

Lab 01: Python Building Blocks

COMP 102 — Introduction to Computing
Forman Christian University — Spring 2026

A Walkthrough Tutorial

Duration: ~3 hours — Based on Lectures 2 & 3

How to Use This Lab

This lab is a **guided walkthrough** — not a test. For every exercise, follow this pattern:

Step 1: Predict — Write down what you think will happen *before* touching the computer.

Step 2: Type — Enter the code in Thonny's **Shell** (interpreter) or **Code Editor**.

Step 3: Verify — Compare your prediction with the actual result. If they differ, figure out *why*.

Setup: Open **Thonny** on your computer. You should see the Shell window at the bottom (with >>>).

1 Scalar Objects & Types (~15 min)

Ref: Lecture 2 — Scalar Objects slides

Exercise 1

Very Easy

Type the following in the Thonny Shell. Predict the output of each **before** you type it.

```
>>> type(42)
>>> type(3.14)
```

Your predictions: _____

Exercise 2

Very Easy

Predict the output, then verify:

```
>>> type(True)
>>> type(None)
```

Your predictions: _____

Exercise 3

Medium

Careful! Predict the output of each, then verify:

```
_____
```

```
>>> type(9.0)
>>> type(7)
>>> type(False)
```

Why is `type(9.0)` not `int`, even though 9 is a whole number?

Exercise 4

Medium

Without running, predict the type returned by each. Then verify:

```
>>> type(0)
>>> type(0.0)
>>> type(-3.5)
```

Your predictions: _____

Exercise 5

Hard

Without running any code, predict the **type** of the result of each expression. Then verify with `type(...)`.

```
>>> 3 + 4.0
>>> 10 / 2
>>> 10 // 2
```

Your predictions: _____

Hint: Think about what happens when you mix `int` and `float` in an operation.

2 Type Conversions (~15 min)

Ref: Lecture 2 — Type Casting slides

Exercise 6

Very Easy

Predict the value, then verify in the Shell:

```
>>> float(5)
>>> int(3.0)
```

Your predictions: _____

Exercise 7

Very Easy

Predict the value, then verify:

```
>>> float(True)
>>> int(False)
```

Your predictions: _____

Exercise 8

Medium

Tricky! Predict the value of each. Pay close attention to the difference:

```
>>> int(7.9)
>>> round(7.9)
```

Your predictions: _____
Why are they different?

Exercise 9

Medium

Predict the value and type:

```
>>> int(3.1) + int(4.9)
>>> round(3.1) + round(4.9)
```

Your predictions: _____

Exercise 10

Hard

Trace through this expression step by step. Write down each intermediate result:

```
>>> float(round(int(9.7) + 0.6))
```

Step 1: `int(9.7)` = _____
Step 2: `___ + 0.6` = _____
Step 3: `round(___)` = _____
Step 4: `float(___)` = _____

3 Arithmetic Expressions (~25 min)

Ref: Lecture 2 — Expressions slides

Exercise 11

Very Easy

Predict the value of each expression, then verify:

```
>>> 3 + 4
>>> 10 - 6
```

Your predictions: _____

Exercise 12

Very Easy

Predict the value of each expression, then verify:

```
_____
```

```
>>> 10 / 2
>>> 2 ** 3
```

Your predictions: _____

Exercise 13

Very Easy

Predict the value of each expression, then verify:

```
>>> 7 % 3
>>> 10 % 5
```

Your predictions: _____

Exercise 14

Very Easy

Predict the value of each expression, then verify:

```
>>> 15 // 4
>>> 7 // 2
```

Your predictions: _____

Exercise 15

Medium

Operator precedence matters! Predict the value of each:

```
>>> 3 + 4 * 2
>>> (3 + 4) * 2
```

Your predictions: _____

Why are they different?

Exercise 16

Medium

Predict the **type** returned by each:

```
>>> type(4 * 3)
>>> type(4.0 * 3)
```

Your predictions: _____

Exercise 17

Medium

Predict the value:

```
>>> (13 - 4) / (12 * 12)
```

Your prediction: _____

Hint: What does the / operator always return?

