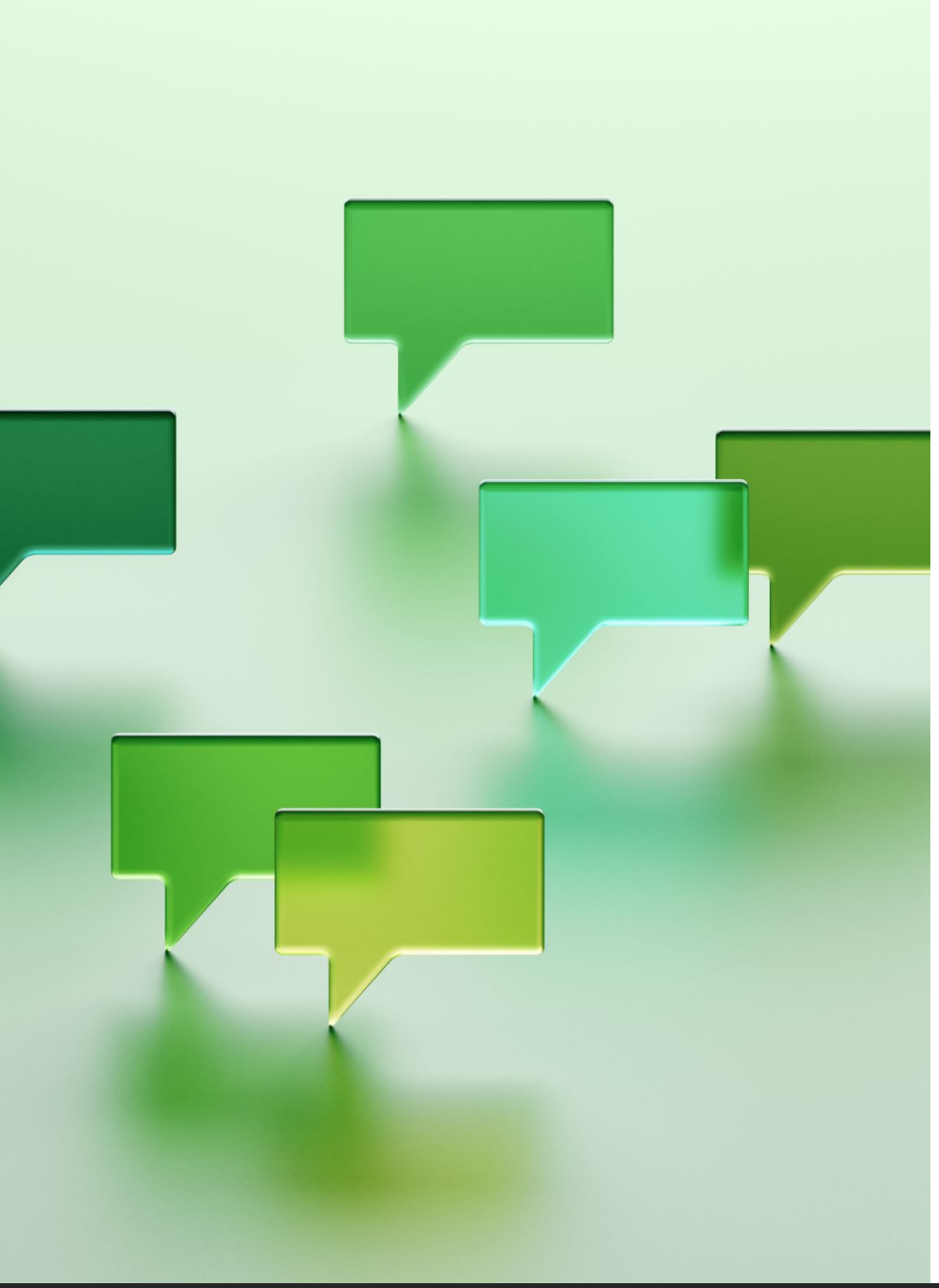
Program 2-9 (variable_demo3.py)

```
1  # This program demonstrates a variable.
2  room = 503
3  print('I am staying in room number', room)
```

Program Output

Comments

- **Comments** are notes of explanation that document lines or sections of a program.
- They are intended for people who may be reading the source code.
- Lines starting in # are comments to the user
- You can leave comments for yourself, of future readers of your code!



Comments

Program 2-5 (comment1.py)

```
# This program displays a person's
# name and address.
print('Kate Austen')
print('123 Full Circle Drive')
print('Asheville, NC 28899')
```

Different ways to comment

Program 2-6 (comment2.py)

```
print('Kate Austen')              # Display the name.
print('123 Full Circle Drive')    # Display the address.
print('Asheville, NC 28899')      # Display the city, state, and ZIP.
```

Checkpoint 4

6. Look at the following assignment statements:

```
value1 = 99
value2 = 45.9
value3 = 7.0
value4 = 7
value5 = 'abc'
```

After these statements execute, what is the Python data type of the values referenced by each variable?

