Appendix A Python Quick Reference

Chapter 2: Writing Simple Programs

Reserved Words

False
None
True
and
as
assert
break
class
continue
def
del
elif
else
except
finally
for
from
global
if
import
in
is
lambda
nonlocal
not
or
pass
raise
return
try
while
with
yield

Built-in Functions

abs()
all()
any()
ascii()
bin()
bool()
bytearray()
bytes()
callable()
chr()
classmethod()
compile()
complex()
delattr()
dict()
dir()
divmod()
enumerate()
eval()
exec()
filter()
float()
format()
frozenset()
getattr()
globals()
hasattr()
hash()
help()
hex()
id()
input()
int()
isinstance()
issubclass()
iter()
len()
list()
locals()
map()
max()
memoryview()
min()
next()
object()
oct()
open()
ord()
pow()
print()
property()
range()
repr()
reversed()
round()
set()
setattr()
slice()
sorted()
staticmethod()
str()
sum()
super()
tuple()
type()
vars()
zip()
__import__()

Print Function

```
print(<expr>, <expr>, ..., <expr>)
print()
print(<expr>, <expr>, ..., <expr>, end="\n")
```

```
<variable> = <expr>
<variable1>, <variable2>, ..., <variableN> = <expr1>,<expr2>, ..., <exprN>
```

Input (numeric)

```
<variable> = eval(input(<prompt>))
<variable1>, <variable2>, ..., <variableN> = eval(input(<prompt>))
```

Definite Loop

```
for <var> in <sequence>:
    <body>
```

Chapter 3: Computing with Numbers

Numeric Operators

operator	operation
+	addition
−	subtraction
*	multiplication
/	float division
**	exponentiation
abs()	absolute value
//	integer division
%	remainder

Module Import

```
import <module_name>
```

Math Library Functions

Python	mathematics	English
`pi`	π	An approximation of pi.
`e`	e	An approximation of e.
`sqrt(x)`	$\sqrt{x}$	The square root of x.
`sin(x)`	$\sin x$	The sine of x.
`cos(x)`	$\cos x$	The cosine of x.
`tan(x)`	$\tan x$	The tangent of x.
`asin(x)`	$\arcsin x$	The inverse of sine x.
`acos(x)`	$\arccos x$	The inverse of cosine x.
`atan(x)`	$\arctan x$	The inverse of tangent x.
`log(x)`	$\ln x$	The natural (base e) logarithm of x
`log10(x)`	$\log_{10} x$	The common (base 10) logarithm of x.
`exp(x)`	e^x	The exponential of x.
`ceil(x)`	$\lceil x \rceil$	The smallest whole number >= x.
`floor(x)`	$\lfloor x \rfloor$	The largest whole number <= x.

Built-in Functions

function	description
`range(stop)`	Returns list of ints from `0` to `stop-1`.
`range(start, stop)`	Returns list of ints from `start` to `stop-1`.
`range(start, stop, step)`	Returns a list of ints from `start` to `stop` counting by `step`.
`type(x)`	Returns the Python data type of x.
`int(x)`	Returns value of x converted to int. x may be either numeric or string.
`float(x)`	Returns value of x converted to a float. x may be either numeric or string.
`round(x)`	Returns nearest whole value of x (as a float).

Chapter 4: Objects and Graphics

Direct Import from Module

```
from <module> import <name1>, <name2>, ...
from <module> import *
```

Object Constructor

```
<class-name>(<param1>, <param2>, ... )
```

Object Method Call

```
<object>.<method-name>(<param1>, <param2>, ... )
```

For a summary of the objects and methods contained in the `graphics` module included with this book, see Section 4.8.

