Learn Python Programming with Manga:

A Beginner's Guide

Table of Contents

Chapter 1: Prologue

1-1: New Employee Training

Chapter 2: Preparing the Basics

2-1: What is a Program?
2-2: Setting Up the Development Environment
2-3: Running a Simple Program
2-4: Future Direction

Chapter 3: Learning the Fundamentals

3-1: Execution Order and Indentation
3-2: Comments
3-3: Elements of a Program 1
3-4: Functions
3-5: Values
3-6: Operators
3-7: Elements of a Program 2
3-8: Variables and Assignment
3-9: Importing
3-10: pip
3-11: Custom Modules

Chapter 4: Data Types

4-1: Data Types
4-2: Integers and Floats
4-3: Text
4-4: Boolean Values and Comparison Operators
4-5: None

Chapter 5: Data Types with Multiple Values

5-1: Lists
5-2: Converting Between Lists and Text
5-3: Tuples
5-4: Dictionaries
5-5: Sets

Chapter 6: Conditional Statements

6-1: if Statements
6-2: if Statements 2
6-3: Multiple Comparisons
6-4: Various Comparisons

Chapter 7: Looping Structures

7-1: while Loops
7-2: for Loops
7-3: The enumerate Function
7-4: The range Function
7-5: Using Dictionaries with Loops
7-6: Nested Lists and Loops
7-7: break and continue Statements
7-8: List Comprehensions

Chapter 8: Creating Functions

8-1: Defining Functions
8-2: pass Statement
8-3: Default Values and Keyword Arguments
8-4: Unpacking and Variable-Length Arguments
8-5: Lambda Functions

Chapter 9: Errors and Error Handling

9-1: Encountering Errors
9-2: Debugging by Reading Error Messages
9-3: Another Error Occurred
9-4: Error Handling

Chapter 10: File Handling

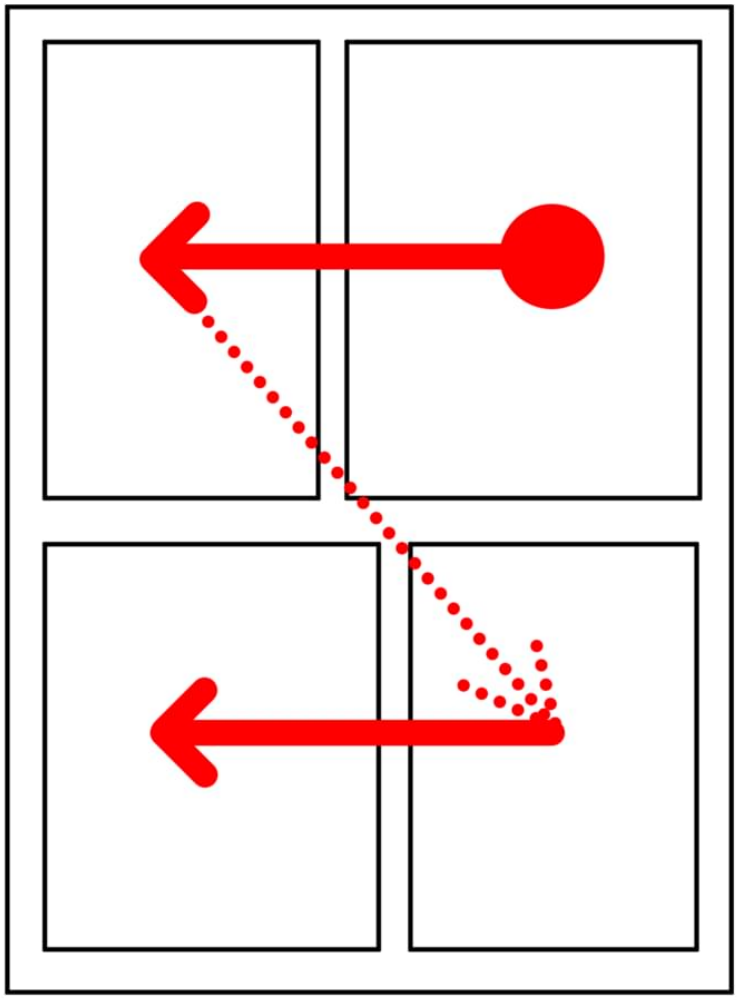
10-1: CWD and Program Paths
10-2: Program Paths and Filenames
10-3: Path Manipulation
10-4: Reading Text Files
10-5: Writing Text Files
10-6: Reading and Writing CSV Files
10-7: Reading and Writing JSON Files
10-8: Listing Files
10-9: Copying and Deleting Files

Chapter 11: Epilogue

11-1: Summary
11-2: Looking Ahead
11-3: Reporting to the CEO

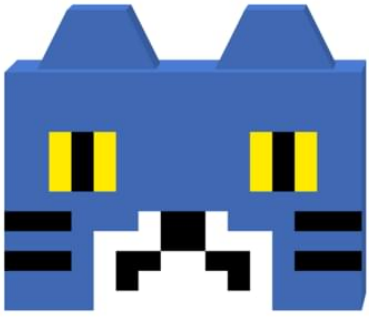
How to Read This Book

- This book is in a manga (Japanese comic) style.
- The panels start from the top right.
- The panels proceed from right to left,
  and when moving to the next row,
  they proceed from right to left again.

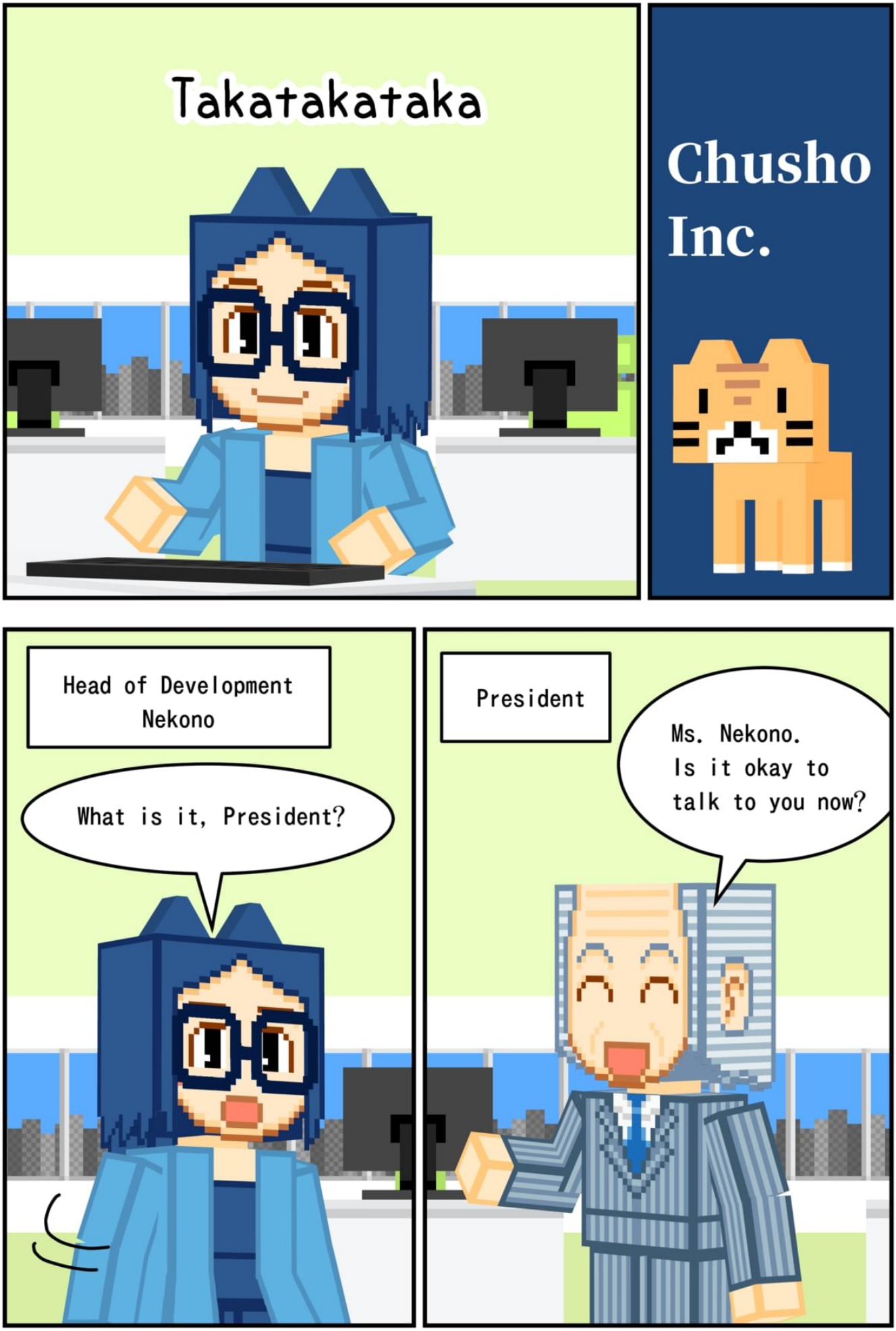


Chapter 1

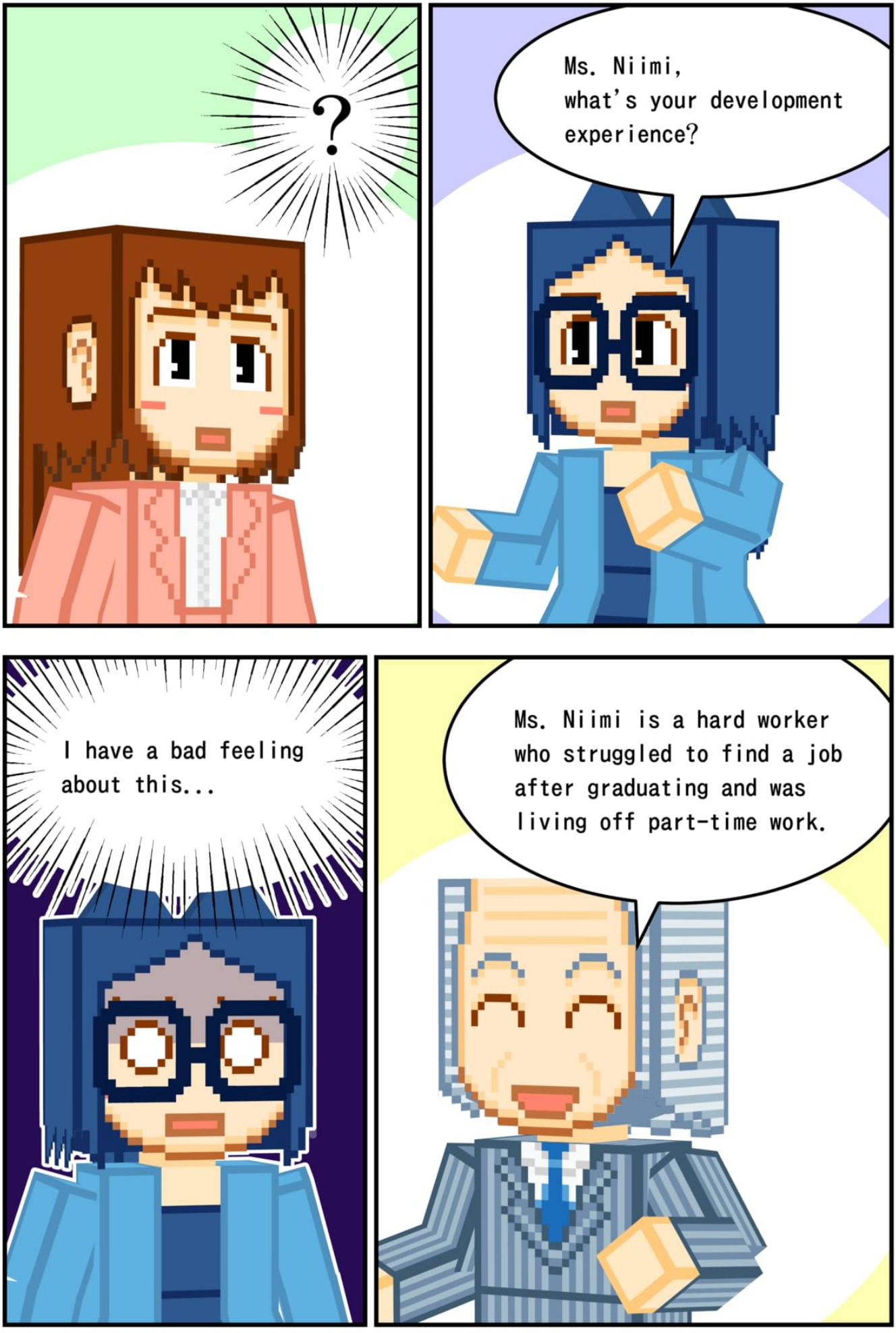
Prologue




1-1: New Employee Training


Takatakataka
Chusho Inc.
Head of Development Nekono
What is it, President?
President
Ms. Nekono. Is it okay to talk to you now?


Huh?! When did you hire them?
I wasn't consulted at all about the selection...
You've been saying we need more programmers, so I hired a new employee.
I'm Niimi!
Nice to meet you!
Here she is.


?
Ms. Niimi,
what's your development experience?
I have a bad feeling about this...
Ms. Niimi is a hard worker who struggled to find a job after graduating and was living off part-time work.


This president has done it again...
I happened to hear their story at a shop, so I scouted her!
Ah, um... Right...
Leader! Looking forward to working with you!
Where should I start?!


Ha...
haha...
Anyway,
I'll leave the programming
training to you, Ms. Nekono.
Now our workforce
is doubled!
Alright,
let's take our time
and learn.
By the way, Leader,
what is programming?