Reading Input from Keyboard

input

Syntax:

```
variable = input(prompt)
```

Example:

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

Data Conversion

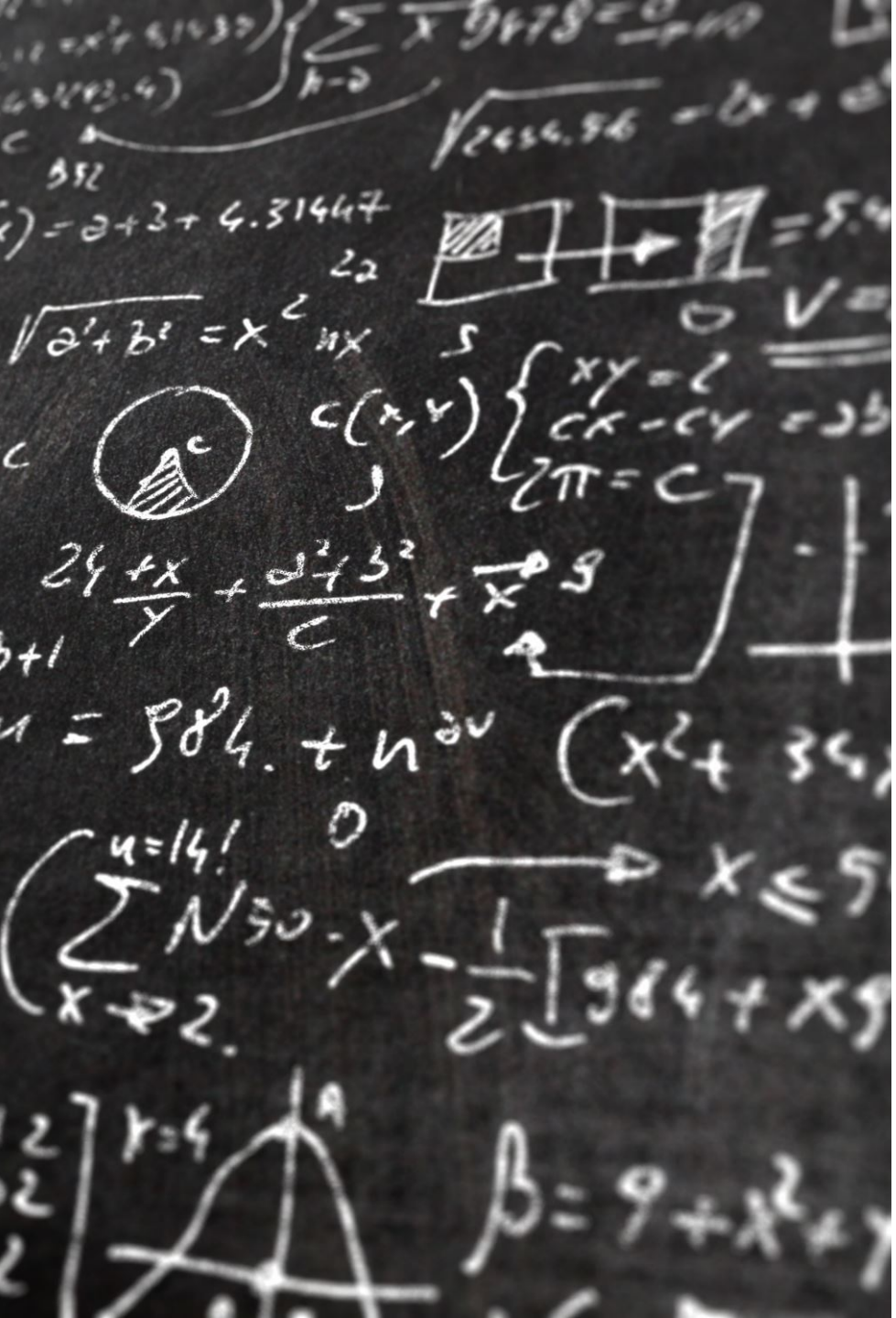
```
string_value = input('How many hours did you work? ')
hours = int(string_value)
```

```
string_value = int(input('How many hours did you work? '))
```

Nested function call

Program 2-13 (input.py)

```
1  # Get the user's name, age, and income.
2  name = input('What is your name? ')
3  age = int(input('What is your age? '))
4  income = float(input('What is your income? '))
5
6  # Display the data.
7  print('Here is the data you entered:')
8  print('Name:', name)
9  print('Age:', age)
10 print('Income:', income)
```



Mathematical Operations

Table 2-3 Python math operators

Symbol	Operation	Description
+	Addition	Adds two numbers
–	Subtraction	Subtracts one number from another
*	Multiplication	Multiplies one number by another
/	Division	Divides one number by another and gives the result as a floating-point number
//	Integer division	Divides one number by another and gives the result as a whole number
%	Remainder	Divides one number by another and gives the remainder
**	Exponent	Raises a number to a power

Order of Math Operations

What is the results of the following expressions:

3+6*2

(3*6)+2

Do we get the same answers?