Chapter 5: Sequences: Strings, Lists, and Files

Input (strings)

```
<variable> = input(<prompt>)
```

Sequence Operations (strings and lists)

operator	meaning
`<sequence>+<sequence>`	Returns concatenation of sequences. Sequences must be of same type.
`<sequence>*<n>`	Returns sequence concatenated with itself n times. n must be int.
`<sequence>[<n>]`	Returns item at n (0 based from left). n must be int.
`<sequence>[<n>]` where $n < 0$	Returns item at n (1 based from right). n must be int.
`len(<sequence>)`	Returns the length of the sequence.
`<sequence>[<start>:<end> ]`	Returns subsequence from start up to (not including) end.
`for <var> in <sequence>:`	Iterates through items in sequence.

String Methods

function	meaning
`s.capitalize()`	Copy of `s` with only the first character capitalized.
`s.center(width)`	Copy of `s` centered in a field of given `width`.
`s.count(sub)`	Count the number of occurrences of `sub` in `s`.
`s.find(sub)`	Find the first position where `sub` occurs in `s`.
`s.join(list)`	Concatenate `list` into a string, using `s` as separator.
`s.ljust(width)`	Like `center`, but `s` is left-justified.
`s.lower()`	Copy of `s` in all lowercase characters.
`s.lstrip()`	Copy of `s` with leading white space removed.
`s.replace(oldsub,newsub)`	Replace all occurrences of `oldsub` in `s` with `newsub`.
`s.rfind(sub)`	Like `find`, but returns the rightmost position.
`s.rjust(width)`	Like `center`, but `s` is right-justified.
`s.rstrip()`	Copy of `s` with trailing white space removed.
`s.split()`	Split `s` into a list of substrings (see text).
`s.title()`	Copy of `s` with first character of each word capitalized.
`s.upper()`	Copy of `s` with all characters converted to upper case.

Appending to a List

```
<list>.append(<item>)
```

Type Conversion Functions

function	meaning
`float(<expr>)`	Converts `expr` to a floating-point value.
`int(<expr>)`	Converts `expr` to an integer value.
`str(<expr>)`	Returns a string representation of `expr`.
`eval(<string>)`	Evaluates `string` as an expression.

String Formatting

Expression Syntax

```
<template-string>.format(<value0>, <value1>, <value2>, ...)
```

Specifier Syntax

```
{<index>}
{<index>:<width>}
{<index>:<width>.<precision>}
{<index>:<width>.<places>f}
```

Notes:

- The last form is for a fixed number of decimal places.
- Width of 0 means use whatever space is required.
- Width with leading 0 means pad as necessary with 0 (space is default).
- Width may be preceded by < for left-justify, > for right-justify, or ^ for center.

File Processing

Opening and Closing Files

`<variable> = open(<name>, <mode>)`
Mode is `"r"` for reading, `"w"` for writing, `"a"` for appending.

`<fileobj>.close()`

Reading a File

`<file>.read()` Returns the entire remaining contents of the file as a single (potentially large, multi-line) string.

`<file>.readline()` Returns the next line of the file. That is, all text up to *and including* the next newline character.

`<file>.readlines()` Returns a list of the remaining lines in the file. Each list item is a single line including the newline character at the end.

Note: The file object may also be used in a `for` loop where it is treated as a sequence of lines.

Writing to a File

```
print(..., file=<outputFile>)
```

Chapter 6: Defining Functions

Function Definition

```
def <name>(<formal-param1>, <formal-param2>, ... )
    <body>
```

Function Call

```
<name>(<actual-param1>, <actual-param2>, ... )
```

Return Statement

```
return <value1>, <value2>, ...
```

Chapter 7: Decision Structures

Simple Conditions

```
<expr><relop><expr>
```

Relational Operators

Python	mathematics	meaning
<	$<$	less than
<=	$\leq$	less than or equal to
==	$=$	equal to
>=	$\geq$	greater than or equal to
>	$>$	greater than
!=	$\neq$	not equal to

Note: These operators return a bool value (`True`/`False`).

If Statements

```
if <condition>:
    <statements>

if <condition>:
    <statements1>
else:
    <statements2>

if <condition1>:
    <case1 statements>
elif <condition2>:
    <case2 statements>
...
else:
    <default statements>
```

Note: `else` clause is optional in `elif` form.