Characters

Niimi

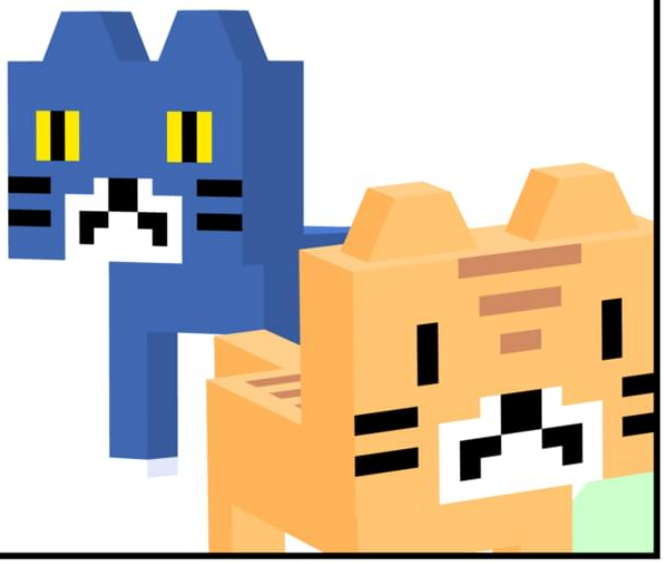
New employee. Hard worker.
No programming experience.

Nekono

Head of Development.
A programmer struggling with solo work

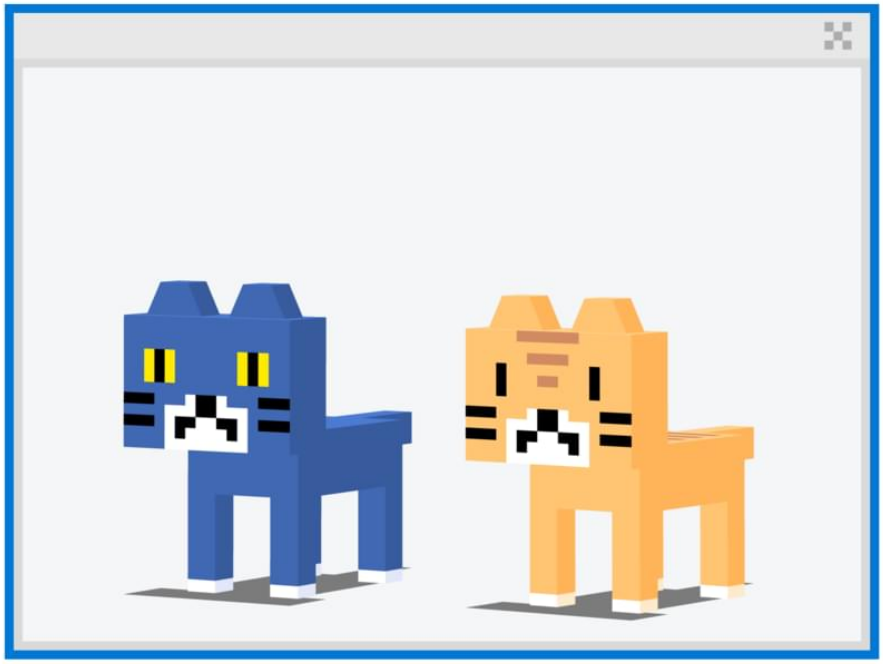
Blue & Tiger

Company cats. They just live in the office.



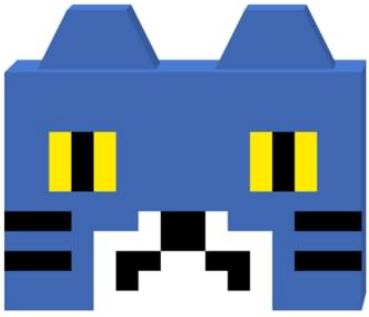
President

Picks up employees like stray puppies.



Chapter 2

Preparing the Basics

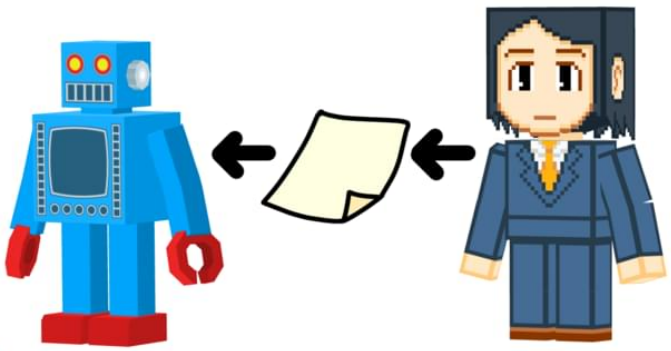


2-1: What is a Program?


Uh, okay...
New employee training...
Where shall I start?
Chusho Inc.
Development Department
Meow
Yeah, that's a good place to start.
Leader!
So, what exactly is a program?

Program

Human
↓
Instructions
↓
Computer

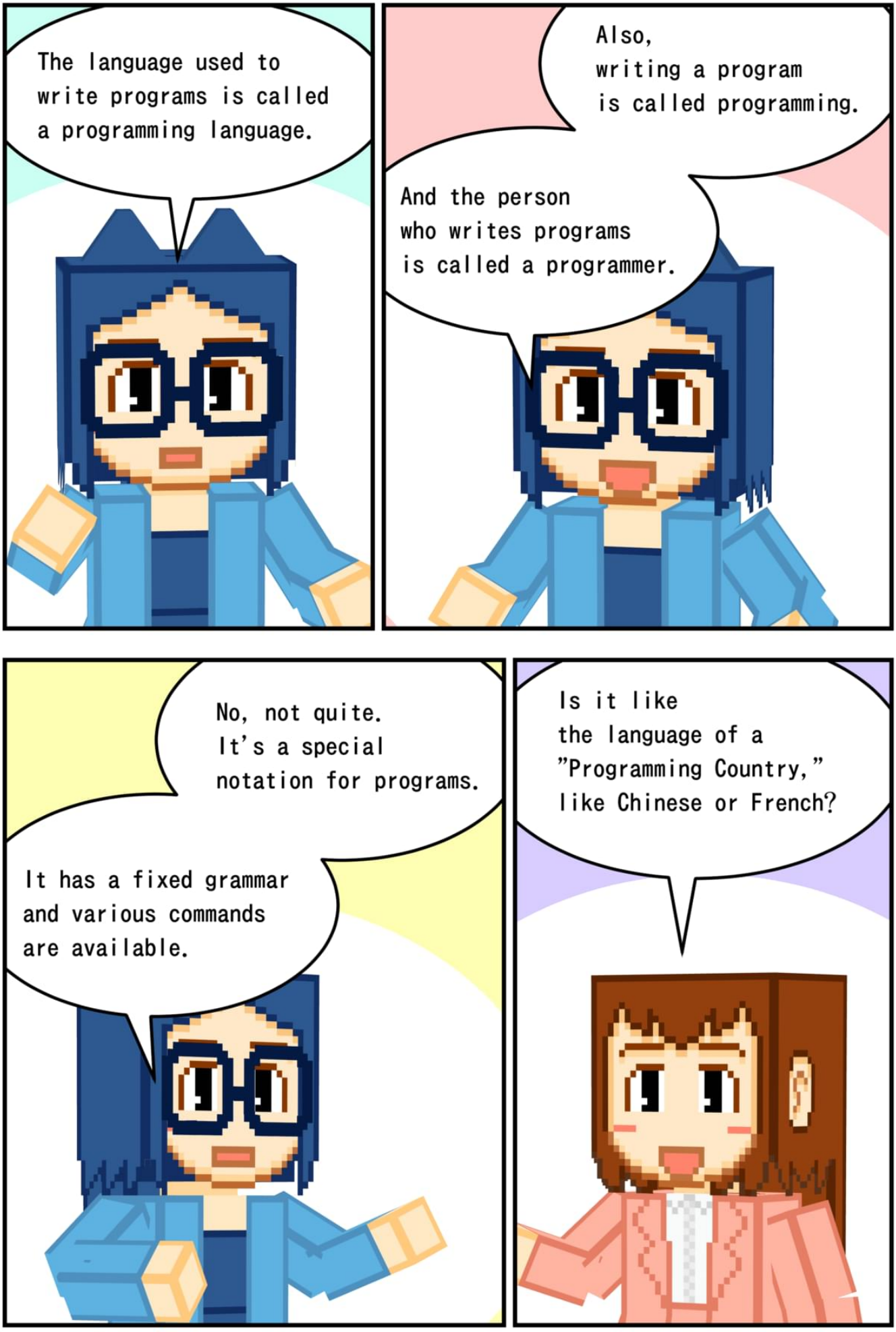


A program is
an instruction manual
for a computer.

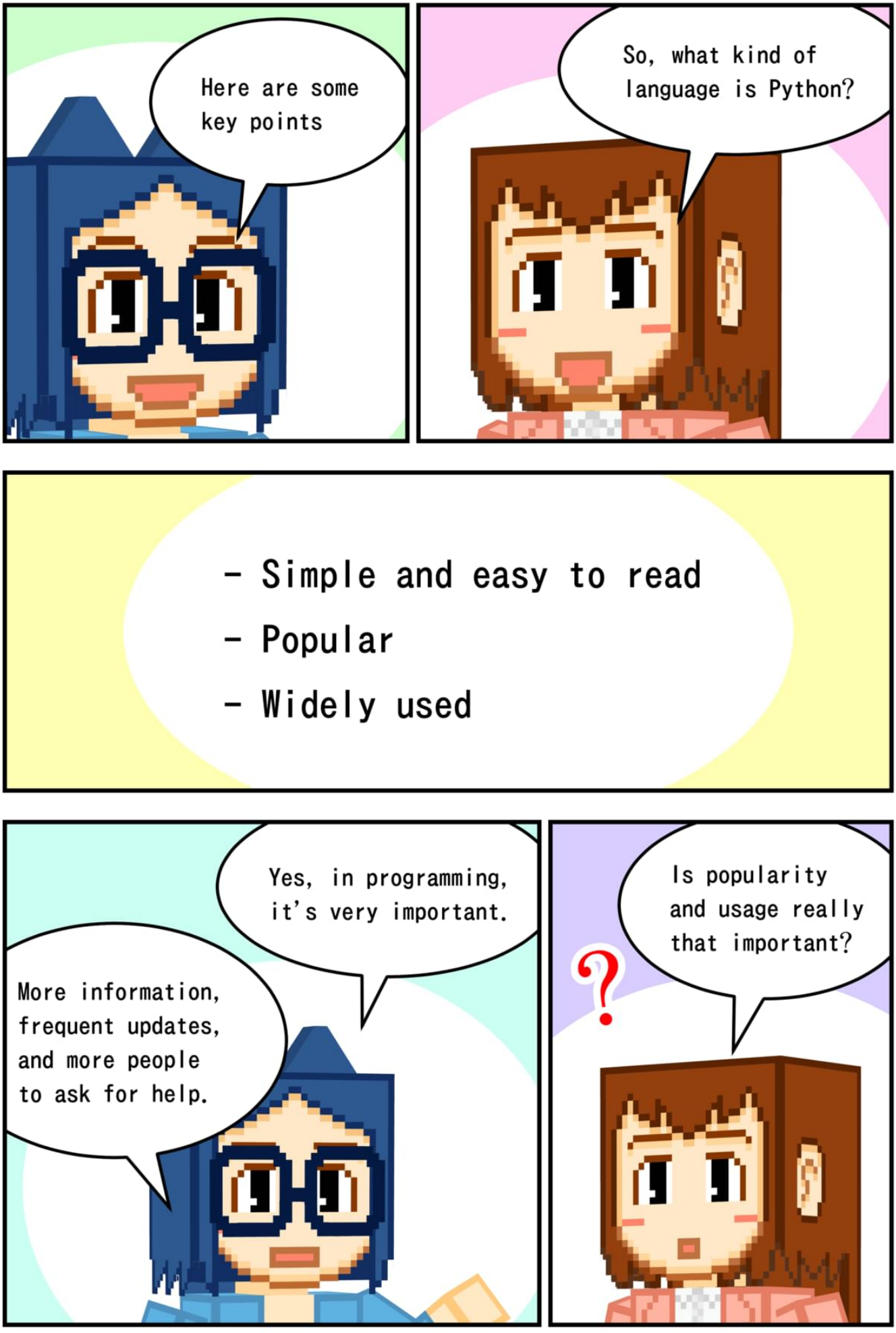
A document that outlines
what happens in order
is called a program.

We also use the word "program"
for event schedules,
like a graduation ceremony
or a sports day.

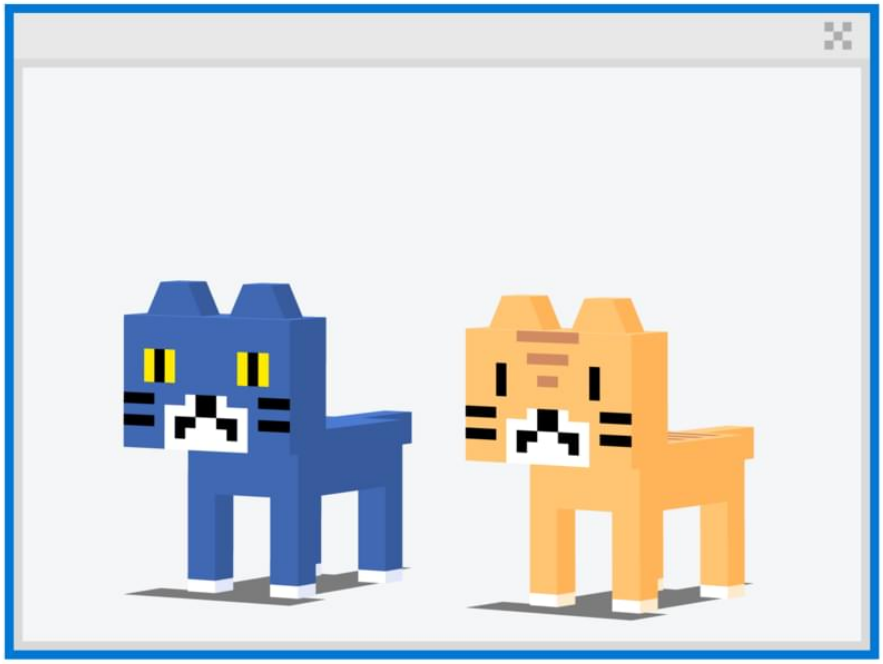
1. Opening Remarks
2. Entrance of Graduates
3. School Anthem Performance
4. Diploma Presentation
5. Principal's Address
6. Farewell Speech from C[illegible]s
7. Response from Graduates
8. Closing Remarks


The language used to write programs is called a programming language.
Also, writing a program is called programming.
And the person who writes programs is called a programmer.
No, not quite. It's a special notation for programs.
It has a fixed grammar and various commands are available.
Is it like the language of a "Programming Country," like Chinese or French?