Write python code to print the two expressions

Operator Precedence

1. Exponentiation: **
2. Multiplication, division, and remainder: * / // %
3. Addition and subtraction: + −

Conditional Execution

The if-statement

```
if condition:
    statement 1
    statement 2
    . . .
    statement N
```

The if-else Statement

```
statements . . .


if condition:
    statement 1a
    statement 2a
    . . .
    statement Na
else:
    statement 1b
    statement 2b
    . . .
    statement Nb
```

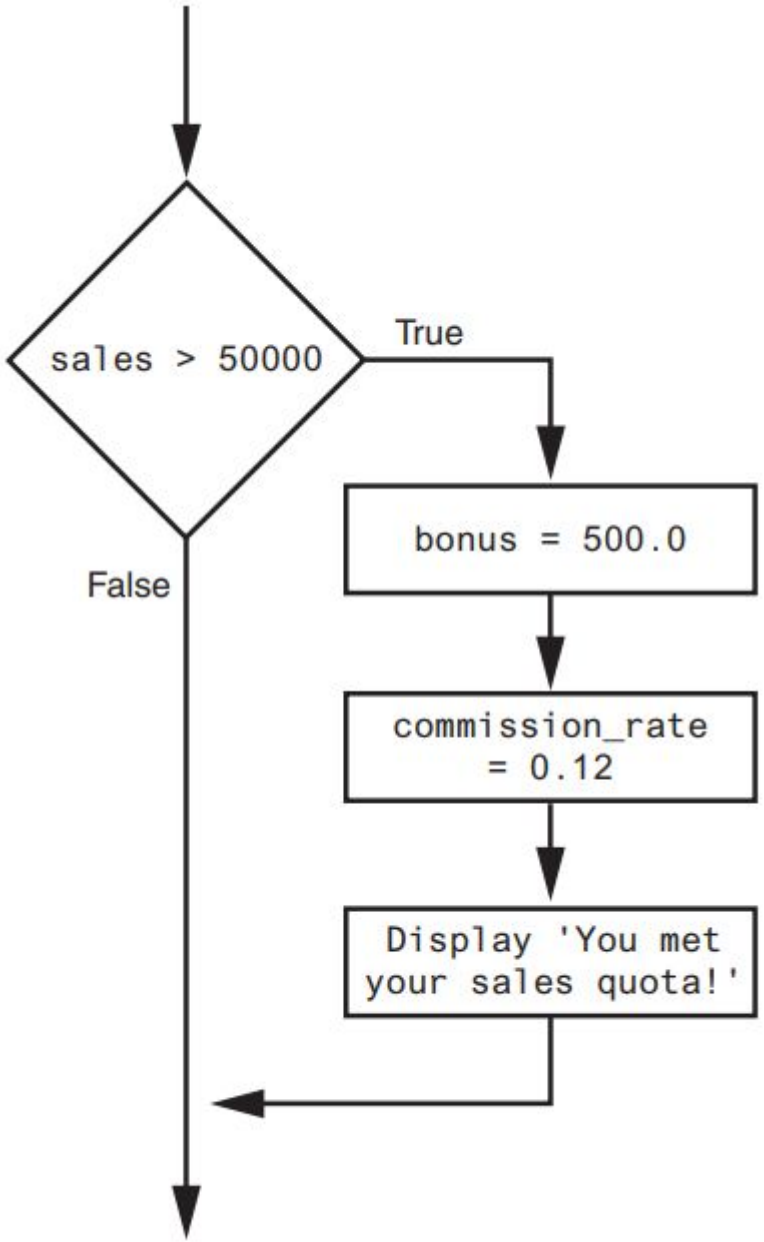
The elif statement

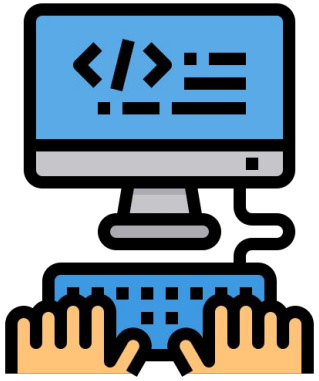
```
statements . . .


if conditionA:
    statements
elif conditionB:
    statements


statements . . .
```

```
statements . . .

if conditionA:
    statements
elif conditionB:
    statements
elif conditionC:
    statements

statements . . .
```

```
statements . . .

if conditionA:
    statements
elif conditionB:
    statements
elif conditionC:
    statements
elif conditionD:
    statements

statements . . .
```

```
statements . . .

if conditionA:
    statements
elif conditionB:
    statements
elif conditionC:
    statements
else:
    statements

statements . . .
```

