// STARTERCLASS FOR PYTHON

Write your first Python code this week.

A friendly, no-experience-needed starter guide: install Python, learn the core ideas and finish your first mini challenge.

Python 3.15 · 2026 edition | About 60 minutes | Exercises + answers | Glossary included

hello.py

```
name = input("What's your name? ")
print(f"Hi {name}, welcome to Python!")
```

OUTPUT

```
What's your name? Sal
Hi Sal, welcome to Python!
```

By Marsallis “Sal” Faircloth

Self-taught software engineer · creator of Sal Talks Life

saltalkslife.com →

// START HERE

Ready to learn Python?

Hey, I'm Sal! I taught myself to code at night while working a minimum-wage job at Devereux, right around the time my daughter was born. A bootcamp, an internship and two software engineering jobs later, I can tell you: Python is a great place to start.

In this free guide you'll...

- ✓ Learn what Python is and what it can do
- ✓ Install the latest version (Python 3.15)
- ✓ Write and run your first program
- ✓ Use variables, data types, input and math
- ✓ Dodge the 5 most common beginner mistakes
- ✓ Finish 6 practice exercises and a mini challenge

How to use this guide

- ☐ Go in order. Each page builds on the last one.
- ☐ Type every example yourself. Don't copy-paste.
- ☐ Do the exercises before peeking at the answers.
- ☐ Stuck? That's normal. Re-read, then try again.

What you need

A computer
Windows, Mac or Linux. A Chromebook works with an online editor.

About an hour
Split it up: 15 minutes a day adds up fast.

Zero experience
Seriously. If you can type, you can do this.

WANT THE FULL ROADMAP?

This starter guide is part 1. The **Zero to Hero Masterclass for Python** goes deeper with collections, loops, functions, libraries, real projects and career tips. Find it in my Payhip store (more on the last page).

Disclosure: some links in this guide are affiliate links (marked AFFILIATE). If you buy through them, I may earn a small commission at no extra cost to you. I only recommend things I would use myself.

// THE BIG PICTURE

What is Python?

Python is a programming language: a way to give your computer step-by-step instructions in words that look a lot like plain English.

Why beginners love it

- ✓ Clean, readable syntax with less punctuation
- ✓ Free and open-source, runs on Windows, Mac and Linux
- ✓ Huge community, so answers are easy to find
- ✓ Used for real jobs: web, data, automation and AI

Fun fact 🐍

Python is named after the comedy group **Monty Python**, not the snake. That's why tutorials love using "spam" and "eggs" as example words. It was first released in 1991 and is still one of the most popular languages today.

What can you build with it?

Automate boring tasks

Rename 1,000 photos, organize folders or fill in spreadsheets while you grab a coffee.

`Teachers · students · office work`

Work with data

Clean up spreadsheets, make charts and find patterns with libraries like pandas.

`Analysts · small business owners`

Build websites & apps

Frameworks like Django, Flask and FastAPI power real web apps and APIs.

`Freelancers · founders`

Use AI

Connect to AI model APIs to draft content, answer questions or summarize documents.

`Creators · marketers · investors`

KEY TAKEAWAY

Learn Python once and you can use it for websites, automation, data and AI.

// WHICH LANGUAGE FIRST?

Python or JavaScript?

Both are great first languages. Here's the honest difference, plus why Python code is so easy to read.

Python

Best for data

Data & automation

Pairs well with SQL

- ✓ Scripts, spreadsheets & reports
- ✓ AI and machine learning
- ✓ Good path to a tech job (plus Git)

JavaScript

Best for websites

Build websites

After HTML & CSS

- ✓ Front-end web pages
- ✓ Interactive apps in the browser
- ✓ Good path to a tech job (plus Git)

✦ Both are beginner-friendly, and the ideas **carry over**. Learn Python first and JavaScript gets easier later.

Same job, two languages

Here's a program that prints a greeting, written in Java and in Python. Notice how much less there is to type and understand in Python.

JAVA · 5 LINES

Hello.java

```java
public class Hello {
  public static void main(String[] args) {
    System.out.println("Hi!");
  }
}
```

PYTHON · 1 LINE

hello.py

```python
print("Hi!")
```

Less "ceremony" means you spend your energy on **ideas**, not punctuation. That's why so many schools and bootcamps start with Python.

KEY TAKEAWAY

Pick the one that matches your goal, start today and don't switch every week. Consistency matters more than the "perfect" language.