Preventing Execution on Import

```
if __name__ == "__main__":
    main()
```

Exception Handling

```
try:
    <statements>
except <ExceptionType>:
    <handler1>
except <ExceptionType>:
    <handler2>
...d
except:
    <default handler>
```

Chapter 8: Loop Structures and Booleans

For Loop

```
for <var> in <sequence>:
    <body>
```

While Loop

```
while <condition>:
    <body>
```

Break Statement

```
while True:
    ...
    if <cond>: break
    ....
```

Boolean Expressions

Literals: `True`, `False`
Operators: `and`, `or`, `not`

operator	operational definition
x and y	If x is false, return x. Otherwise, return y.
x or y	If x is true, return x. Otherwise, return y.
not x	If x is false, return `True`. Otherwise, return `False`.

Type conversion function: `bool`

Chapter 9: Simulation and Design

Random Library

`random()` Returns a uniformly distributed pseudo-random value in the range [0,1).

`randrange(<params>)` Returns a uniformly distributed pseudo-random from `range(<params>)`.

Chapter 10: Defining Classes

Class Definition

```
class <class-name>:
    <method-definitions>
```

Notes:

- A method definition is a function with a special first parameter, `self`, that refers to the object to which the method is being applied.

- The constructor is a method named `__init__`.

Documentation Strings

A string at the beginning of a module, class, function, or method can be used for documentation. Docstrings are carried along at runtime and are used for interactive help and the pydoc utility.

Chapter 11: Data Collections

Sequence Operations (Lists and Strings)

operator	meaning
`<seq> + <seq>`	concatenation
`<seq> * <int-expr>`	repetition
`<seq>[ ]`	indexing
`len(<seq>)`	length
`<seq>[:]`	slicing
`for <var> in <seq>:`	iteration
`<expr> in <seq>`	membership check (returns a Boolean)

List Methods

method	meaning
`<list>.append(x)`	Adds element x to end of list.
`<list>.sort()`	Sorts (orders) the list. Keyword parameters: `key`, `reverse`.
`<list>.reverse()`	Reverses the list.
`<list>.index(x)`	Returns index of first occurrence of x.
`<list>.insert(i,x)`	Inserts x into list at index i.
`<list>.count(x)`	Returns the number of occurrences of x in list.
`<list>.remove(x)`	Deletes the first occurrence of x in list.
`<list>.pop(i)`	Deletes the ith element of the list and returns its value.

Dictionaries

Dictionary Literal: `{<key1>:<value1>, <key2>:<value2>, ...}`

method	meaning
`<key> in <dict>`	Returns true if dictionary contains the specified key, false if it doesn't.
`<dict>.keys()`	Returns a sequence keys.
`<dict>.values()`	Returns a sequence of values.
`<dict>.items()`	Returns a sequence of tuples `(key,value)` representing the key-value pairs.
`<dict>.get(<key>, <default>)`	If dictionary has `key` returns its value; otherwise returns `default`.
`del <dict>[<key>]`	Deletes the specified entry.
`<dict>.clear()`	Deletes all entries.
`for <var> in <dict>:`	Loops over the keys.

Appendix B Glossary

abstraction The purposeful hiding or ignoring of some details in order to concentrate on those that are relevant.

accessor method A method that returns the value of one or more of an object's instance variable(s), but does not modify the object.

accumulator pattern A common programming pattern in which a final answer is built a piece at a time in a loop.

accumulator variable A variable that is used to hold the result in the accumulator programming pattern.

actual parameter A value that is passed to a function when it is called.

algorithm A detailed sequence of steps for carrying out some process. A recipe.

aliasing The situation in which two or more variables refer to exactly the same object. If the object is mutable, then changes made through one variable will be seen by the others.

analysis 1) In the context of the software development lifecycle, this refers to the process of studying a problem and figuring out what a computer program might do to solve it. 2) Studying a problem or algorithm mathematically to determine some of its properties, such as time efficiency.

and A binary Boolean operator that returns true when both of its subexpressions are true.

application programming interface (API) A specification of the functionality provided by a library module. A programmer needs to understand the API to be able to use a module.

argument An actual parameter.

array A collection of similar objects that can be accessed through indexing. Usually arrays are fixed-sized and homogeneous (all elements are of the same type). *Compare to* list.