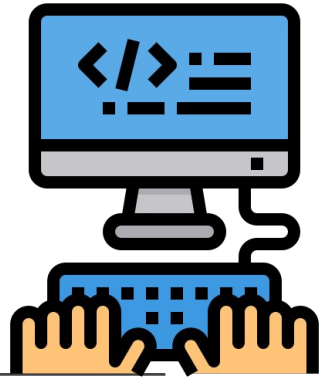




Try this:

```
if sales > 50000:
    bonus = 500.0
    commission_rate = 0.12
    print('You met your sales quota!')
```

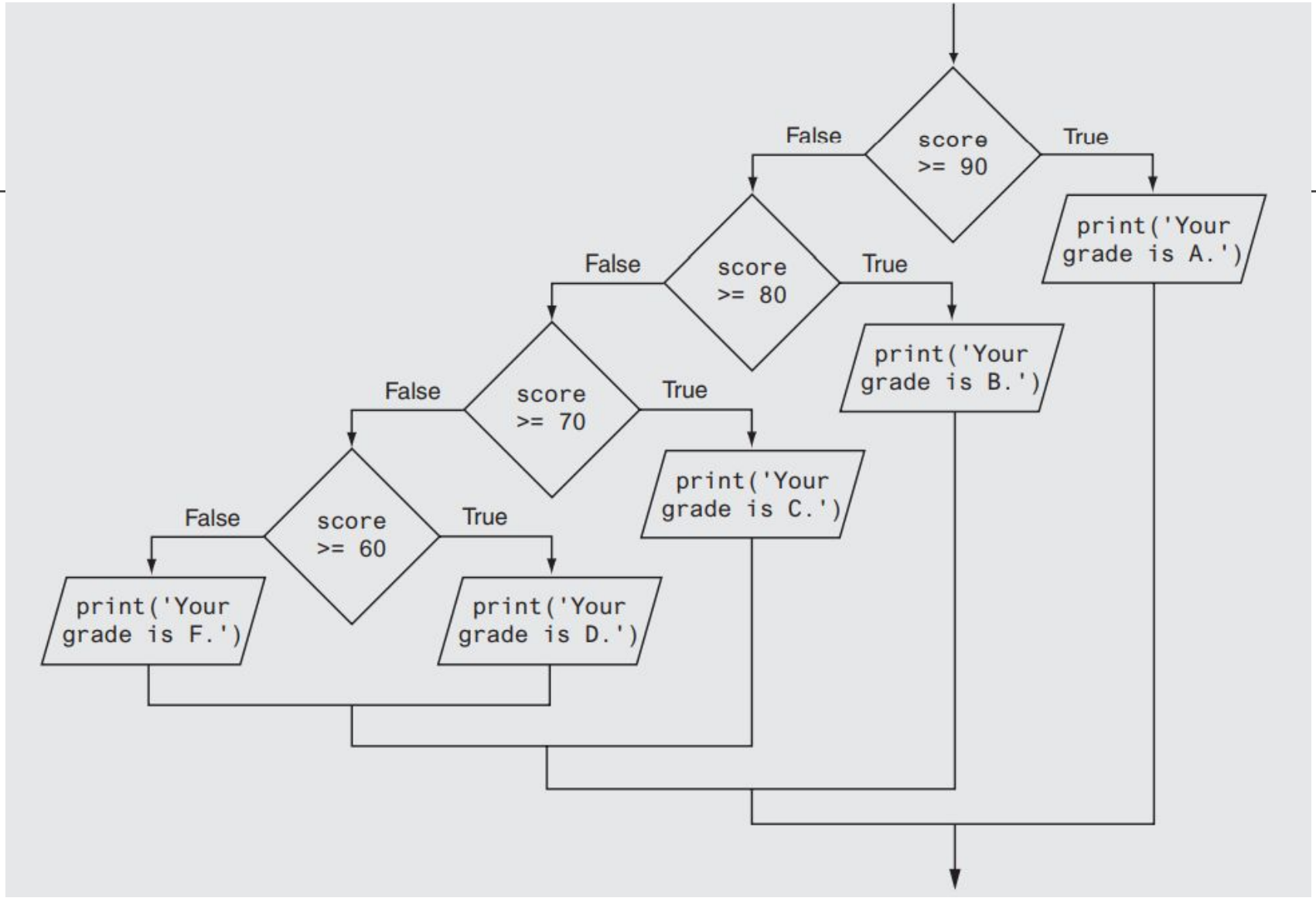
Try-it-yourself

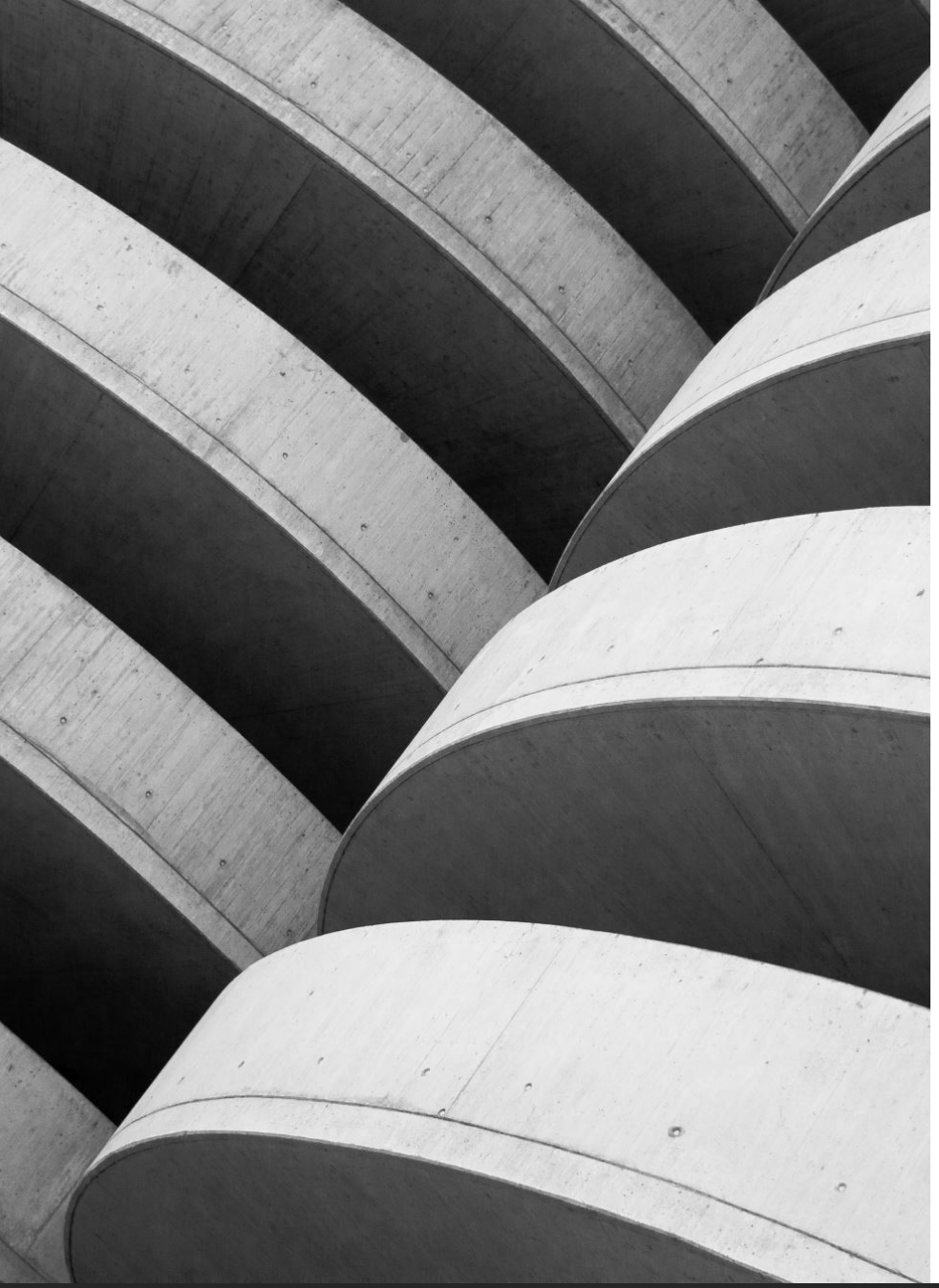


- Accepts a number as input
- Prints out the letter grade that you will receive

```
What is your score? 97
Your grade is A

What is your score? 72
Your grade is C

What is your score? 50
Your grade is F
```

Grade	Score Range
A	90-100
B	80-89
C	70-79
D	60-69
F	below 60


score
>= 90
False
True
print('Your
grade is A.')
score
>= 80
False
True
print('Your
grade is B.')
score
>= 70
False
True
print('Your
grade is C.')
score
>= 60
False
True
print('Your
grade is F.')
print('Your
grade is D.')