// STEP 1

Install Python

Python is free. The latest stable version as of fall 2026 is **Python 3.15**. Pick your computer below.

WINDOWS

Install manager

Go to python.org/downloads and get the **Python install manager** (python.org's recommended way on Windows now). It sets everything up for you.

Using the classic installer instead? Tick **"Add python.exe to PATH"** on the first screen.

MAC

macOS installer

Download the macOS installer from python.org/downloads, open it and click through. On a Mac, the command is `python3`, not `python`.

LINUX / CHROMEBOOK

Probably there

Most Linux systems already have Python 3. Check with the command below. On a Chromebook or school laptop, use Replit in your browser.

Check that it worked

Open a terminal: **Command Prompt** or PowerShell on Windows, **Terminal** on Mac. Type this and press Enter:

terminal

```
$ python --version
Python 3.15.1
# Mac/Linux: use python3 --version
```

See a version number that starts with **3**? You're ready to code. 🎉

"PYTHON IS NOT RECOGNIZED"

Python isn't on your PATH. Re-run the installer and tick "Add python.exe to PATH", or use the install manager. Then close and reopen the terminal.

OPENED THE MICROSOFT STORE?

On Windows, typing `python` before installing can open the Store. Install from python.org first, then try again.

Need a walkthrough?

Official: Using Python on Windows →

Install guide for every OS →

// STEP 2

Your first program

Code is just a text file with instructions. Python reads it from top to bottom, one line at a time.

1

IDLE

Comes with Python. Search your computer for "IDLE". Great for day one.

2

VS Code

Free editor most pros use. Add the Python extension.

3

Replit

Code in your browser. Nothing to install.

Write it

Create a file called `hello.py` and type:

hello.py

```
# This is a comment. Python ignores it.
print("Hello, world!")
print("I'm learning Python in 2026.")
```

Run it

In the terminal, go to the folder with your file and run:

terminal

```
$ python hello.py
Hello, world!
I'm learning Python in 2026.
```

What just happened?

- ✓ `print()` is a **function**: a built-in tool that shows text on the screen.
- ✓ The text inside quotes is a **string**.
- ✓ Lines starting with `#` are **comments**: notes for humans.
- ✓ Python ran each line in order, top to bottom.

TRY THE INTERACTIVE SHELL

Type `python` (or `python3`) with no file name. You get a `>>>` prompt where each line runs instantly. Type `exit()` to leave.

KEY TAKEAWAY

A Python program is a text file ending in .py. Write it, save it, run it with python file.py.

// STEP 3

Variables & data types

Think of a variable as a labeled storage box. You put a value in it and use the label to get it back later.

variables.py

```
name = "Alice"      # str
age = 27            # int
height = 5.6        # float
is_student = True   # bool

print(name, "is", age)
print(type(height))
```

OUTPUT

```
Alice is 27
<class 'float'>
```

The = sign means “store”

`age = 27` doesn’t mean “age equals 27” like in math. It means **put 27 in the box called age**. You can change it later: `age = 28`.

Tip: use `type()` to check what kind of value a variable holds.

TYPE	WHAT IT HOLDS	EXAMPLE	TYPE() SAYS
str	Text, inside quotes	`"spam"`	`<class 'str'>`
int	Whole numbers	`42`, `-7`	`<class 'int'>`
float	Decimal numbers	`12.5`	`<class 'float'>`
bool	Yes/no logic	`True`, `False`	`<class 'bool'>`

Naming rules

DO

- ✓ `first_name` lowercase with underscores
- ✓ `total_price` names that explain the value
- ✓ `score2` numbers are fine after a letter

DON'T

- ✗ `2score` can’t start with a number
- ✗ `first name` no spaces
- ✗ `print` don’t reuse built-in names

// STEP 4

Talk to your code

`print()` sends messages out. `input()` brings answers in. f-strings glue them together.

f-strings

Put an `f` before the quotes and drop variables inside `{curly braces}`.

python

```
city = "Newburgh"
year = 2026
print(f"Hello from {city} in {year}!")
```

OUTPUT

```
Hello from Newburgh in 2026!
```

input()

Pauses the program and waits for the user to type something.

python

```
food = input("Favorite food? ")
print(f"Mmm, {food} sounds good.")
```

OUTPUT

```
Favorite food? tacos
Mmm, tacos sounds good.
```

HEADS UP: INPUT() ALWAYS GIVES YOU A STRING

Even if someone types `25`, Python sees the text `"25"`. To do math with it, convert it with `int()` or `float()`.