Exercise 18

Medium

Predict the value of each:

```
>>> 5 % 2
>>> 2 ** 5
>>> 17 % 5
```

Your predictions: _____

Exercise 19

Medium

Right-to-left! The `**` operator is right-associative. Predict the value:

```
>>> 2 ** 3 ** 2
```

Your prediction: _____

*Hint: This evaluates as $2 ** (3 ** 2)$, not $(2 ** 3) ** 2$.*

Exercise 20

Hard

Trace through this expression step by step on paper, then verify:

```
>>> (13 - 4) / (2 * 3) + 5 % 3 ** 2
```

Show your work:

4 Call Expressions & Imports (~20 min)

Ref: Lecture 2 — Call Expressions slides

Exercise 21

Very Easy

Predict the output, then verify:

```
>>> max(3, 5)
>>> max(10, 2, 7)
```

Your predictions: _____

Exercise 22

Very Easy

Predict the output, then verify:

```
>>> min(1, 2, 3)
```

```
>>> min(8, 5)
```

Your predictions: _____

Exercise 23

Medium

Order matters! Predict the output of each:

```
>>> pow(2, 10)
>>> pow(10, 2)
```

Your predictions: _____

What does `pow(a, b)` compute?

Exercise 24

Medium

Type these two lines in the Shell:

```
>>> from math import sqrt
>>> sqrt(144)
```

Your prediction for `sqrt(144)`: _____

What is the **type** of the result? _____

Exercise 25

Hard

First, type this import in the Shell:

```
>>> from operator import add, mul, sub
```

Now trace through this nested expression step by step:

```
>>> add(mul(2, 3), pow(2, sub(5, 2)))
```

Step 1: `mul(2, 3)` = _____

Step 2: `sub(5, 2)` = _____

Step 3: `pow(2, ___)` = _____

Step 4: `add(____, ___)` = _____

5 Variables & Assignment (~25 min)

Ref: *Lecture 2 — Variables slides*

Exercise 26

Very Easy

Type in the Shell:

```
>>> x = 10
>>> print(x)
```

What do you see? _____

Exercise 27

Very Easy

Type in the Shell:

```
>>> greeting = "Hello"  
>>> print(greeting)
```

What do you see? _____

Exercise 28

Very Easy

Which of the following are **valid** variable names in Python? Write **Valid** or **Invalid** next to each:

- (a) my_age _____
- (b) 2fast _____
- (c) _count _____
- (d) my-name _____
- (e) lunchPrice _____

Exercise 29

Very Easy

Type in the Shell:

```
>>> pi = 3.14  
>>> radius = 5  
>>> area = pi * radius ** 2  
>>> print(area)
```

What does print(area) display? _____

Exercise 30

Medium

Predict the value of x after this code runs:

```
>>> x = 5  
>>> x = x + 1  
>>> print(x)
```

Your prediction: _____

Exercise 31

Medium

Predict the value of y after this code:

```
>>> x = 3
>>> y = x * 2
>>> x = 10
>>> print(y)
```

Your prediction: _____
Does changing x afterward affect y ?

Exercise 32

Medium

Trace through this code. What is the value of `feet` at the end?

```
>>> meters = 100
>>> feet = 3.2808 * meters
>>> meters = 200
>>> print(feet)
```

Your prediction: _____
Why doesn't `feet` update when `meters` changes?

Exercise 33

Medium

Predict the value of `b`:

```
>>> a = 5
>>> b = a
>>> a = 10
>>> print(b)
```

Your prediction: _____

Exercise 34

Medium

Trace through carefully. What are the final values of `x` and `y`?

```
>>> x = 2
>>> y = x + 1
>>> x = y + 1
>>> y = x + 1
>>> print(x, y)
```

Your prediction for `x`: _____ `y`: _____

Exercise 35

Hard

Given `x = 1` and `y = 2`, swap their values so that `x` becomes 2 and `y` becomes 1. You **must not** type the numbers 1 or 2 directly — use a **temporary variable**.

```
x = 1
```

```
y = 2

# Write your swap code below:

# After your code, x should be 2 and y should be 1
print(x, y)    # Should print: 2 1
```

Write your solution:

6 Strings — Basics (~20 min)

Ref: Lecture 3 — Strings slides

Exercise 36

Very Easy

Type in the Shell:

```
>>> name = "Python"
>>> print(name)
>>> len(name)
```

What does `len(name)` return? _____

Exercise 37

Very Easy

Type in the Shell:

```
>>> first = "Hello"
>>> second = "World"
>>> print(first + " " + second)
```