Repetition Structure

A REPETITION STRUCTURE CAUSES A STATEMENT OR SET OF STATEMENTS TO EXECUTE REPEATEDLY.

while loop: syntax

```
statements . . .

while conditionA:
    statementA
    statementB
    . . .
    statementN

statements . . .
```

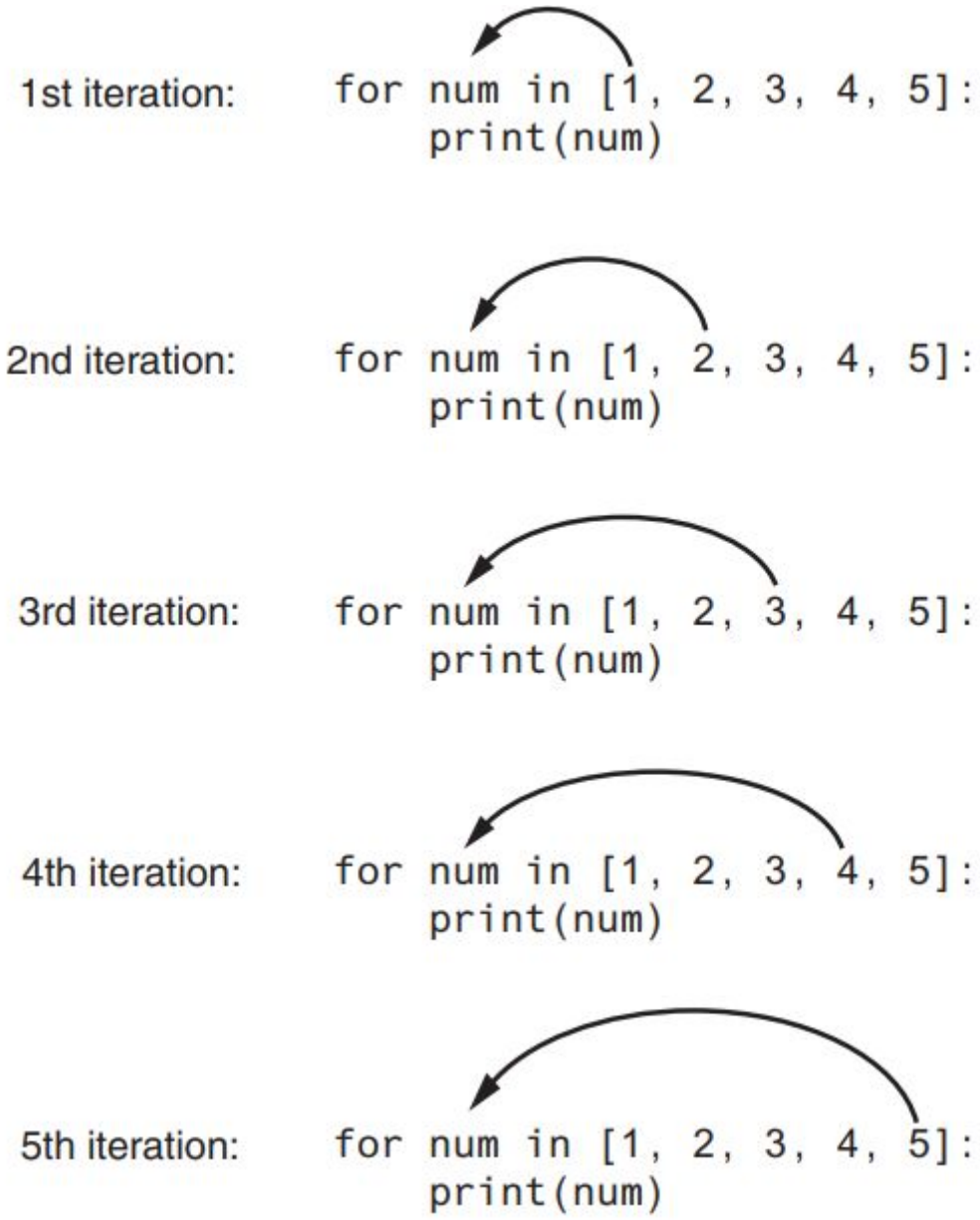
Index variable

```
index = 0

while index < 10:
    print('Print', index)
    # Can add other lines here too
    index = index + 1
```

for loop: syntax

```
statements . . .

for variable in [v1, v2, ...]:
    statementA
    statementB
    . . .
    statementN

statements . . .
```


```python
print('Example: for loop')

for num in [1, 2, 3, 4, 5]:
  print(num)
```



for vs. while

```python
index = 0

while index < 10:
    print('Print', index)
    index = index + 1
```

```python
for index in range(10):
    print('Print', index)
```