ASCII American Standard Code for Information Interchange. A standard for encoding text where each character is represented by a number 0–127.

assignment The process of giving a value to a variable.

associative array A collection where values are associated with keys. Called a *dictionary* in Python.

attributes The instance variables and methods of an object.

base case In a recursive function or definition, a situation in which recursion is not required. All proper recursions must have one or more base cases.

batch A mode of processing in which input and output is done through files rather than interactively.

binary The base-2 numbering system in which the only digits are 0 and 1.

binary search A very efficient searching algorithm for finding items in a sorted collection. Requires time proportional to $\log_2 n$ where n is the size of the collection.

bit Binary digit. A fundamental unit of information. Usually represented using 0 and 1.

body A generic term for the block of statements inside a control structure such as a loop or decision.

Boolean algebra The rules that govern simplification and rewriting of Boolean expressions.

Boolean expression A truth statement. A Boolean expression evaluates to either true or false.

Boolean logic *See* Boolean algebra.

Boolean operations Connectives for constructing Boolean expressions. In Python, `and`, `or`, and `not`.

bug An error in a program.

butterfly effect A classic example of dynamical systems in nature (chaos). Supposedly, an event as small as the flapping of a butterfly's wing can significantly influence subsequent large-scale weather patterns.

byte code An intermediate form of computer language. High-level languages are sometimes compiled into byte code, which is then interpreted. In Python, files with a `pyc` extension are byte code.

call The process of invoking a function's definition.

central processing unit (CPU) The "brain" of the computer where numeric and logical operations are carried out.

cipher alphabet The symbols that are used to encrypt a message.

ciphertext The encrypted form of a message.

class A class describes a set of related objects. The `class` mechanism in Python is used as a "factory" to produce objects.

client In programming, a module that interfaces with another component is called a client for the component.

code injection A form of computer attack in which a malicious user introduces computer instructions into an executing program, causing the application to deviate from its original design.

coding The process of turning an algorithm into a computer program.

comment Text placed in a program for the benefit of human readers. Comments are ignored by the computer.

compiler A complex program that translates a program written in a high-level language into the machine language that can be executed by a particular computer.

computer A machine that stores and manipulates information under the control of a changeable program.

computer science The study of what can be computed.

conditional Another term for a decision control structure.

constructor A function that creates a new object. In a Python class, it is the `__init__` method.

control codes Special characters that do not print, but are used in the interchange of information.

control structure A programming language statement that controls the execution of other statements (e.g., `if` and `while`).

coordinate transformation In graphical programming, the mathematics of changing a point or set of points from one coordinate system to a related one.

counted loop A loop written to iterate a specific number of times.

CPU *See* central processing unit.

cryptography The study of techniques for encoding information to keep it secure.

data The information that a computer program manipulates.

data type A particular way of representing data. The data type of an item determines what values it can have and what operations it supports.

debugging The process of finding and eliminating errors in a program.

decision structure A control structure that allows different parts of a program to execute depending on the exact situation. Usually decisions are controlled by Boolean expressions.

decision tree A complex decision structure in which an initial decision branches into more decisions, which branch into more decisions in a cascading fashion.

definite loop A kind of loop where the number of iterations is known at the time the loop begins executing.

design The process of developing a system that can solve some problem. Also the product of that process.

dictionary An unordered Python collection object that allows values to be associated with arbitrary keys.

docstring A documentation technique in Python that associates a string with a program component.

empty string An object that has the data type string, but contains no characters ("").

encapsulation Hiding the details of something. Usually this is the term used to describe the distinction between the implementation and use of an object or function. Details are encapsulated in the definition.

encryption The process of encoding information to keep it private.

end-of-file loop A programming pattern used to read a file line by line.

event In GUI programming, an outside action such a mouse click that causes something to happen in a program. Also used to describe the object that is created to encapsulate the information about the event.