Worked example: birthday math

birthday.py

```
age = int(input("How old are you? "))
next_age = age + 1
print(f"Next year you'll be {next_age}!")
```

OUTPUT

```
How old are you? 29
Next year you'll be 30!
```

Line by line

1. Ask a question and wait for an answer.
2. `int()` turns the text into a whole number.
3. Add 1 and store it in `next_age`.
4. Print a sentence with the result inside.

KEY TAKEAWAY

print() shows, input() asks, f-strings mix text and values. Convert input before doing math.

// STEP 5

Math & operators

Yes, Python can check your homework. These work on ints and floats.

OP	MEANING	EXAMPLE → RESULT
+	Add	7 + 3 → 10
-	Subtract	7 - 3 → 4
*	Multiply	7 * 3 → 21
/	Divide	7 / 2 → 3.5
//	Floor divide	7 // 2 → 3
%	Remainder	10 % 3 → 1
**	Power	2 ** 3 → 8

Order of operations

Python follows the same order you learned in school: parentheses, powers, multiply/divide, then add/subtract.

python

```
print(2 + 3 * 4)      # 14
print((2 + 3) * 4)    # 20
```

Note: `/` always gives a float, even for `8 / 2` (4.0).

Remainder trick: `number % 2 == 0` is True for even numbers.

Worked example: tip calculator

tip.py

```
bill = float(input("Bill total: $"))
tip_percent = 20
tip = bill * tip_percent / 100
total = bill + tip
print(f"Tip: ${tip:.2f}")
print(f"Total: ${total:.2f}")
```

OUTPUT

```
Bill total: $48.50
Tip: $9.70
Total: $58.20
```

New trick: :.2f

Inside an f-string, `{total:.2f}` rounds to **2 decimal places**, perfect for money.

Try it: change `tip_percent` to 18 or 25 and run it again.

Practice resources (15 minutes a day adds up!)

- **Python basics →** W3Schools
- **Full beginner course (video) →** freeCodeCamp
- **Code in your browser →** Replit

// SAVE YOURSELF AN HOUR

5 common beginner mistakes

Every programmer makes these. Errors aren't failure; they're Python telling you exactly where to look.

1 Adding text and numbers

× Breaks

TypeError

```
age = 30
print("Age: " + age)
```

✓ Works

fixed

```
age = 30
print(f"Age: {age}")
```

You can't + a string and an int. Use an f-string or str(age).

2 Forgetting to convert input

× Breaks

wrong answer

```
n = input("Number: ")
print(n * 2)  # "55"
```

✓ Works

fixed

```
n = int(input("Number: "))
print(n * 2)  # 10
```

input() returns text, so "5" * 2 repeats it. Convert with int().

3 Mismatched quotes or brackets

× Breaks

SyntaxError

```
print("Hello')
```

✓ Works

fixed

```
print("Hello")
```

Start and end with the same quote type, and close every (you open.

// COMMON MISTAKES · CONTINUED

4 Using = when you mean ==

× Breaks

SyntaxError

```
if age = 18:
    print("Adult")
```

✓ Works

fixed

```
if age == 18:
    print("Adult")
```

One = stores a value. Two == compares values.

5 Capital letters matter

× Breaks

NameError

```
Print("hi")
```

✓ Works

fixed

```
print("hi")
```

Python is case-sensitive: it's print, True and False, exactly like that.

// DEBUGGING 101

How to read an error message

Python error messages look scary, but they always tell you three things. Read them from the **bottom up**.

terminal

```
Traceback (most recent call last):
  File "age.py", line 2, in <module>
    print("Age: " + age)
TypeError: can only concatenate str (not "int") to str
```

1 · What
The last line names the error type: TypeError.

2 · Where
"line 2" in age.py is where it broke.

3 · Why
The message: you tried to + a string and an int.

Still stuck? Paste the last line of the error into a search engine or an AI assistant and ask it to **explain** the error, not just fix it.

// YOUR TURN

Practice exercises

Write each one in a new file. Answers are on the next page, but try first: struggling a little is how it sticks.

1 Say hello

Print your name and your favorite hobby on two lines.

2 Store & show

Make variables for a city and a year. Print "I live in ___ in ____".

Hint: use an f-string

3 Type detective

Print the type of 7, 7.0, "7" and True.

Hint: type()

4 Minutes in a week

Calculate and print how many minutes are in one week.

Hint: 60 × 24 × 7

5 Even or odd?

Ask for a number and print True if it's even.