Functions

A FUNCTION IS A GROUP OF STATEMENTS THAT EXIST WITHIN A PROGRAM FOR THE PURPOSE OF PERFORMING A SPECIFIC TASK

Use Functions to Divide a Large Task

This program is one long, complex sequence of statements.

```
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
statement
```

In this program the task has been divided into smaller tasks, each of which is performed by a separate function.

```
def function1():
    statement
    statement
    statement
```

function

```
def function2():
    statement
    statement
    statement
```

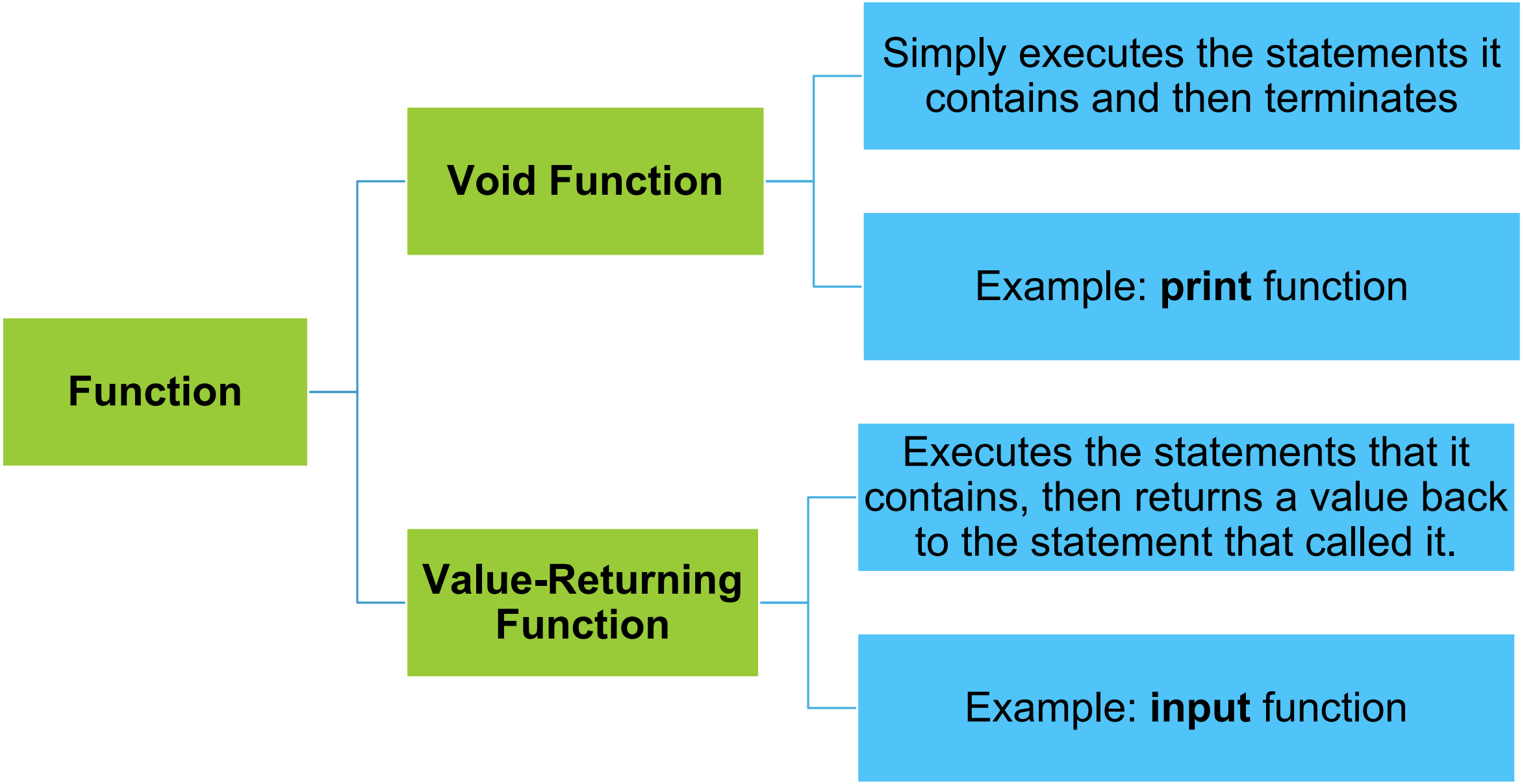
function

```
def function3():
    statement
    statement
    statement
```

function

```
def function4():
    statement
    statement
    statement
```

function


Function
Void Function
Simply executes the statements it contains and then terminates
Example: **print** function
Value-Returning Function
Executes the statements that it contains, then returns a value back to the statement that called it.
Example: **input** function

