Contents

Introduction

If you have ever picked up a programming manual and felt your forehead go clammy and your eyes cross as you attempt to make sense of the long-winded explanations, this is the guide for you.

I have been in your position, attempting to learn how to program and having to rely on the traditional style of guides. I know from painful experience how quickly I glaze over and my brain solidifies; after only a few pages the tedium leaves me blindly reading words without any real notion of what they mean any more. Inevitably I give up and the whole process makes me feel like a limp failure, gasping for breath after I surface from drowning in technical jargon.

I hated having to read through pointless drivel and then be presented with a short program telling me exactly what to type in and then spend the next 20 pages reading about what I have just done and the 101 ways I could run it. I hated having no control over trying things out for myself and I hated the way these guides would only contain one or two challenges at the end of a chapter of theory.

I knew there had to be a better way, and thankfully there is. I wrote it and you are presently reading it, so aren't you lucky? This guide is refreshingly different and helps you learn how to program with Python by using practical examples rather than self-important explanations.

Many programmers learn through experimentation, looking at others' code and working out what method is best for a given situation. This book is a hands-on approach to learning programming. After minimal reading you are set a number of challenges to create the programs. You can explore and experiment with the programming language and look at the example solutions to learn how to think like a programmer. There are no chapters entitled "the architecture of a computer", "the theory of programming" or any other gobbledy-gook other authors like to waste time with. I don't want to baffle you with theory or blind you with overbearing explanations that suck out your enthusiasm for learning to program.

Hopefully, you want to get stuck into creating programs, solving problems and enjoying the sense of accomplishment that you get as you proudly look over your lines of code, knowing that you created something that works. That is great, your eagerness is to be applauded and I salute those who are reading this while already sitting at their computers, fingers poised and ready to get going. If that is the case, that you already have Python open on your screen and are itching to get going, then away you go and I'll see you in the first chapter called "The Basics" on page 11.

For everyone who is still with us and is feeling a little more timid, there are just a few more things to tell you about before you take the plunge.

How to Use This Book

This book builds from very simple programs to more complex ones. If you are new to programming or new to Python, start with "The Basics" and work through the chapters in order.

If you are familiar with Python programming and feel confident with the basics, the theory and logic surrounding programming, then you can just dip in and out of the book to get help on the specifics you need.

The book is split into two sections:

Part I

In Part I, each chapter takes you through some basic programming rules and challenges for you to complete and includes:

- a simple explanation giving you pointers, which is useful if you are new to programming in Python;
- examples of code with a short explanation, which you can use as a basis to solve the challenges;
- a list of challenges for you to work through that get harder as you move through them. Each challenge should only take between a couple of minutes and 20 minutes to solve; however, some of the more complex challenges near the end of Part I will take longer as you build up the techniques you will be using. Don't panic if you take longer than this, as long as you solve the problems without *too* much copying from the suggested solution, you are doing fine;
- code containing a possible solution for each challenge; there is often more than one answer available, but we include just a single program as a possible solution that you can refer to if you get stuck on a particular aspect of the code.

Part II

In Part II, you are given some larger challenges which utilize the programming skills you learnt in Part I and allow you to consolidate and reinforce the techniques you have been practising. In this section, you are not given the help and example code that is given in Part I and it will take longer to solve each challenge. After each challenge, you are given one possible answer that you may find useful if you are stuck. However, you may have found another solution that works just as well.

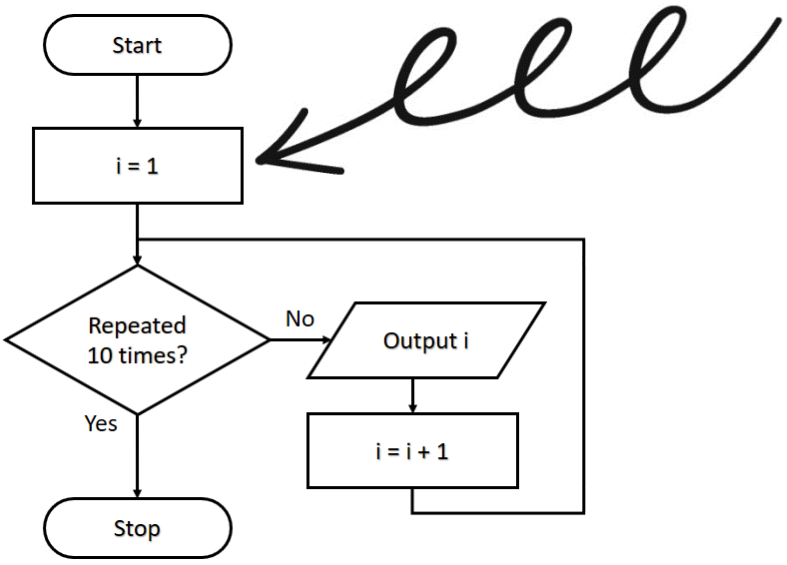
Who Is This Book For?

