TECHNITEA BOOKS · DIGITAL SKILLS

Code Without Fear

Python for complete beginners. Write your first real programs in a weekend, one small step at a time.



Technitea

technitea.com

Code Without Fear

Python for complete beginners. Write your first real programs in a weekend, one small step at a time.

Code Without Fear
Python for complete beginners. Write your first real programs in a weekend, one small step at a time.
First edition, November 2026

Written and edited by the Technitea editorial team.

Published by Technitea.
PT Arena Karya Nusantara, Tuban, East Java, Indonesia.
technitea.com

© 2026 PT Arena Karya Nusantara. All rights reserved. No part of this book may be reproduced, shared or resold in any form without written permission from the publisher, except for brief quotations in reviews. Your purchase licenses one reader. The worksheets may be printed for your own use.

Cover illustration generated with AI image tools under art direction. Every program in this book was run and checked before publication.

Every program in this book was tested with Python 3.13 in November 2026 and works with Python 3.10 or later. The publisher accepts no liability for losses arising from the use of this book.

Contents

	Before you start
01	Say hello to Python
02	Boxes with names: values and variables
03	Ask a question, use the answer
04	Let the program decide
05	Do it again: loops
06	Lists: many things in one place
07	Functions: name your own commands
08	Dictionaries: look things up
09	Errors are messages, not insults
10	Build a whole program
A	Cheat sheet
B	Answers to the Try It exercises

Before you start

If you have ever followed a recipe, filled in a form or given someone directions, you already understand the main idea of programming. A program is a list of instructions, written so exactly that a machine can follow them without asking questions.

That's it. There is no secret math, no genius required. What makes programming feel hard is mostly the strange words and the fear of breaking something. This book removes both. Every new word gets explained the first time it appears, and nothing you type in these pages can damage your computer.

What you'll be able to do

By the last chapter you will have written a dozen small programs that actually do something: a tip calculator, a to-do list, a temperature converter, a word counter and a complete number-guessing game. More importantly, you'll understand *why* each line is there, so you can change it and write your own.

The language we use is **Python**. It's free, it runs on Windows, Mac and Linux, and its instructions read almost like English. It is also one of the most widely used programming languages in the world, in everything from websites to science labs to the spreadsheets of people who just got tired of doing the same thing by hand every Monday.

How to use this book

Read with a computer open next to you. Type every example yourself instead of copying and pasting. It feels slower, but your fingers learn things your eyes skip, and the small typing mistakes you make are exactly the ones you need to learn to spot.

In each chapter you'll see:

- **Code blocks** in a dark box. This is what you type.
- **Output blocks** in a light box. This is what Python shows back.
- **Try it** boxes with a small challenge. Answers are at the back, but give each one a real attempt first.

Chapters are short on purpose. One chapter per sitting is a good pace. If something doesn't click, read it again tomorrow; it usually clicks the second time.

> ***Nobody is born knowing how to code. Everybody who can do it learned it one small step at a time.***

CHAPTER

01

Say hello to Python

IN THIS CHAPTER

- Install Python and open the program you'll write code in
- Run your first instruction and your first saved program
- Know the difference between the shell and a script

Let's get Python onto your computer and make it say something. By the end of this chapter you'll have written and run a real program, even if it's a short one.

1.1 Install Python

Python is free from its official website, **python.org**. Don't download it from anywhere else.

1. Go to python.org and choose **Downloads**. The site detects your system and offers the latest version.
2. Download the installer and open it.
3. **On Windows:** on the first screen, tick the box that says **Add python.exe to PATH**, then click **Install Now**. That one tick saves a lot of trouble later.
4. **On a Mac:** open the downloaded package and click through the installer with the default settings.
5. When it finishes, look for a program called **IDLE**. On Windows, search for it in the Start menu. On a Mac, it's in the Python folder inside Applications.

Any version of Python 3 from 3.10 onward runs every example in this book. If the website offers a newer version, take it.

> NOTE
>
> Using a Chromebook, a work laptop you can't install on, or a tablet? The home page of python.org has a button to launch an interactive shell in your browser, which is enough for the first few chapters. For saving programs, you'll eventually want a computer where you can install Python.

1.2 Meet IDLE

IDLE is a simple program that comes with Python. Open it and you'll see a window with some version information and three arrows:

> OUTPUT
>
> ```
> >>>
> ```

This window is called the **shell**. The arrows are a **prompt**: Python is waiting for you to type something. Type this and press Enter:

```
print("Hello, world!")
```

> OUTPUT
>
> ```
> Hello, world!
> ```

Congratulations. You just told a computer to do something, in its own language, and it did it.

What just happened

`print` is a **command** built into Python. It means "show this on the screen." The round brackets hold what you want to show. The quotation marks tell Python that `Hello, world!` is text, not a command.

Programmers call a piece of text a **string**, as in a string of letters.

Try a few more. Change the words. Use single quotes instead of double; Python accepts both, as long as they match.

```
print('My name is Sam.')
print("I am learning Python.")
```

OUTPUT

```
My name is Sam.
I am learning Python.
```

Python as a calculator

The shell also does math. Type these one at a time:

```
print(2 + 3)
print(10 - 4)
print(6 * 7)
print(20 / 4)
```

OUTPUT

```
5
6
42
5.0
```

`*` means multiply and `/` means divide. Notice that `20 / 4` gives `5.0`, not `5`. Division in Python always gives a number with a decimal point, even when the answer is whole. That's normal, and we'll come back to it.

TIP

You don't even need `print` in the shell: type `2 + 3` and press Enter, and the shell shows `5`. But inside a saved program, which is what we do next, you always need `print` to see anything. Building the habit now saves confusion.

1.3 Your first saved program

The shell forgets everything when you close it. To keep a program, you write it in a file, called a **script**, and run the whole file at once.

1. In IDLE, choose **File → New File**. A blank window opens. This is the editor.
2. Type the three lines below.
3. Choose **File → Save**, make a folder called `python-practice` in your Documents, and save the file as `hello.py`. The `.py` ending tells your computer it's a Python program.
4. Choose **Run → Run Module**, or press **F5**.

```
print("Hello!")
print("This is my first program.")
print("2 + 2 is", 2 + 2)
```

OUTPUT

```
Hello!
This is my first program.
2 + 2 is 4
```

The output appears in the shell window. Python ran your lines in order, top to bottom, one after another. That order matters: a program is a sequence of steps.

Look at the third line. `print` can show several things at once if you separate them with commas. It puts a space between them for you.

> **WATCH OUT**
>
> Python is fussy about spelling and capitals. `Print("Hello")` with a capital P won't work, and neither will a missing quotation mark or bracket. If you get red text, compare your line with the book character by character. Chapter 9 explains how to read those red messages, and you'll find them a lot less scary.

> **TRY IT**
>
> **Try it 1.1** Write a program called `about_me.py` that prints three lines: your name, the city you live in, and how many years old you'll be in ten years. Let Python do the math for the last one.

CHAPTER SUMMARY

- Python is free from python.org. On Windows, tick “Add python.exe to PATH” while installing.
- IDLE gives you a shell (for quick experiments) and an editor (for saved programs).
- `print(...)` shows things on the screen. Text goes in quotes; numbers and math don’t.
- Python runs a script’s lines in order, top to bottom.

NEXT

In Chapter 2 you’ll give values names, so your programs can remember things.

CHAPTER

02

Boxes with names: values and variables

IN THIS CHAPTER

- Store a value in a variable and use it later
- Tell the difference between numbers and text
- Choose clear names that make programs easy to read