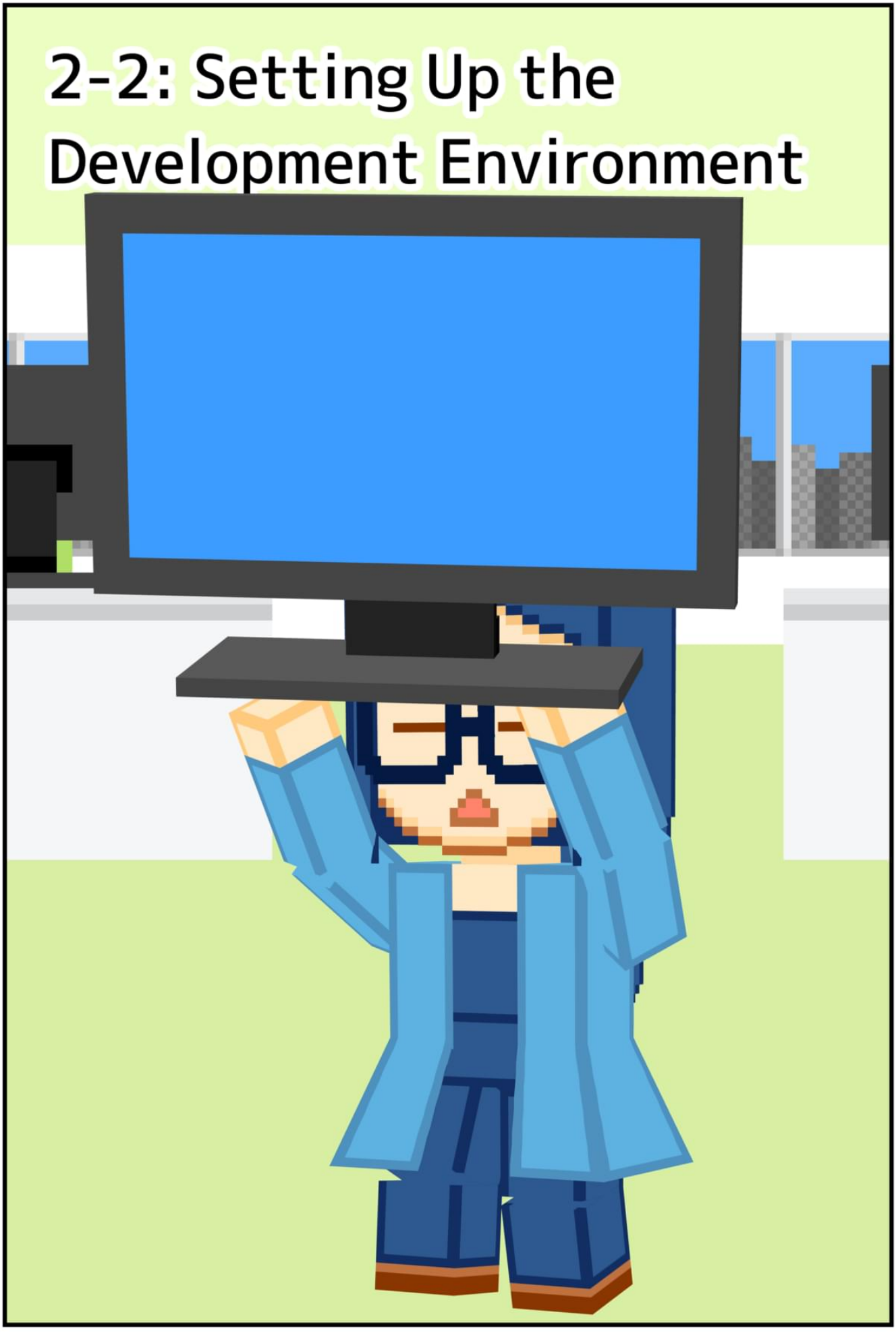

There are many kinds,
like Python, Java,
and JavaScript.
When writing a program,
the first step
is choosing a
programming language.
I've already
thought about that.
I'll teach you Python.
Please choose
the easiest one.

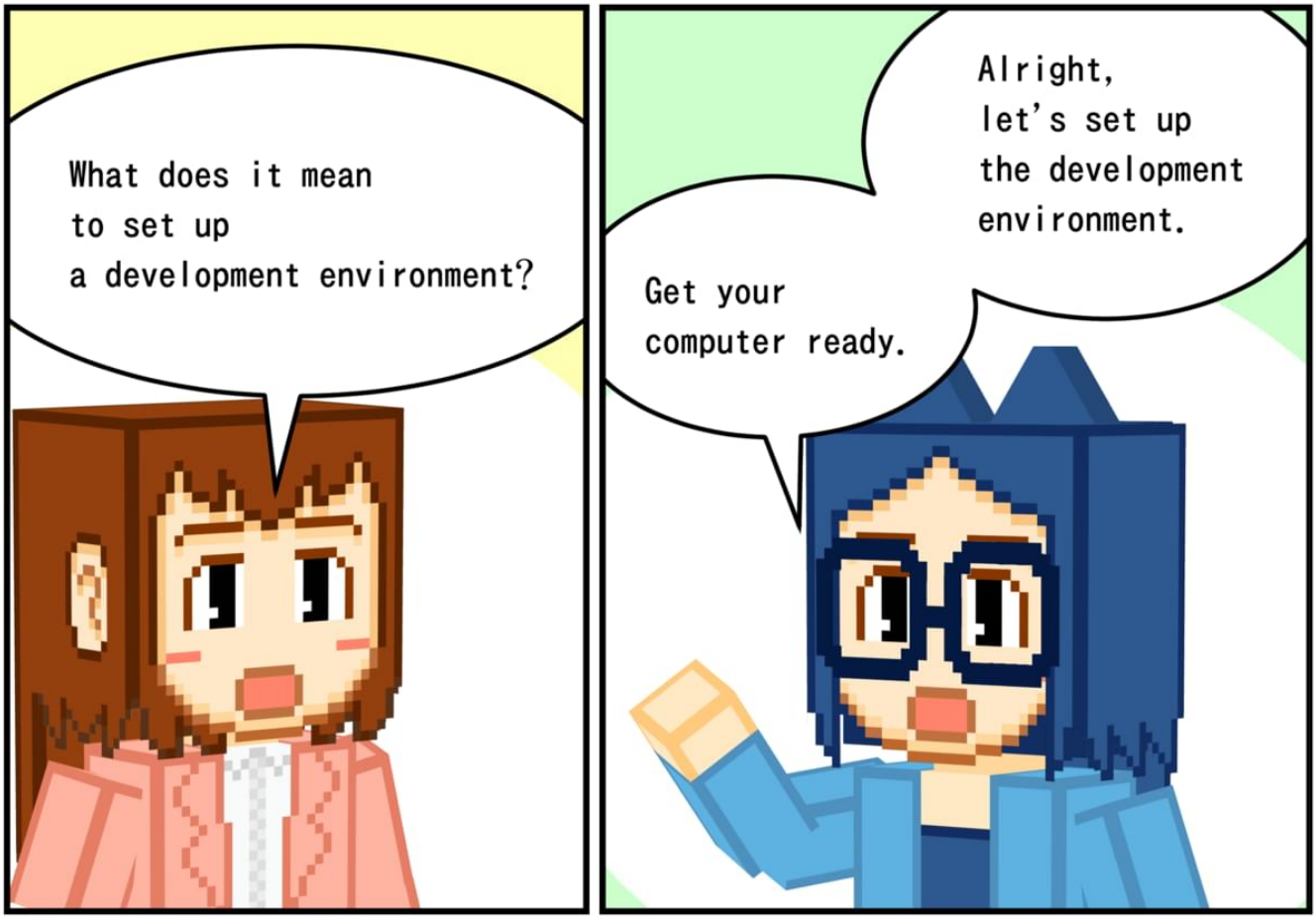

Here are some key points
So, what kind of language is Python?
- Simple and easy to read
- Popular
- Widely used
Yes, in programming, it's very important.
More information, frequent updates, and more people to ask for help.
Is popularity and usage really that important?
?


Next,
we'll set up
the development
environment.
I see!
That makes sense!
Meow
But first,
let's take a break.



2-2: Setting Up the Development Environment



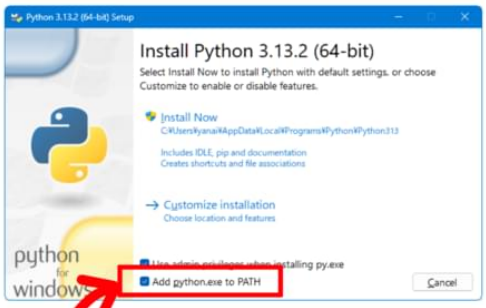

What does it mean
to set up
a development environment?
Alright,
let's set up
the development
environment.
Get your
computer ready.


I see!
You need to install
software to run
Python programs.
Also,
You need to install
an editor to
write programs.

Python Runtime Software

https://www.python.org/downloads/

Installation Dialog



For Windows, check this box.

Download and install each one from the URLs below.

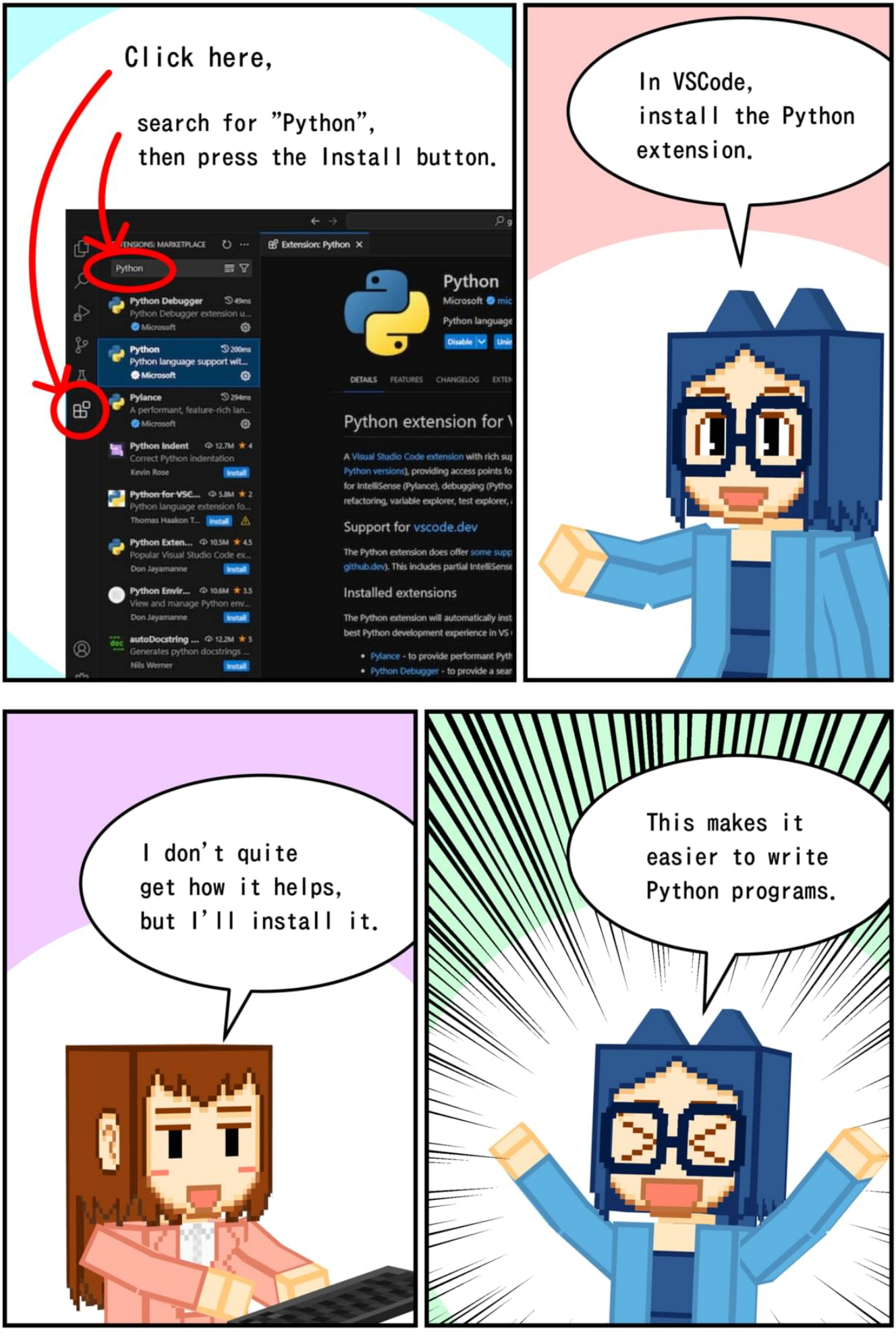
Is that all?

Just one more step.