What do you see? _____

Exercise 38

Medium

Predict the value of `s1`:

```
>>> b = ":"
>>> c = ")"
>>> s1 = b + 2*c
>>> print(s1)
```

Your prediction: _____

Exercise 39

Medium

Predict the values:

```
>>> word = "ab"
>>> result = word * 3
>>> print(result)
>>> print(len(result))
```

result = _____ len(result) = _____

Exercise 40

Hard

Predict the value of s2:

```
>>> f = "a"
>>> g = "b"
>>> h = "3"
>>> s2 = (f + g) * int(h)
>>> print(s2)
```

Step 1: f + g = _____

Step 2: int(h) = _____

Step 3: ___ * ___ = _____

7 String Indexing & Slicing (~25 min)

Ref: Lecture 3 — Indexing & Slicing slides

s = "	c	o	m	p	1	0	2	"
	0	1	2	3	4	5	6	
	-7	-6	-5	-4	-3	-2	-1	

+ve indices

-ve indices

Exercise 41

Very Easy

Using s = "comp102", predict the value of each:

```
>>> s = "comp102"
>>> s[0]
>>> s[3]
```

Your predictions: _____

Exercise 42

Very Easy

Predict the value of each:

```
>>> s = "comp102"
>>> s[4]
```

```
>>> s[6]
```

Your predictions: _____

Exercise 43

Very Easy

Predict the value of each slice:

```
>>> s = "abcdefgh"
>>> s[2:5]
>>> s[0:3]
```

Your predictions: _____

Exercise 44

Very Easy

Predict the value:

```
>>> s = "abcdefgh"
>>> s[:4]
>>> s[5:]
```

Your predictions: _____

Exercise 45

Medium

Negative indexing! Predict the value of each:

```
>>> s = "comp102"
>>> s[-1]
>>> s[-3]
```

Your predictions: _____

Exercise 46

Medium

Predict the value:

```
>>> s = "abcdefgh"
>>> s[::2]
>>> s[1::2]
```

Your predictions: _____

What does the step of 2 do?

Exercise 47

Medium

Predict the value:

```
_____
```

```
>>> s = "abcdefgh"
>>> s[::-1]
```

Your prediction: _____
 What does a step of -1 do?

Exercise 48

Medium

Predict the value:

```
>>> s = "ABC d3f ghi"
>>> s[3:len(s)-1]
```

Step 1: `len(s)` = _____
 Step 2: `len(s) - 1` = _____
 Step 3: `s[3:___]` = _____

Exercise 49

Medium

Predict the value:

```
>>> s = "ABC d3f ghi"
>>> s[4:0:-1]
```

Your prediction: _____
 Hint: With a negative step, you go from index 4 **down to** (but not including) index 0.

Exercise 50

Hard

Two challenges:

(a) Predict the value:

```
>>> s = "abcdefgh"
>>> s[6:1:-2]
```

Your prediction: _____

(b) Strings are **immutable** — you cannot change individual characters. Given `s = "cat"`, write an expression that produces the string "rat" using `s`, **without** typing any of the letters `a` or `t` directly.

Hint: Concatenate a new first character with a slice of `s`.

8 Input / Output (~20 min)

Ref: Lecture 3 — I/O slides

Remember

`input()` **always** returns a **string**, even if the user types a number. You must cast it with `int()` or `float()` if you want to do arithmetic.

Exercise 51

Very Easy

Write the following in Thonny's **code editor** (not the Shell) and run it:

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

What appears in the output? _____

Exercise 52

Very Easy

Write and run this program:

```
name = input("What is your name? ")
print("Hello, " + name + "!")
```

If you enter Ali, what does it print? _____

Exercise 53

Medium

Write and run this program. Enter 5 when asked:

```
num = int(input("Enter a number: "))
result = num * 3
print(result)
```

What does it print? _____

What would happen if you removed the `int()` cast?

Exercise 54

Medium

Predict first! If the user enters 4, what does each line print?

```
x = input("Number: ")
print(x * 3)
print(int(x) * 3)
```

Line 2 prints: _____ Line 3 prints: _____

Why are they different?