Void Function

Syntax

```
def function_name():
    statementA
    statementB
    . . .
    statementN
```

Function definition

```
statements . . .
function_name()
statements . . .
```

Function call

Greetings

```
#First, we define a function named greetings.
def greetings():
    print("Good day!")
    print("Welcome to our Program.")
    print("My name is Appy.")
    print("**********************")

#Call the function greetings.
greetings()
```

Function Definition & Function Block

Function call

What is the output of this program?

```
#First, we define a function named greetings.
def greetings():
    print("Good day!")
    print("Welcome to our Program.")
    print("My name is Appy.")
    print("**********************")

#Call the function greetings.
for i in range(3):
    greetings()
```

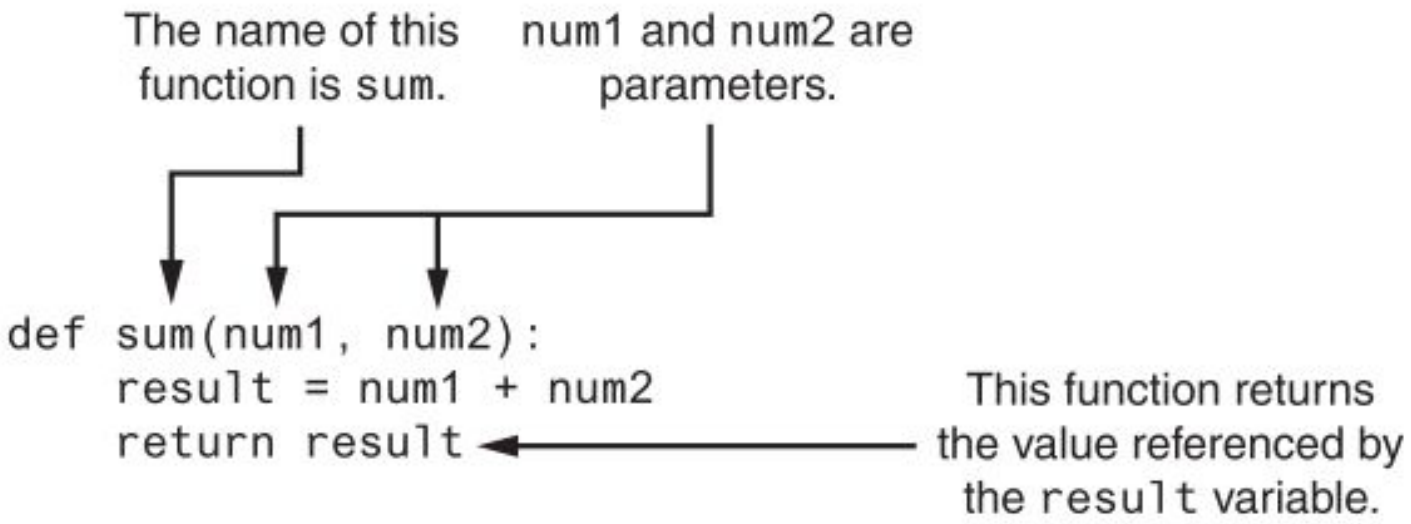
How many times print statements are called in total?

Syntax: Value-Returning Function

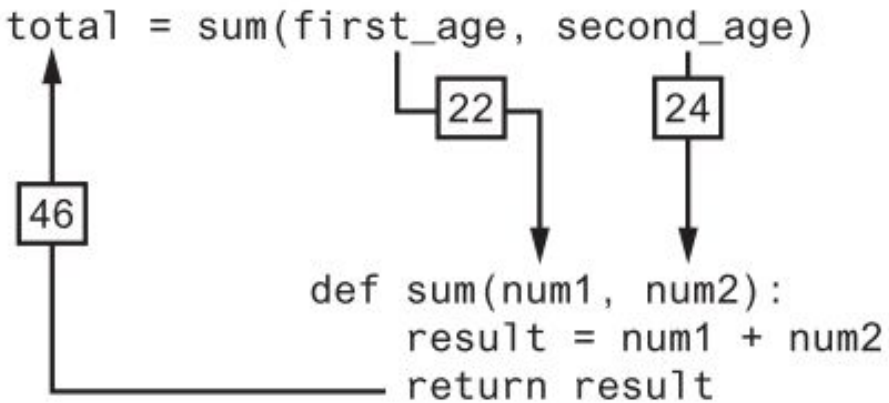
```
def function_name():
    statementA
    statementB
    . . .
    return expression
```

- The value of the expression that follows the keyword return will be sent back to the part of the program that called the function.
- This can be any value, variable, or expression that has a value (such as a math expression).

Example



Arguments are passed to the sum function and a value is returned




Questions?
HSM
ELON