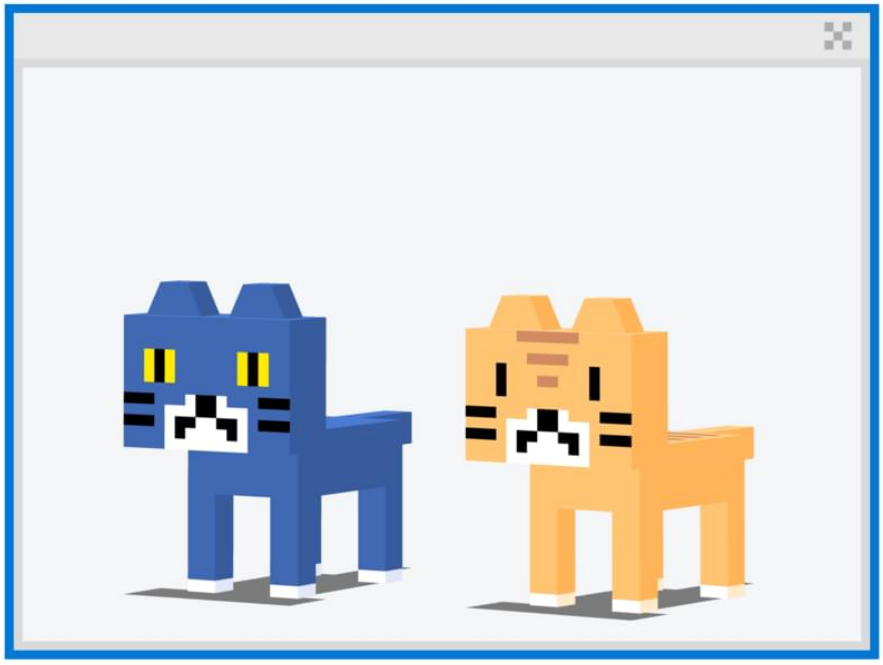
Programming Editor
Visual Studio Code

https://code.visualstudio.com/Download

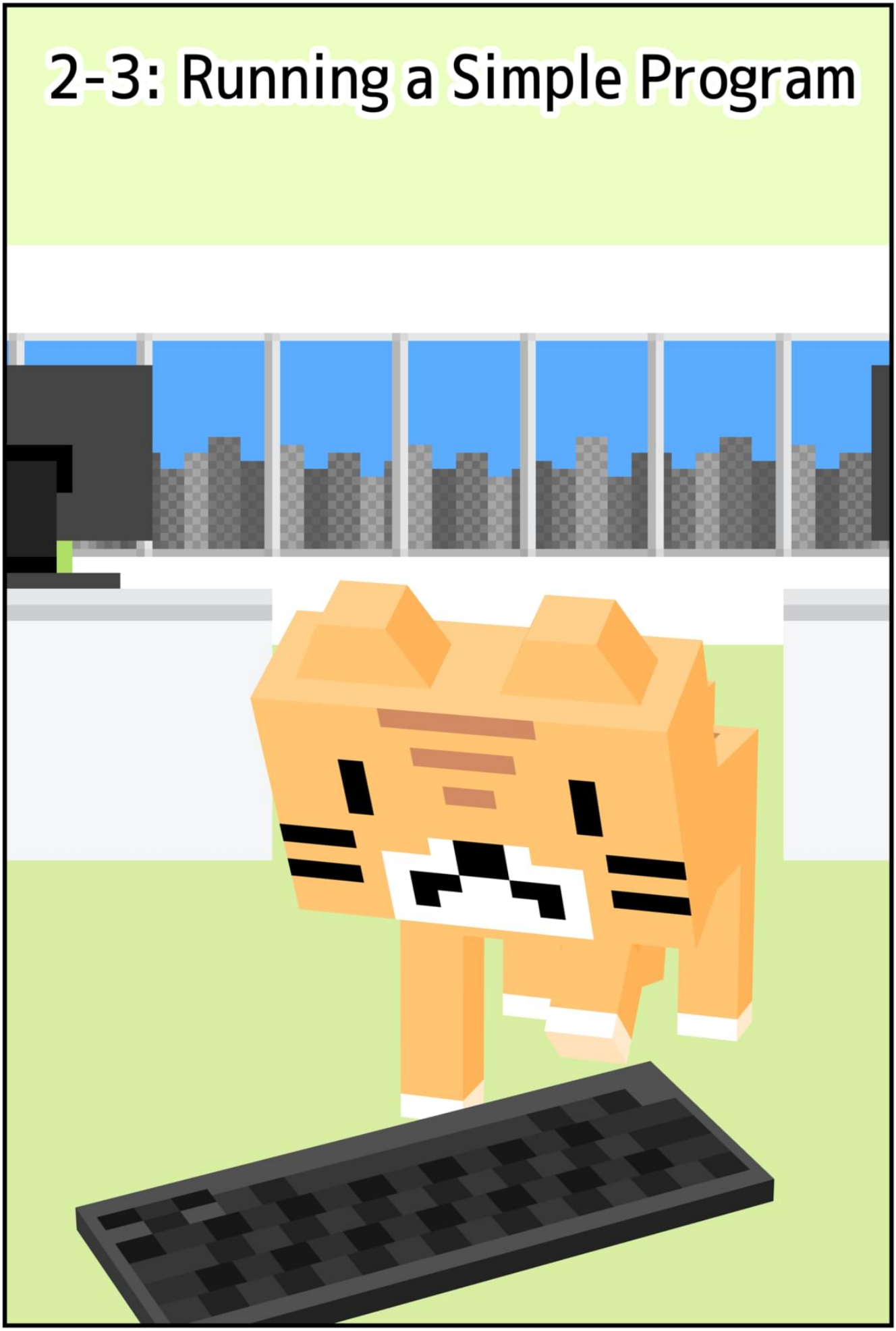
People often call it VSCode.

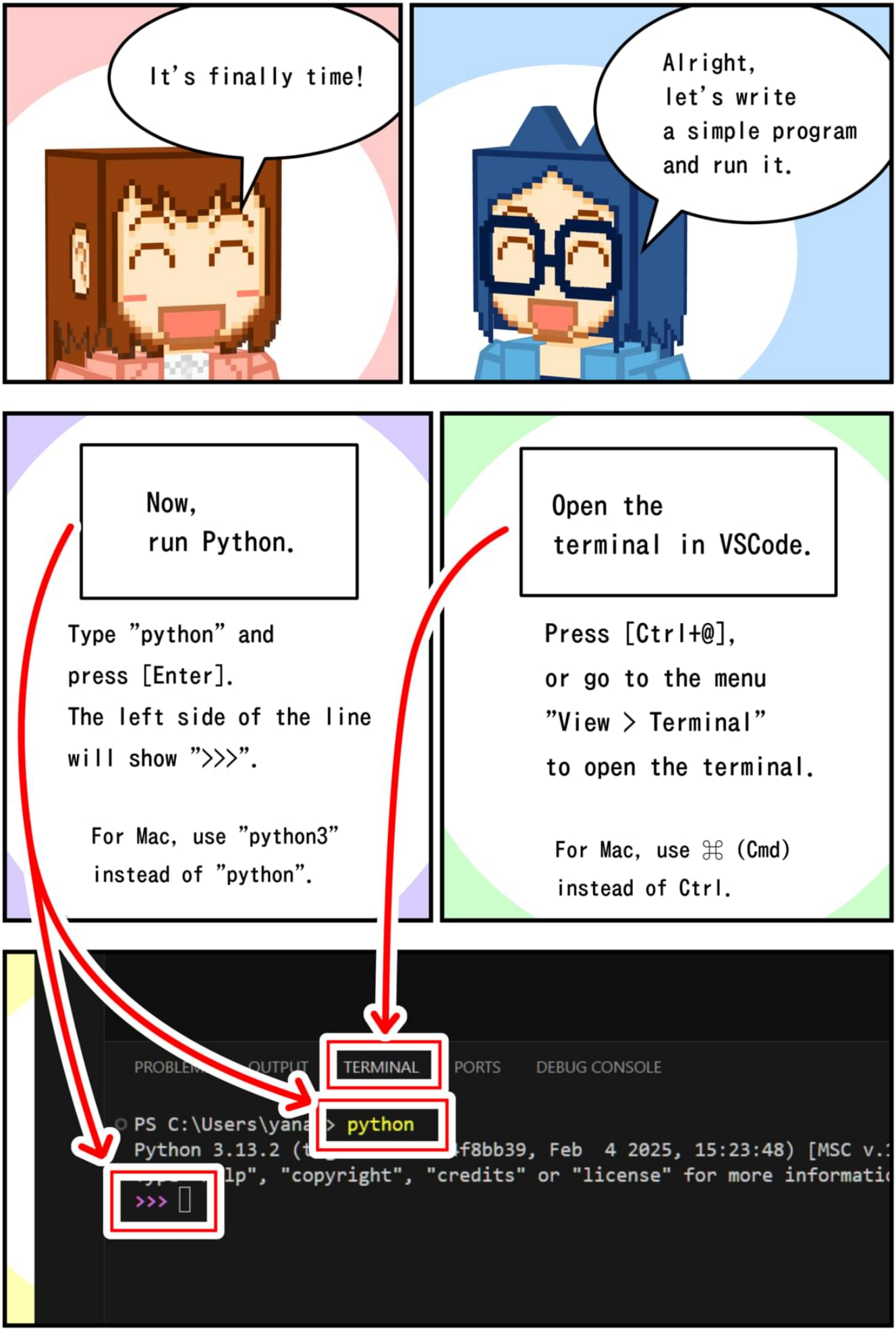

Click here,
search for "Python",
then press the Install button.
Python
Python Debugger
Python Debugger extension u...
Microsoft
Python
Python language support wit...
Microsoft
Pylance
A performant, feature-rich lan...
Microsoft
Python Indent
Correct Python indentation
Kevin Rose
Install
Python for VSC...
Python language extension fo...
Thomas Haakon T...
Python Exten...
Popular Visual Studio Code ex...
Don Jayamanne
Python Envir...
View and manage Python env...
Don Jayamanne
autoDocstring ...
Generates python docstrings ...
Nils Werner
Extension: Python
Python
Microsoft
Python language
Disable
DETAILS
FEATURES
CHANGELOG
Python extension for
Support for vscode.dev
Installed extensions
Pylance - to provide performant Pyth
Python Debugger - to provide a sear
In VSCode,
install the Python
extension.
I don't quite
get how it helps,
but I'll install it.
This makes it
easier to write
Python programs.


Gulp...
Now it's the real deal!
Now we have the basic development setup.
Next, we'll write a simple program.
Pet pet
No need to be nervous.
Pet a cat and relax.



2-3: Running a Simple Program



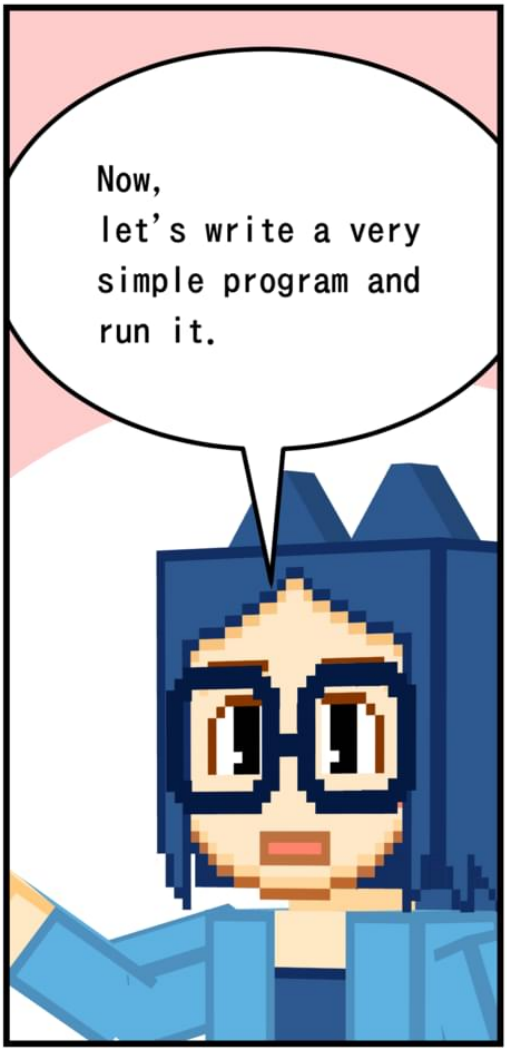

It's finally time!
Alright,
let's write
a simple program
and run it.
Now,
run Python.
Type "python" and
press [Enter].
The left side of the line
will show ">>>".
For Mac, use "python3"
instead of "python".
Open the
terminal in VSCode.
Press [Ctrl+@],
or go to the menu
"View > Terminal"
to open the terminal.
For Mac, use ⌘ (Cmd)
instead of Ctrl.
PROBLEMS
OUTPUT
TERMINAL
PORTS
DEBUG CONSOLE
PS C:\Users\yana > python
Python 3.13.2 (t
f8bb39, Feb 4 2025, 15:23:48) [MSC v.
lp", "copyright", "credits" or "license" for more informati
>>>

Type "1+2" and press [Enter].

```
>>> 1+2
3
```

The result, "3",
will be displayed.

Press [Ctrl+Z] to exit.




This is great for programs that are about 1-3 lines long.
REPL
Read-Eval-Print Loop
What you just did is called REPL.
It's an interactive way to run programs.
I see.
For longer programs, you write them in a file and run it.
I'll explain that now.

Create a "test" folder,
and drag & drop it into
the VSCode window.

VSCode

test

Here's how it goes.

First, create a folder
for your programs
and open it in VSCode.

Open the terminal,
type the command,
press [Enter] to run it.