Hint: int() and %

6 Pizza split

Ask for the number of slices and people. Print slices each and leftovers.

Hint: // and %

// MINI CHALLENGE

The favorite-color sentence

- [ ] Ask for the user's name
- [ ] Ask for their favorite color
- [ ] Print a fun sentence using both

Try it in Replit or VS Code. Hint: `input()` + an f-string.

example output

```
Hi Alice, I bet your
favorite color is blue
because you're awesome!
```

// CHECK YOUR WORK

Answers

Your code can look different and still be right. If it runs and prints the right thing, you nailed it.

1 · SAY HELLO

```python
print("Sal")
print("Basketball")
```

2 · STORE & SHOW

```python
city = "New York"
year = 2026
print(f"I live in {city} in {year}")
```

3 · TYPE DETECTIVE

```python
print(type(7))      # int
print(type(7.0))    # float
print(type("7"))    # str
print(type(True))   # bool
```

4 · MINUTES IN A WEEK

```python
minutes = 60 * 24 * 7
print(minutes)   # 10080
```

5 · EVEN OR ODD?

```python
n = int(input("Number: "))
print(n % 2 == 0)
```

6 · PIZZA SPLIT

```python
slices = int(input("Slices: "))
people = int(input("People: "))
print("Each gets", slices // people)
print("Leftover:", slices % people)
```

MINI CHALLENGE

```challenge.py
name = input("What's your name? ")
color = input("Favorite color? ")
print(f"Hi {name}, I bet your favorite color is {color} because you're awesome!")
```

LEVEL IT UP

Add a third question (favorite food?) or use `.upper()` to shout the color: `color.upper()`.

// CHEAT SHEET

Glossary

Variable	A named box that stores a value.
Integer (int)	A whole number, like 42.
Boolean (bool)	True or False.
Argument	The value you pass into a function.
Syntax	The grammar rules of a language.
Terminal	A text window for typing commands.
String (str)	Text inside quotes, like "hello".
Float	A decimal number, like 3.5.
Function	A reusable tool you call with (), like print().
f-string	A string starting with f that fills in {variables}.
Error / bug	Something that stops code or makes it wrong.
PATH	The list of places your computer looks for programs.

// WHAT'S NEXT

Next steps: pick one this week

1. Build a tiny app: a calculator or to-do list
2. Learn to store data with lists and dictionaries
3. Use Python to automate something on your computer
4. Explore real tools like pandas, Django or AI APIs

GUIDED PRACTICE

I learned on DataCamp

Short interactive lessons right in your browser, so you practice as you learn. A solid next step after this guide.

Try DataCamp's Python courses →

The DataCamp button is an affiliate link. If you sign up through it, I may earn a small commission at no extra cost to you.

// PLAYER BIO

Hey, I'm Sal.

Marsallis Faircloth: dad of four, husband, software engineer and the creator of Sal Talks Life.

I didn't start in tech. I taught myself to code **at night**, while working a minimum-wage job at Devereux supporting people with disabilities, right around the time my daughter was born.

Those late nights turned into a software engineering bootcamp, an extended internship and then two software engineering jobs. Along the way I picked up data science, NetSuite and Celigo integrations, and I still build software for clients today.

I make these guides because I remember how confusing the start was. Everything here is the plain-English version I wish someone had handed me.

FIND ME ONLINE

Site · saltalkslife.com
TikTok · @saltalkslife
YouTube · @SalTalksLife
LinkedIn · in/marsallisfaircloth
GitHub · Marsallis

Night-shift learner
Self-taught first, while working full time

Bootcamp → jobs
Internship, then two engineering roles

Still building
Python, data, NetSuite & Celigo, web apps

// NEXT LEVEL

Go from beginner to job-ready

You just wrote real Python. Here's how to keep the momentum going.

1

Code daily

15–30 minutes beats one long weekend session.

2

Build something

Small projects teach more than more tutorials.

3

Share it

Post it on GitHub or LinkedIn. Proof beats promises.

// FREE NEWSLETTER

Get my best tech, money & life tips

Plain-English lessons, new free guides and what I'm building. No spam, unsubscribe anytime.

Join the newsletter →

// THE FULL ROADMAP

Zero to Hero Masterclass for Python

My favorite free & paid course picks, real-world project ideas, AI + web dev + data science explained simply, and career resources to turn Python into income.

Get it on Payhip →

WEBSITE
saltalkslife.com

TIKTOK
@saltalkslife

PAYHIP STORE
payhip.com/saltalkslife