event-driven A style of programming in which the program waits for events to happen and responds accordingly. This approach is frequently used in graphical user interface (GUI) programming.

exception handling A programming language mechanism that allows the programmer to gracefully deal with errors that are detected when a program is running.

execute To run a program or segment of a program.

exponential time An algorithm that requires a number of steps proportional to a function in which the problem size appears as an exponent. Such algorithms are generally considered intractable.

expression A part of a program that produces data.

fetch-execute cycle The process a computer carries out to execute a machine code program.

float A data type for representing numbers with fractional values. Short for "floating point."

flowchart A graphical depiction of the flow of control in a program or algorithm.

function A subprogram within a program. Functions take parameters as input and can return values.

functional decomposition *See* top-down design.

garbage collection A process carried out by dynamic programming languages (e.g., Python, Lisp, Java) in which memory locations that contain values that are no longer in use are freed up so that they can store new values.

graphical user interface (GUI) A style of interaction with a computer application that involves heavy use of graphical components such as windows, menus, and buttons.

graphics window An on-screen window where graphics can be drawn.

GUI *See* graphical user interface.

halting problem A famous unsolvable problem. A program that determines if another program will halt on a given input.

hardware The physical components of a computing system. If it goes "crash" when you toss it out the window, then it's hardware.

hash Another term for *associative array* or *dictionary*.

hello, world The ubiquitous first computer program.

heterogeneous Capable of containing more than a single data type at one time. Python lists, for example.

homogeneous Capable of holding values of only a single type.

identifiers The names that are given to program entities.

if statement A control structure for implementing decisions in a program.

import statement A statement that makes an external library module available for use within a program.

indefinite loop A loop for which the number of iterations required is not necessarily known at the time the loop begins to execute.

indexing Selecting a single item from a sequence based on its relative position in the sequence.

infinite loop A loop that does not terminate. *See* loop, infinite.

inheritance Defining a new class as a specialization of another class.

input, process, output A common programming pattern. The program prompts for input, processes it, and outputs a response.

input validation The process of checking the values supplied by a user to make sure that they are legitimate before performing a computation with those values.

instance A particular object of some class.

instance variable A piece of data stored inside an object.

int A data type for representing numbers with no fractional component. Int is short for *integer* and represents a number with a fixed number of bits (commonly 32).

integer A positive or negative whole number. *See* int.

interactive loop A loop that allows part of a program to repeat according to the wishes of the user.

interface The connection between two components. For a function or method, the interface consists of the name of the function, its parameters and return values. For an object, it is the set of methods (and their interfaces) that are used to manipulate the object. The term *user interface* is used to describe how a person interacts with a computer application.

interpreter A computer program that simulates the behavior of a computer that understands a high-level language. It executes the lines of source one by one and carries out the operations.

intractable Too difficult to be solved in practice, usually because it would take too long.

invoke To make use of a function.

iterate To do multiple times. Each execution of a loop body is called an iteration.

key 1) In encryption, a special value that must be known to either encode or decode a message. 2) In the context of data collections, a way to look up a value in a dictionary. Values are associated with keys for future access.

lexicographic Having to do with string ordering. Lexicographic order is like alphabetical order, but based on the underlying numeric codes of the string's characters.

library An external collection of useful functions or classes that can be imported and used in a program. For example, the Python `math` and `string` modules.

linear search A search process that examines items in a collection sequentially.

linear time algorithm An algorithm that requires a number of steps proportional to the size of the input problem.

list A general Python data type for representing sequential collections. Lists are heterogeneous and can grow and shrink as needed. Items are accessed through subscripting.

literal A notation for writing a specific value in a programming language. For example, `3` is an int literal and `"Hello"` is a string literal.

local variable A variable defined inside a function. It may only be referred to within the function definition. *See* scope.

log time algorithm An algorithm that requires a number of steps proportional to the log of the size of the input problem.

loop and a half A loop structure that has an exit somewhere in the midst of the loop body. In Python this is accomplished via a `while True:/break` combination.