Result

Command

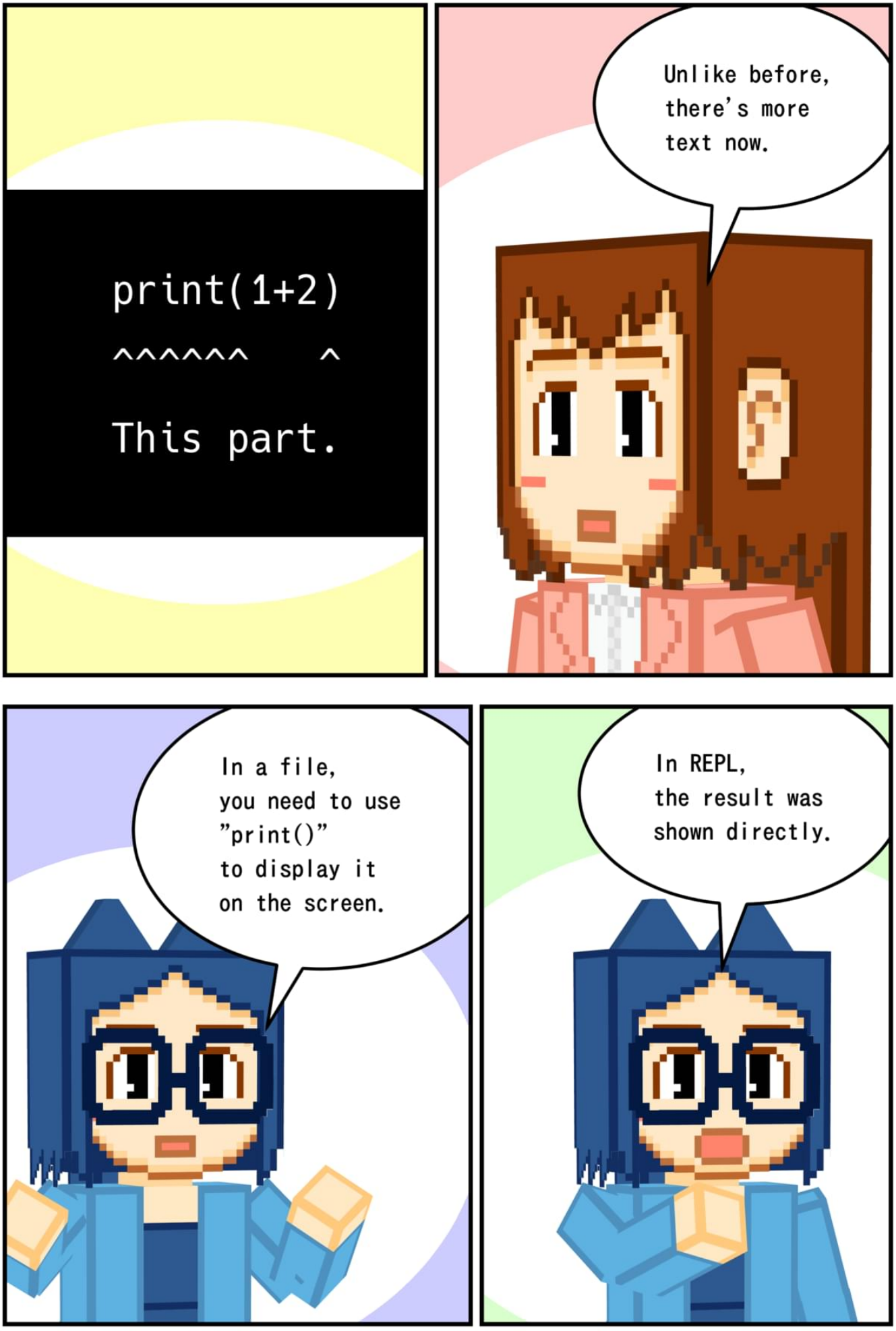
```
python test.py
3
```

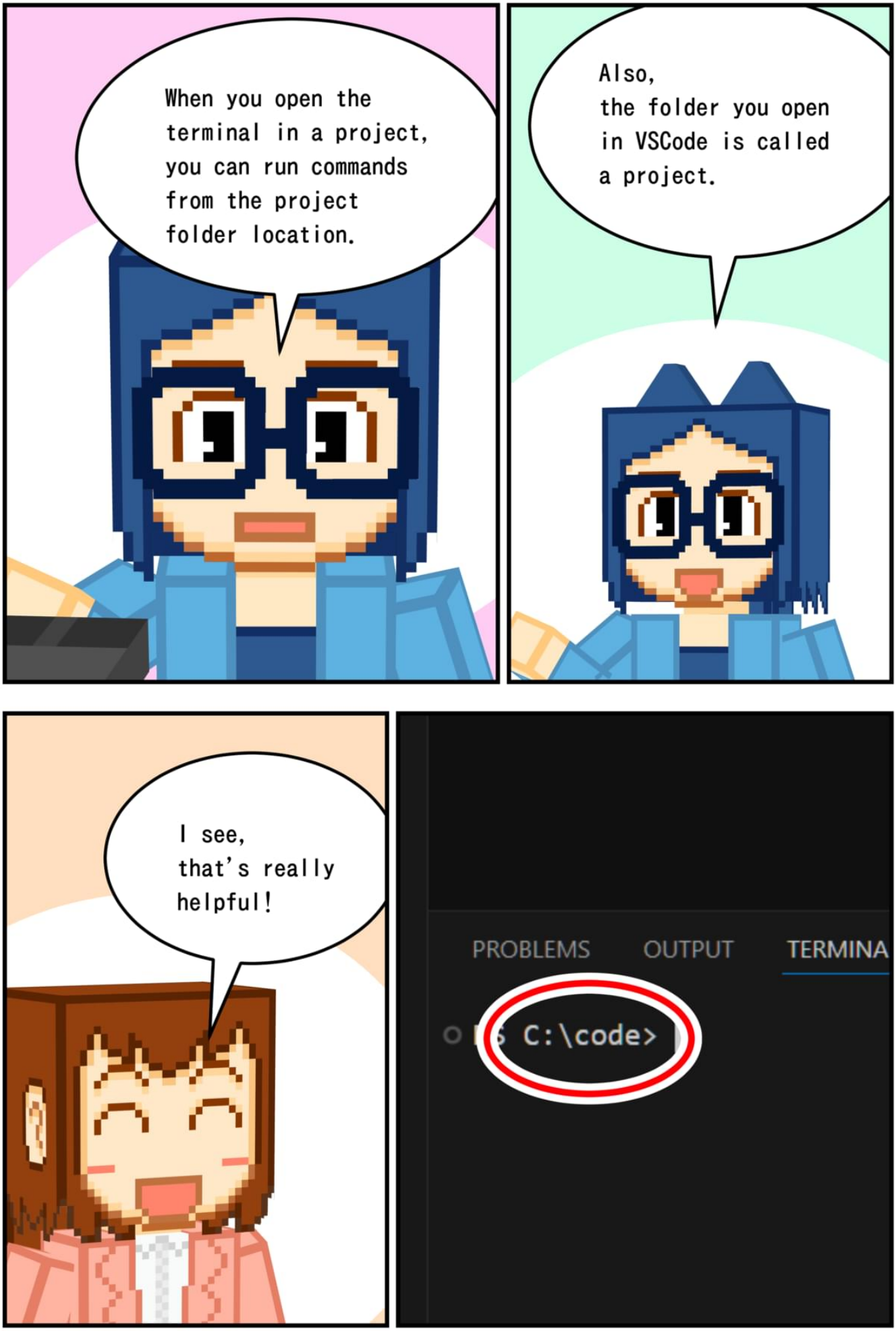
Inside the folder,
create a file,
open it in VSCode,
write the program,
and save it.

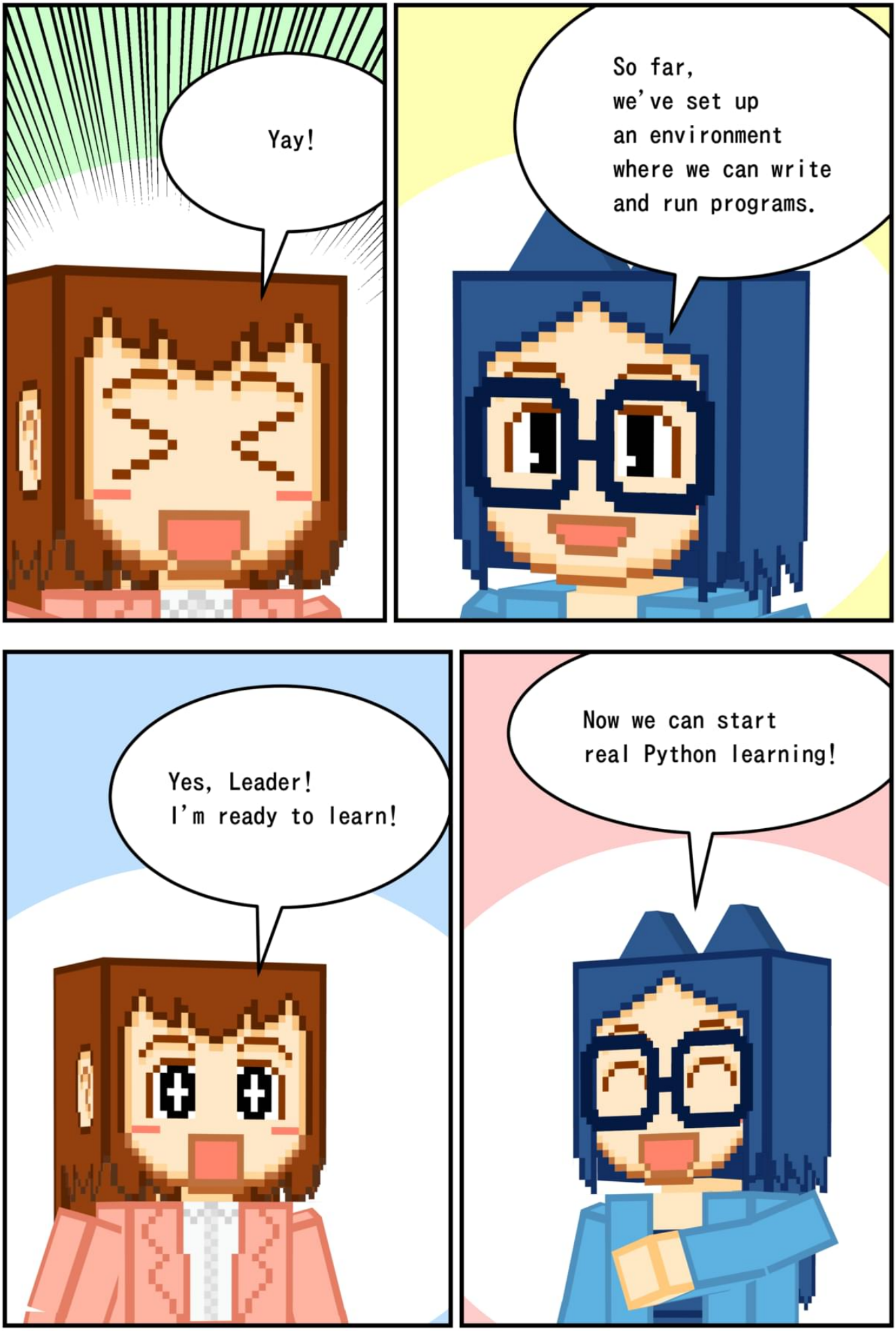
The file extension should
be .py,
so let's name it "test.py".

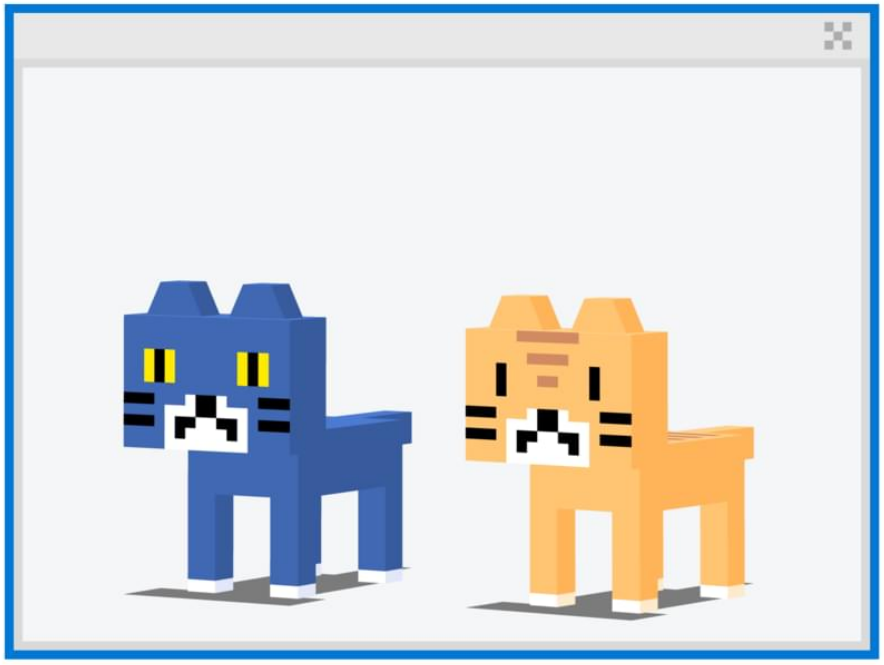
Inside "test.py"

```
print(1+2)
```


print(1+2)
^^^^^^ ^
This part.
Unlike before, there's more text now.
In a file, you need to use "print()" to display it on the screen.
In REPL, the result was shown directly.


When you open the terminal in a project, you can run commands from the project folder location.
Also, the folder you open in VSCode is called a project.
I see, that's really helpful!
PROBLEMS
OUTPUT
TERMINA
C:\code>


Yay!
So far,
we've set up
an environment
where we can write
and run programs.
Yes, Leader!
I'm ready to learn!
Now we can start
real Python learning!