This book is suitable for anyone who wants to learn how to program with Python. It is an essential tool for teachers and students in Key Stage 3 or those studying computer science who need help and ready-made examples to practise programming techniques and build confidence. It can also be used to help with a computer science programming project resource bank, to help pupils needing additional support or just a quick reminder of the syntax when creating programs.

For Loop

Explanation

A **`for loop`** allows Python to keep repeating code a set number of times. It is sometimes known as a **`counting loop`** because you know the number of times the loop will run before it starts.



In this case, it starts at 1 and will keep repeating the loop (displaying i) until it reaches 10 and then stops. This is how this loop would look in Python

```
for i in range(1,10):
    print(i)
```

In this example, the outputs would be 1, 2, 3, 4, 5, 6, 7, 8 and 9. **When it gets to 10 the loop would stop so 10 would not be shown in the output.**

Remember to indent the lines of code within the for loop.

Example Code

The range function is often used in for loops and lists the starting number, the ending number and can also include the steps (e.g. counting in 1s, 5s or any other value you wish).

```
for i in range(1,10):
 print(i)
```

This loop uses a variable called "i" to keep track of the number of times the loop has been repeated. It will start i at 1 (as that is the starting value in the range function) and repeat the loop, adding 1 to i each time and displaying the value of i until it reaches 10 (as dictated by the range function), where it will stop. Therefore, it will not repeat the loop a tenth time and will only have the following output:

1, 2, 3, 4, 5, 6, 7, 8, 9

```
for i in range(1,10,2):
 print(i)
```

This range function includes a third value which shows how much is added to i in each loop (in this case 2). The output will therefore be: **1, 3, 5, 7, 9**

```
for i in range(10,1,-3):
 print(i)
```

This range will subtract 3 from i each time. The output will be: **10, 7, 4**

Using loops is a powerful programming tool that you will use a lot in the more challenging programs.

```
for i in word:
 print(i)
```

This would display each character in a string called "word" as a separate output (i.e. on a separate line).

Challenges

035
Ask the user to enter their name and then display their name three times.

036
Alter program 035 so that it will ask the user to enter their name and a number and then display their name that number of times.

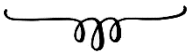
037
Ask the user to enter their name and display each letter in their name on a separate line.

038
Change program 037 to also ask for a number. Display their name (one letter at a time on each line) and repeat this for the number of times they entered.

039
Ask the user to enter a number between 1 and 12 and then display the times table for that number.

040
Ask for a number below 50 and then count down from 50 to that number, making sure you show the number they entered in the output.

041
Ask the user to enter their name and a number. If the number is less than 10, then display their name that number of times; otherwise display the message "Too high" three times.



042
Set a variable called total to 0. Ask the user to enter five numbers and after each input ask them if they want that number included. If they do, then add the number to the total. If they do not want it included, don't add it to the total. After they have entered all five numbers, display the total.

043
Ask which direction the user wants to count (up or down). If they select up, then ask them for the top number and then count from 1 to that number. If they select down, ask them to enter a number below 20 and then count down from 20 to that number. If they entered something other than up or down, display the message "I don't understand".

044
Ask how many people the user wants to invite to a party. If they enter a number below 10, ask for the names and after each name display "[name] has been invited". If they enter a number which is 10 or higher, display the message "Too many people".

Answers

035

```
name = input("Type in your name: ")
for i in range (0,3):
    print(name)
```

036

```
name = input("Type in your name: ")
number = int(input("Enter a number: "))
for i in range (0,number):
    print(name)
```

037

```
name = input("Enter your name: ")
for i in name:
    print(i)
```

038

```
num = int(input("Enter a number: "))
name = input("Enter your name: ")
for x in range(0,num):
    for i in name:
        print(i)
```

039

```
num = int(input("Enter a number between 1 and 12: "))
for i in range(1, 13):
    answer = i * num
    print (i, "x", num, "=", answer)
```

040

```
num = int(input("Enter a number below 50: "))
for i in range(50,num-1, -1):
    print(i)
```

041

```
name = input("Enter your name: ")
num = int(input("Enter a number: "))
if num < 10:
    for i in range(0,num):
        print(name)
else:
    for i in range(0,3):
        print("Too high")
```

042

```
total = 0
for i in range(0,5):
    num = int(input("Enter a number: "))
    ans = input("Do you want this number included? (y/n) ")
    if ans == "y":
        total = total + num
print(total)
```

043

```
direction = input("Do you want to count up or down? (u/d) ")
if direction == "u":
    num = int(input("What is the top number? "))
    for i in range(1,num+1):
        print(i)
elif direction == "d":
    num = int(input("Enter a number below 20: "))
    for i in range(20,num-1, -1):
        print(i)
else:
    print("I don't understand")
```

044

```
num = int(input("How many friends do you want to invite to the party? "))
if num < 10:
    for i in range(0,num):
        name = input("Enter a name: ")
        print(name, "has been invited")
else:
    print("Too many people")
```

End of sample