loop A control construct for executing portions of a program multiple times.

loop index A variable that is used to control a loop. In the statement: `for i in range(n)`, `i` is being used as a loop index.

loop, infinite *See* infinite loop.

machine code A program in machine language.

machine language The low-level (binary) instructions that a given CPU can execute.

main memory The place where all data and program instructions that the CPU is currently working on resides. Also known as random access memory (RAM).

mapping A general association between keys and values. Python dictionaries implement a mapping.

merge The process of combining two sorted lists into a single sorted list.

merge sort An efficient divide-and-conquer sorting algorithm.

meta-language A notation used to describe the syntax of a computer language.

method A function that lives inside an object. Objects are manipulated by calling their methods.

mixed-typed expression An expression involving more than one data type. Usually used in the context of combining ints and floats in numeric computations.

model-view architecture Dividing up a GUI program by separating the problem (model) from the user interface (view).

modal A window or dialog box is modal if it requires the user to interact with it in some way before continuing to use the application that generated it.

modular Consisting of multiple, relatively independent pieces that work together.

module Generally, any relatively independent part of a program. In Python, the term is also used to mean a file containing code that can be imported and executed.

module hierarchy chart A diagram showing the functional decomposition structure of a program. A line between two components shows that the one above uses the one below to accomplish its task.

Monte Carlo A simulation technique that involves probabilistic (random or pseudo-random) elements.

mutable Changeable. An object whose state can be changed is said to be mutable. Python ints and strings are not mutable, but lists are.

mutator method A method that changes the state of an object (i.e., modifies one or more of the instance variables).

n log n algorithm An algorithm that requires a number of steps that is proportional to the size of the input times the log of the size of the input.

n-squared algorithm An algorithm that requires a number of steps that is proportional to the square of the size of the input.

name error An exception that occurs when Python is asked to produce a value for a variable that has not been assigned a value.

namespace An association between identifiers and the things that they represent in a program. In Python, modules, classes, and objects act as namespaces.

nesting The process of placing one control structure inside of another. Loops and decisions may be arbitrarily nested.

newline A special character that marks the division between lines in a file or a multi-line string. In Python, it is denoted `"\n"`.

not A unary Boolean operator to negate an expression.

object A program entity that has some data and a set of operations to manipulate that data.

object-based Design and programming that uses objects as the principle form of abstraction.

object-oriented Object-based design or programming that includes characteristics of polymorphism and inheritance.

open The process of associating a file in secondary memory with a variable in a program through which the file can be manipulated.

operator A function for combining expressions into more complex expressions.

or A binary Boolean operator that returns true when either or both subexpressions are true.

override The term applied to a situation when a subclass changes the behavior of an inherited method.

parameters Special variables in a function that are initialized at the time of call with information passed from the caller.

pass by value A parameter-passing technique used in Python. The formal parameters are assigned the values from the actual parameters. The function cannot change which object an actual parameter variable refers to.

pass by reference A parameter-passing technique used in some computer languages that allows the value of a variable used as an actual parameter to be changed by the called function.

pixel Short for *picture element*. A single dot on a graphical display.

plaintext In encryption, this is the term used for an unencoded message.

polymorphism Literally "many forms." In object-oriented programming, the ability of a particular line of code to be implemented by different methods depending on the data type of the object involved.

portability The ability to run a program unmodified on various different systems.

post-test loop A loop construct where the loop condition is not tested until after the loop body has been executed.

pre-test loop A loop construct where the loop condition is tested before executing the body of the loop.

precision The number of digits of accuracy in a number.

priming read In a sentinel loop, a read before the loop condition is tested.

private key A kind of encryption where the same key is used to both encrypt and decrypt and must therefore be kept secret.

program A detailed set of instructions for a computer to carry out.

programming The process of creating a computer program to solve some problem.

programming environment A special computer program that provides facilities to make programming easier. IDLE (in the standard Python distribution) is an example of a simple programming environment.

programming language A notation for writing computer programs. Usually used to refer to high-level languages such as Python, Java, C++, etc.