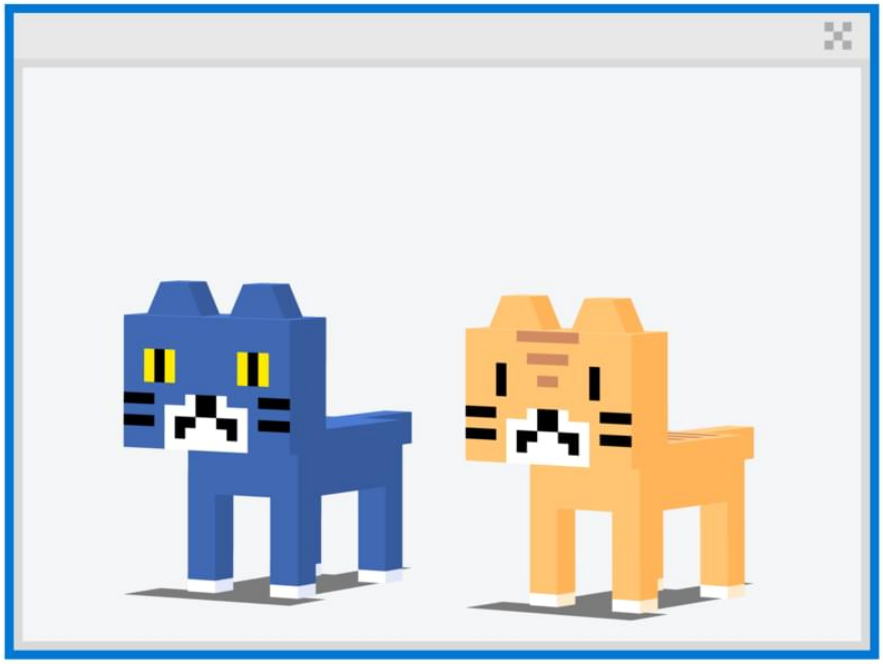


2-4: Future Direction


First, we need to grasp the big picture.
Now that we are ready, I will explain the study plan.
We won't cover all the specifications.
It would be too much for a beginner.
The goal is to be able to read and write simple programs.
Trying to learn everything is endless.
Yeah. Here is the plan.

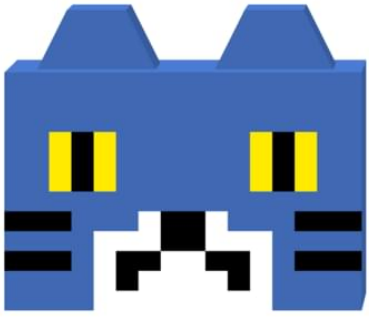

Probably even people who write programs for work don't know everything.
Is there really that much to learn?
They look things up as needed.
But to do that, you still need basic knowledge.
Then how do programmers manage?


Yes!
I'll do my best
to be useful!
So, it's going
to be tough,
but stick with me.
Am I really going
to be okay...?
Hmm...
I wonder when
that will be...

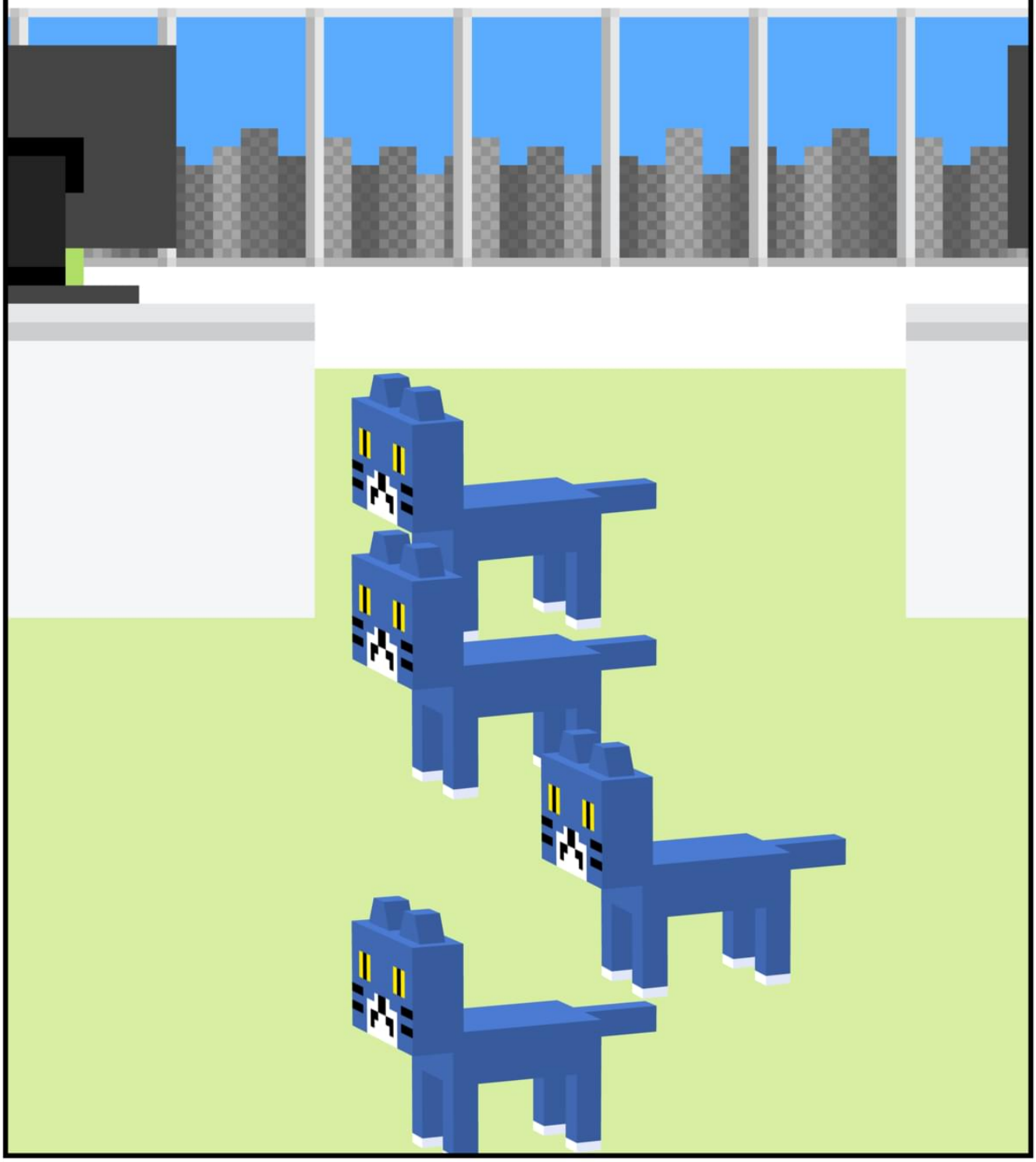


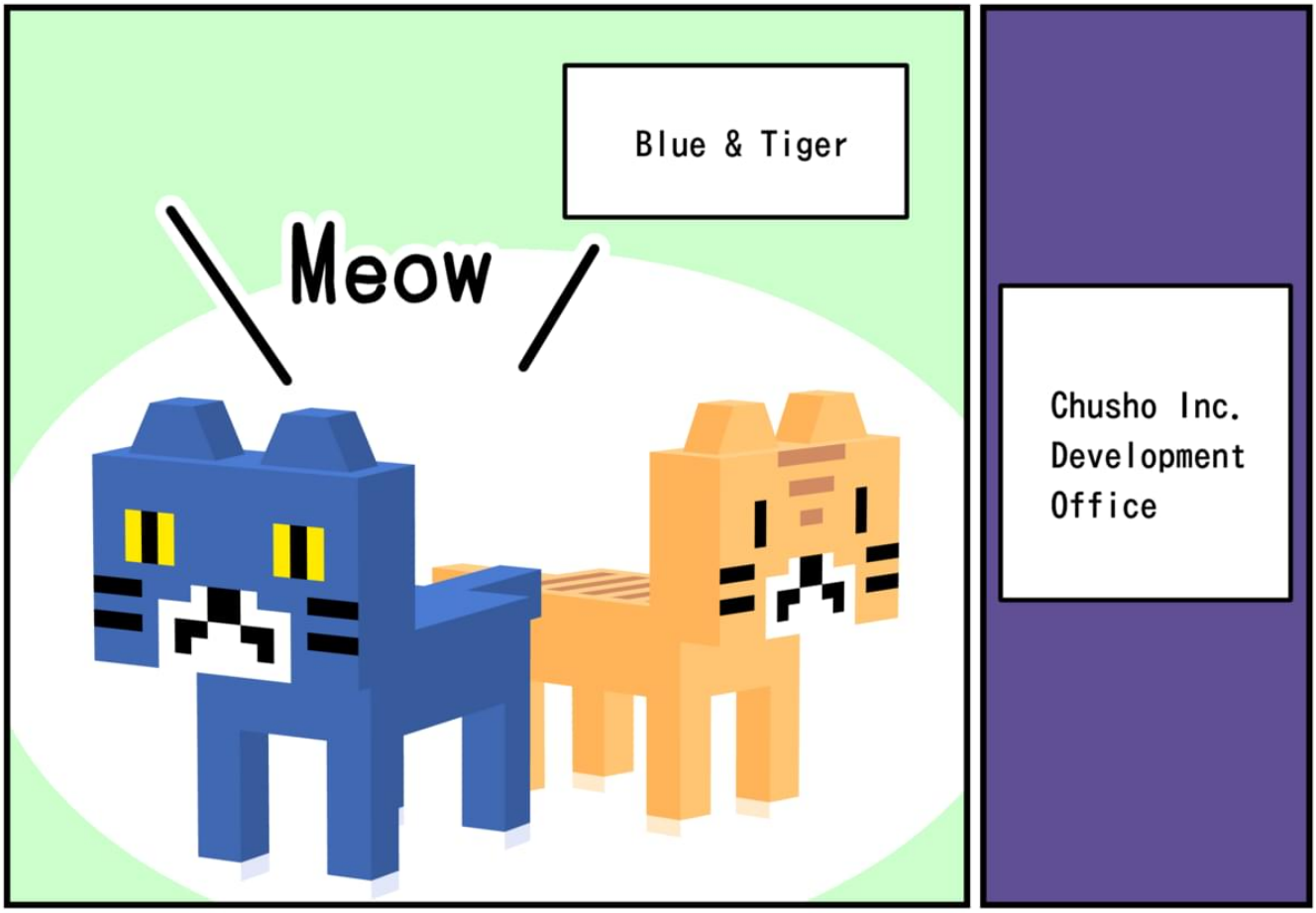
Chapter 3

Learning the Fundamentals



3-1: Execution Order and Indentation




Blue & Tiger
Meow
Chusho Inc.
Development
Office


New Employee:
Niimi
Development Manager:
Nekono


This is the very foundation, right?
From this point, I will teach you the basics ofreading Python programs.
- Execution order
- Indentation
Yes. Here is a summary of today's topics:
I have narrowed down the content quite a bit for this lesson.
Since it's just two topics, I think I can handle it.