Exercise 55

Hard

Write a program that:

1. Asks the user to enter a verb (e.g., **run**)

2. Prints I can ___ better than you! (replacing ___ with the verb)

3. Prints the verb 5 times in a row, separated by spaces

Example output (if user enters run):

```
I can run better than you!  
run run run run run
```

Write your code:

9 f-strings (~10 min)

Ref: Lecture 3 — f-strings slides

Exercise 56

Very Easy

Predict what this prints, then verify:

```
name = "Ali"  
print(f"Hello, {name}!")
```

Your prediction: _____

Exercise 57

Very Easy

Predict what this prints, then verify:

```
x = 10  
y = 20  
print(f"{x} + {y} = {x + y}")
```

Your prediction: _____

Exercise 58

Medium

Predict what this prints:

```
word = "Python"  
print(f"{word} has {len(word)} letters")
```

Your prediction: _____

Exercise 59

Medium

Predict what this prints:

```
a = 7
b = 3
print(f"{a}/{b} = {a/b}")
```

Your prediction: _____

Exercise 60

Hard

Predict what the following code prints if the user enters 3 and 4:

```
base = int(input("Base: "))
exp = int(input("Exponent: "))
result = base ** exp
print(f"{base}^{exp} = {result}")
```

Your prediction: _____

10 Capstone Challenges (~15 min)

These problems combine everything you have learned. Write your solutions in Thonny's code editor.

Exercise 61

Medium

Temperature Converter. Write a program that:

1. Asks the user for a temperature in Celsius
2. Converts it to Fahrenheit using: $F = \frac{9}{5}C + 32$
3. Prints both values using an f-string

Example (user enters 37):

```
37 C is 98.6 F
```

Write your code:

Exercise 62

Medium

Tip Splitter. Write a program that:

1. Asks for the bill amount
2. Asks for the number of friends
3. Adds a 15% tip to the bill
4. Splits the total equally and prints each person's share

Example (user enters 80 and 4):

```
Each person pays: 23.0
```

Write your code:

Exercise 63

Medium

Initials. Write a program that:

1. Asks for a first name and a last name
2. Prints the user's initials in uppercase, separated by a dot

Example (user enters `ali` and `khan`):

```
A . K .
```

Hint: look up what `.upper()` does to a string.

Write your code:

Exercise 64

Hard

Time Breakdown. Write a program that:

1. Asks the user for a total number of seconds
2. Converts it to hours, minutes, and remaining seconds
3. Prints the result

Example (user enters `3672`):

```
1 hour(s), 1 minute(s), 12 second(s)
```

Hint: use integer division (`//`) and modulo (`%`).

Write your code:

Exercise 65

Hard

Swap Halves. Write a program that:

1. Asks the user for a word (assume it has an even number of letters)
2. Prints the word with its two halves swapped

Example (user enters `abcdef`):

```
def abc
```

Hint: use `len()` and slicing.

Write your code:

Exercise 66

Hard

Distance Between Two Points. Write a program that:

1. Asks for four values: `x1`, `y1`, `x2`, `y2`
2. Computes the Euclidean distance: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
3. Prints the result

Example (user enters 0, 0, 3, 4):

```
Distance: 5.0
```

Hint: recall that $\sqrt{x} = x^{0.5}$ in Python.

Write your code:

Exercise 67

Hard

Digit Sum (3-digit number). Write a program that:

1. Asks the user for a 3-digit number
2. Computes the sum of its three digits **without using strings**
3. Prints the result

Example (user enters 472):

```
4 + 7 + 2 = 13
```

Hint: use `//` and `%` to extract each digit.

Write your code:

Self-Assessment Checklist

Before you leave, check off what you can do:

- I can identify the type of a Python literal (`int`, `float`, `bool`, `NoneType`)
- I can convert between types using `int()`, `float()`, `str()`, and `round()`
- I know the difference between `int(7.9)` and `round(7.9)`
- I can evaluate arithmetic expressions with correct operator precedence
- I can use built-in functions (`max`, `min`, `pow`) and import from modules
- I can assign variables and trace through code with re-binding
- I understand that re-assigning a variable does **not** update previous calculations
- I can create, concatenate, and repeat strings
- I can index strings with positive and negative indices
- I can slice strings using `[start:stop:step]`
- I know that `input()` always returns a string
- I can use f-strings to format output with expressions