prompt A printed message that signals to the user of a program that input is expected.

prototype An initial simplified version of a program.

pseudocode A notation for writing algorithms using precise natural language, instead of a computer language.

pseudo-random Sequences of numbers generated by computer algorithms and used to simulate random events.

public key A form of encryption that uses two different keys. A message encoded with a public key can only be decoded using a separate private key.

random access memory (RAM) *See* main memory.

random walk A simulation process in which movement of some object is determined probabilistically.

read A term used to describe computer input. A program is said to read information from the keyboard or a file.

record A collection of information about a single individual or object. For example, a personnel record contains information about an employee.

recursive Having a quality (for a function or definition) of referring to itself. *See* recursive.

recursive function A function that calls itself, either directly or indirectly.

relational operator An operator that makes a comparison between values and returns true or false (e.g., <, <=, ==, >=, >, ! =).

reserved words Identifiers that are part of the built-in syntax of a language.

resolution The number of pixels on a graphics screen. Usually expressed as horizontal by vertical (e.g., 640x480).

RGB value A representation of a color as three numbers (typically in the range 0–255) that represent the brightness of the red, green, and blue color components of a pixel.

scope The area of a program where a given variable may be referenced. For example, variables defined in functions are said to have local scope.

script Another name for a program. Usually used to refer to a relatively simple program written in an interpreted language.

search The process of finding a particular item in a collection.

secondary memory A generic term referring to nonvolatile storage devices such as hard disks, floppy disks, magentic tapes, CD-ROMs, DVDs, etc.

seed The value used to start generation of a pseudorandom sequence.

selection sort An n-squared-time sorting algorithm.

self parameter In Python, the first parameter of a method. It is a reference to the object to which the method is being applied.

semantics The meaning of a construct.

sentinel A special value used to signal the end of a series of inputs.

sentinel loop A loop that continues until a special value is encountered.

short-circuit evaluation An evaluation process that returns an answer as soon as the result is known, without necessarily evaluating all of its subexpressions. In the expression `(True or isover())` the `isover()` function will not be called.

signature Another term for the interface of a function. The signature includes the name, parameter(s), and return value(s).

simulation A program designed to abstractly mimic some real-world process.

simultaneous assignment A statement that allows multiple variables to be assigned in a single step. For example, `x,y = y,x` swaps two variables.

slicing Extracting a subsequence of a string, list, or other sequence object.

software Computer programs.

sorting The process of arranging a sequence of items into a pre-determined ordering.

source code The text of a program in a high-level language.

spiral design Creating a system by first designing a simplified prototype and then gradually adding features.

statement A single command in a programming language.

step-wise refinement The process of designing a system by starting with a very high-level, abstract description and gradually adding in details.

string A data type for representing a sequence of characters (text).

structure chart *See* module hierarchy chart.

subclass When one class inherits from another, the inheriting class is called a subclass of the class from which it inherits.

substring A sequence of contiguous characters inside a string. *See* slicing.

superclass A class which is being inherited from.

syntax The form of a language.

tkinter The Standard GUI framework that comes with Python. The `graphics.py` module used in this book is built on this.

top-down design The process of building a system by starting with a very high-level algorithm that describes a solution in terms of subprograms. Each subprogram is then designed in turn. Other names for this process are *step-wise refinement* and *functional decomposition*.

truth table A table showing the value of a Boolean expression for all possible combinations of values of its subexpressions.

tuple A Python sequence type that acts like an immutable list.

unary An operator that acts on a single operand.

unicode An alternative to ASCII that encodes characters from virtually all of the world's written languages. Unicode is designed to be ASCII-compatible.

unit testing Trying out a component of a program independent of other pieces.

unpack In Python, the assignment of items in a sequence into independent variables. For example, a list or tuple of two values can be unpacked into variables like this: `x,y = myList`.

variable An identifier that labels a value for future reference. The value of a variable can be changed through assignment.

widget A user interface component in a GUI.

write The process of outputting information. For example, data is said to be written to a file.