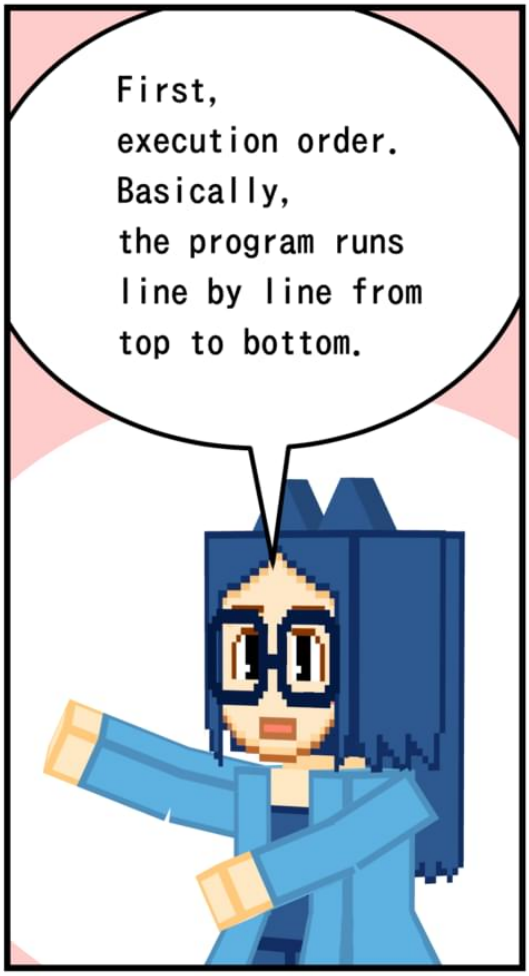
Line 1: Execute this

Line 2: Execute this

Line 3: Execute this

Line 4: Execute this

⋮

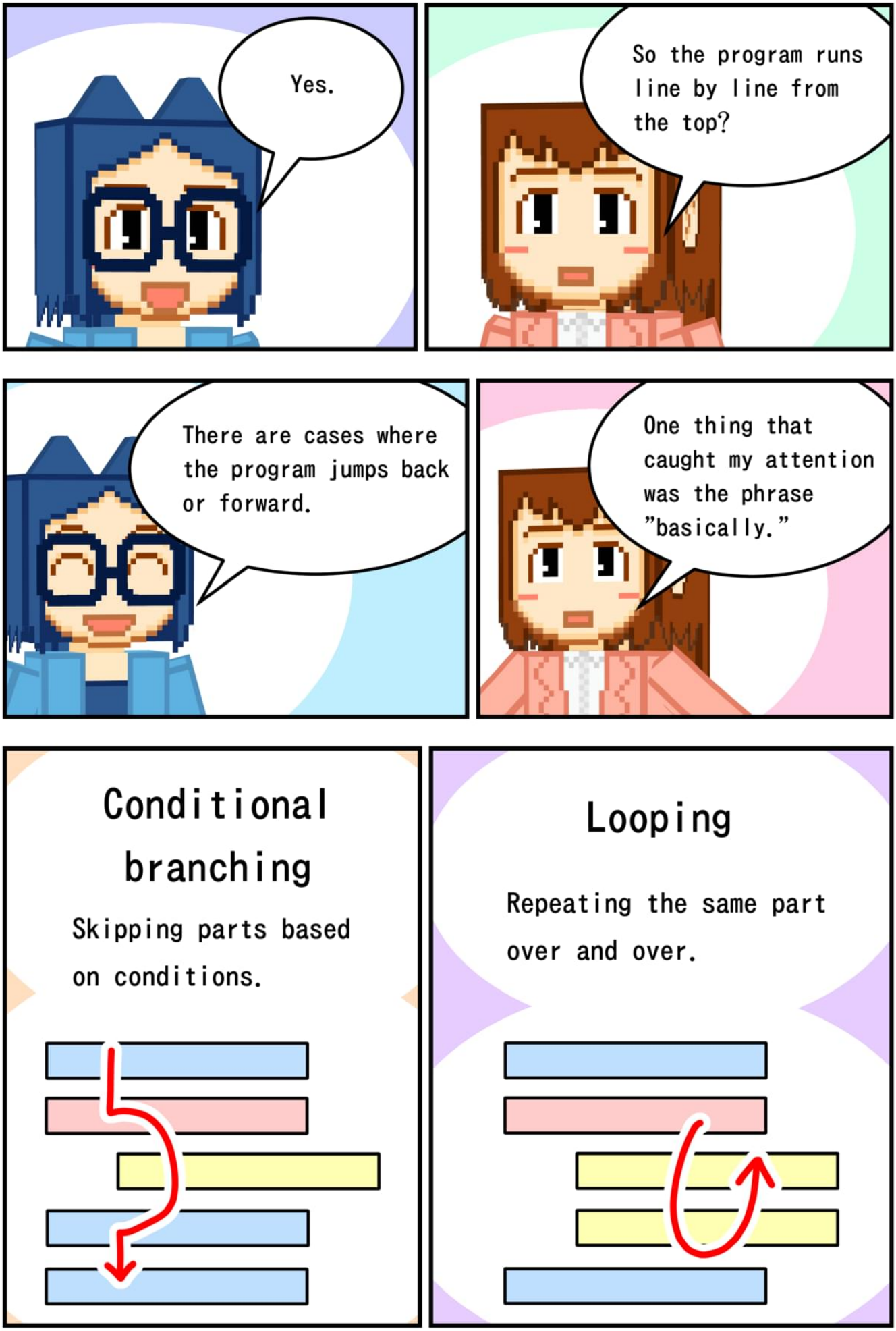


1: Take three steps forward

2: Turn to the right

3: Take two steps forward

4: Bow


Yes.
So the program runs line by line from the top?
There are cases where the program jumps back or forward.
One thing that caught my attention was the phrase "basically."
Conditional branching
Skipping parts based on conditions.
Looping
Repeating the same part over and over.

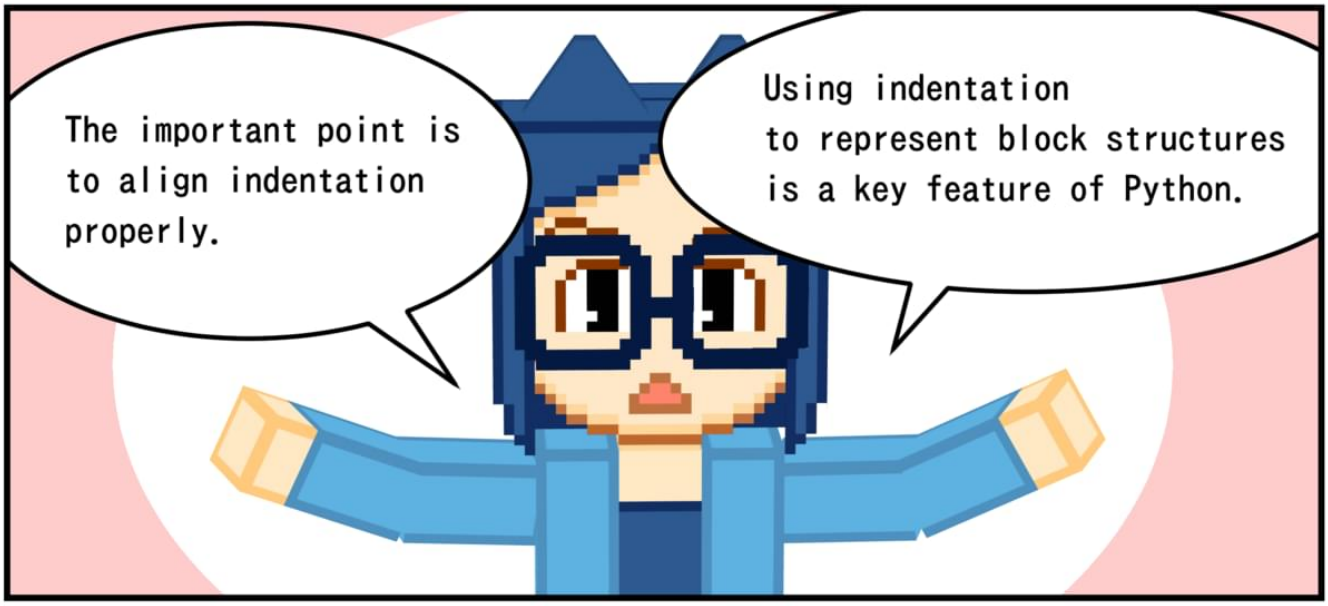

Indentation is done using four spaces.
Next is indentation.
In Python, code blocks are represented by indentation.
[Tab]
Indent
[Shift+Tab]
Remove indent
In VSCode, pressing the Tab key indents by four spaces.

Indent with four spaces.

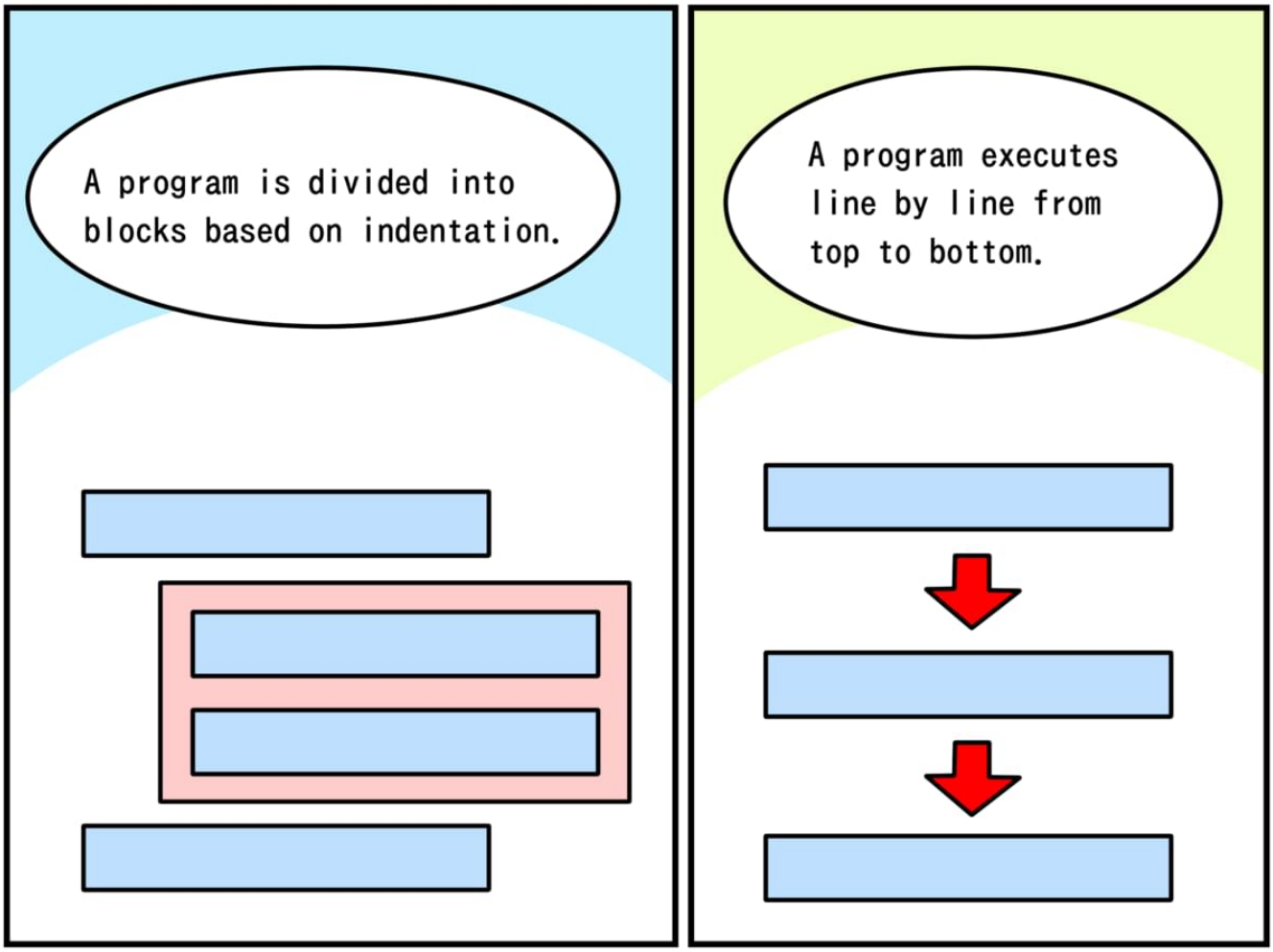
```
program
program
    program
    program
        program
        program
    program
    program
program
program
    program

    program
```

Each indentation level represents a single block.

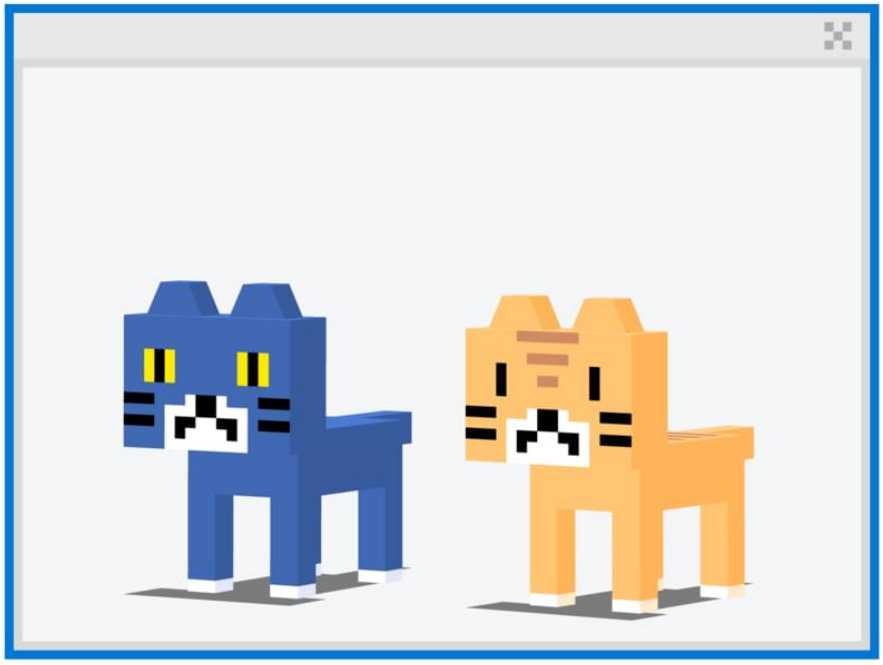
Even with blank lines in between, it's still the same block.



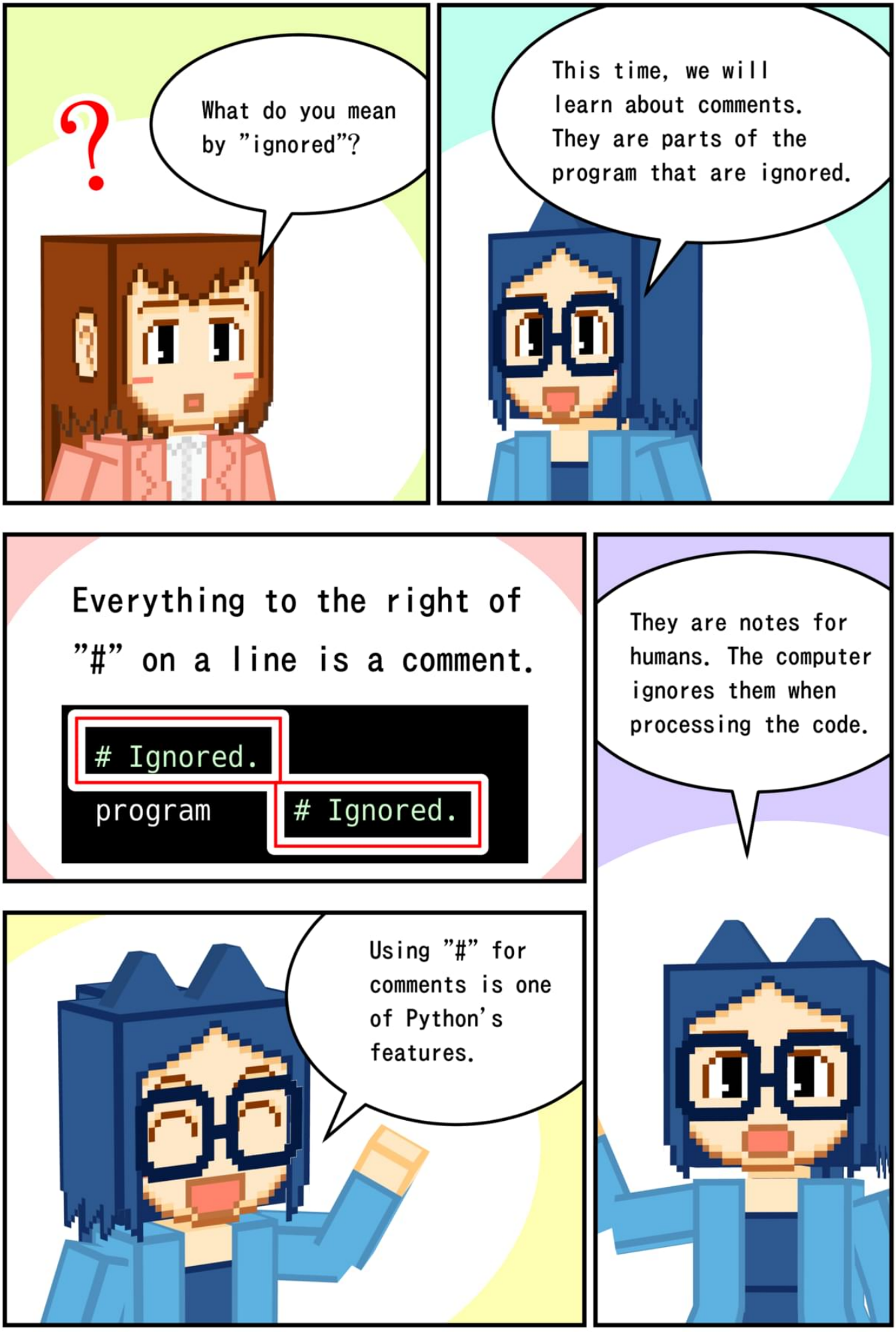

All right.
Now, explain to me what you have learned so far.

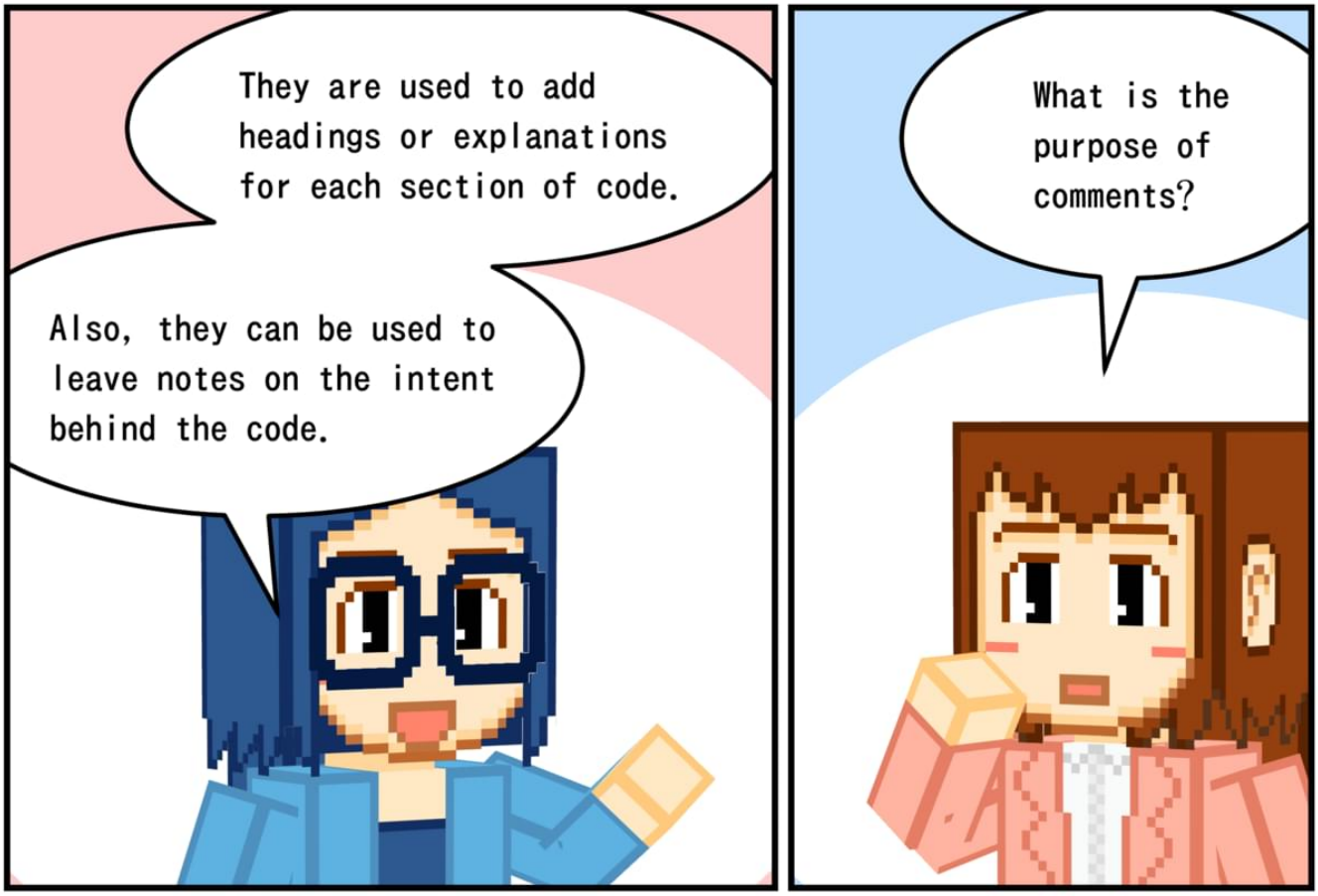

A program is divided into blocks based on indentation.
A program executes line by line from top to bottom.

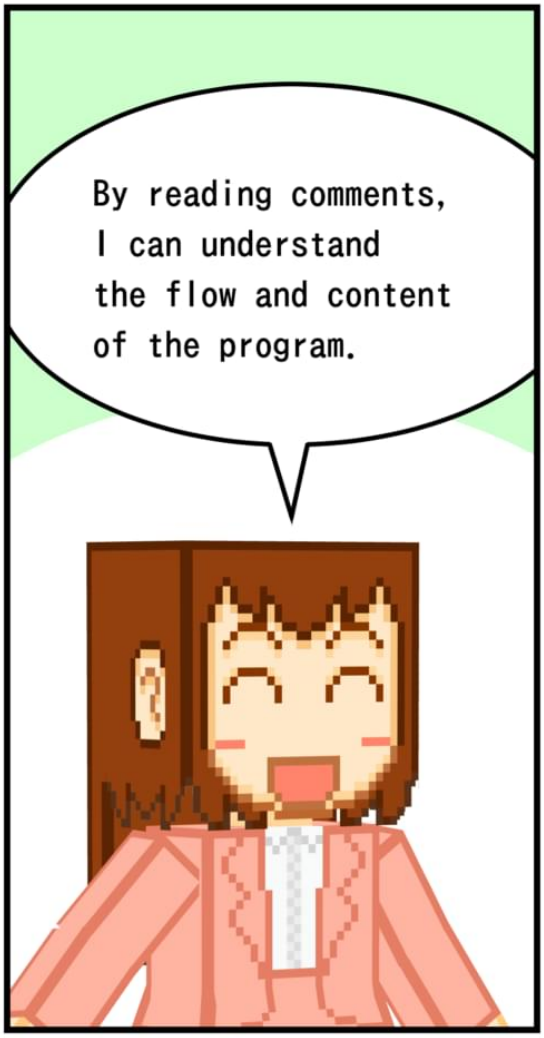

Got it!
Good.
Now you understand the very basics of reading programs.
Next, we will look at the details.
Meow
You're working hard, Niimi.



3-2: Comments


What do you mean by "ignored"?
This time, we will learn about comments. They are parts of the program that are ignored.
Everything to the right of "#" on a line is a comment.
Ignored.
program
Ignored.
They are notes for humans. The computer ignores them when processing the code.
Using "#" for comments is one of Python's features.

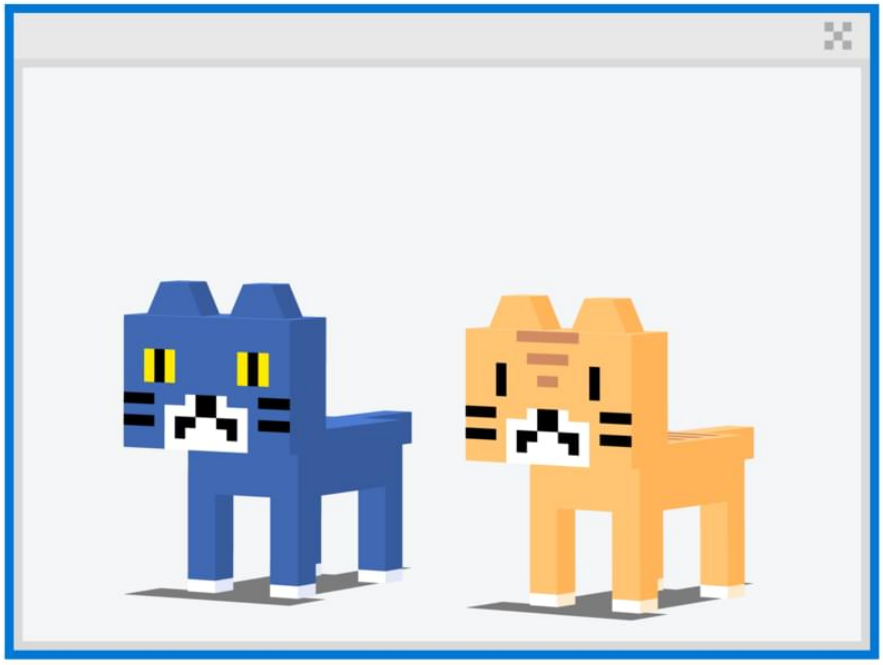


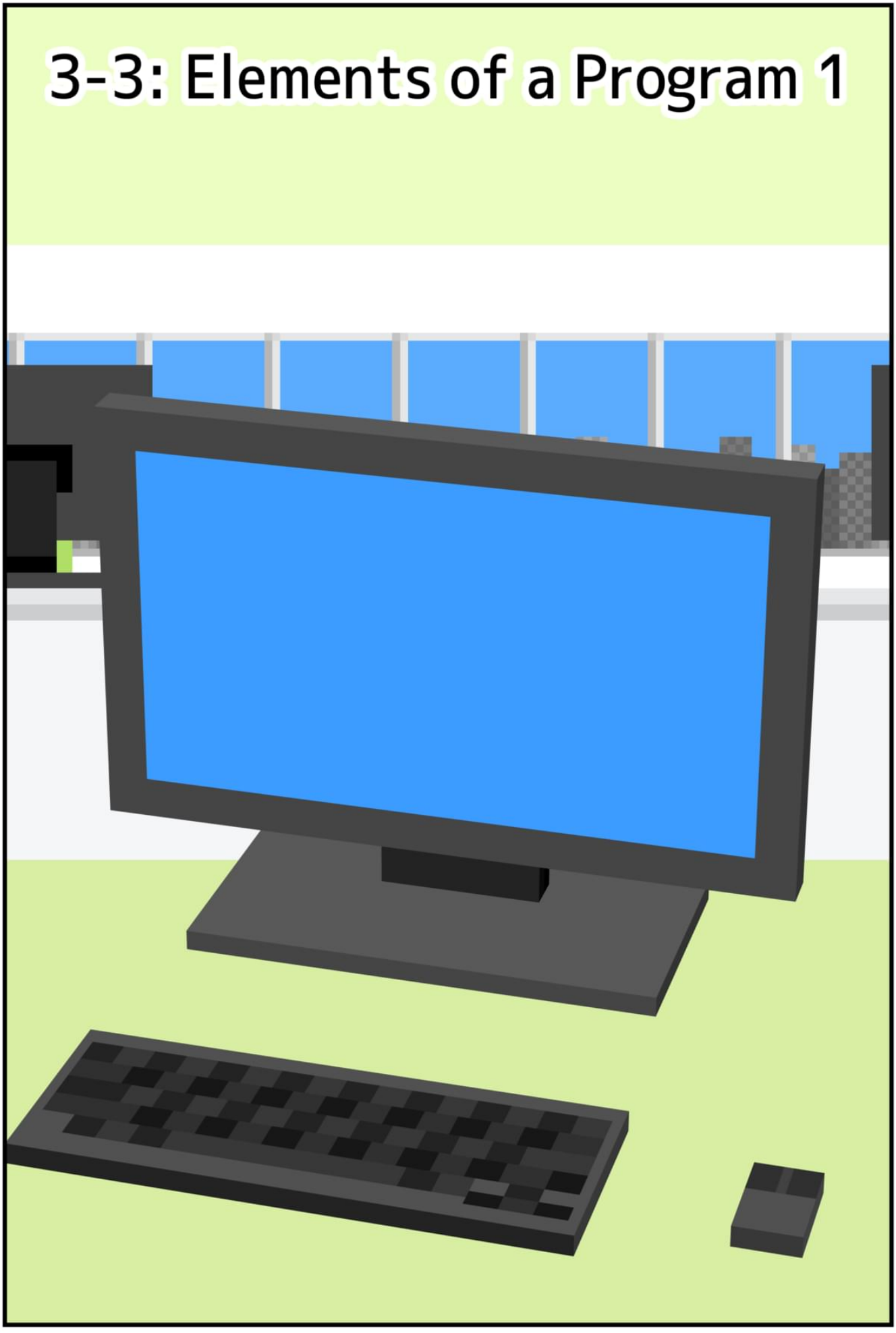



```
# Displaying an image
.....................
.............
    # Reading a file
    .....................
    .............
    # Displaying on screen
    # Shrink if larger
    # than the window
    .....................
    .............
```


They should make the logic and intent clear at a glance for others.
"Others" also includes your future self.
It is best to write plenty of comments.
In about six months, you will forget what you wrote.
Spoken from experience.
My future self?
?

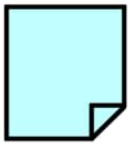

It is like writing word meanings below sentences when learning a new language.
For beginners, it is also helpful to write detailed explanations of what they researched or how the code works.
Exactly. So from now on, make good use of comments in your learning.
I see. At first, I write a lot of notes.




3-3: Elements of a Program 1


We have learned about execution order, code blocks, and ignored parts...
But we haven't actually learned how to write a program yet, right?
By the way...
Hehe.
Good observation. Noticing that yourself is a great sign.

First, create a text file.



main.py

This time,
we will write
a short program
and break it down
into its elements.

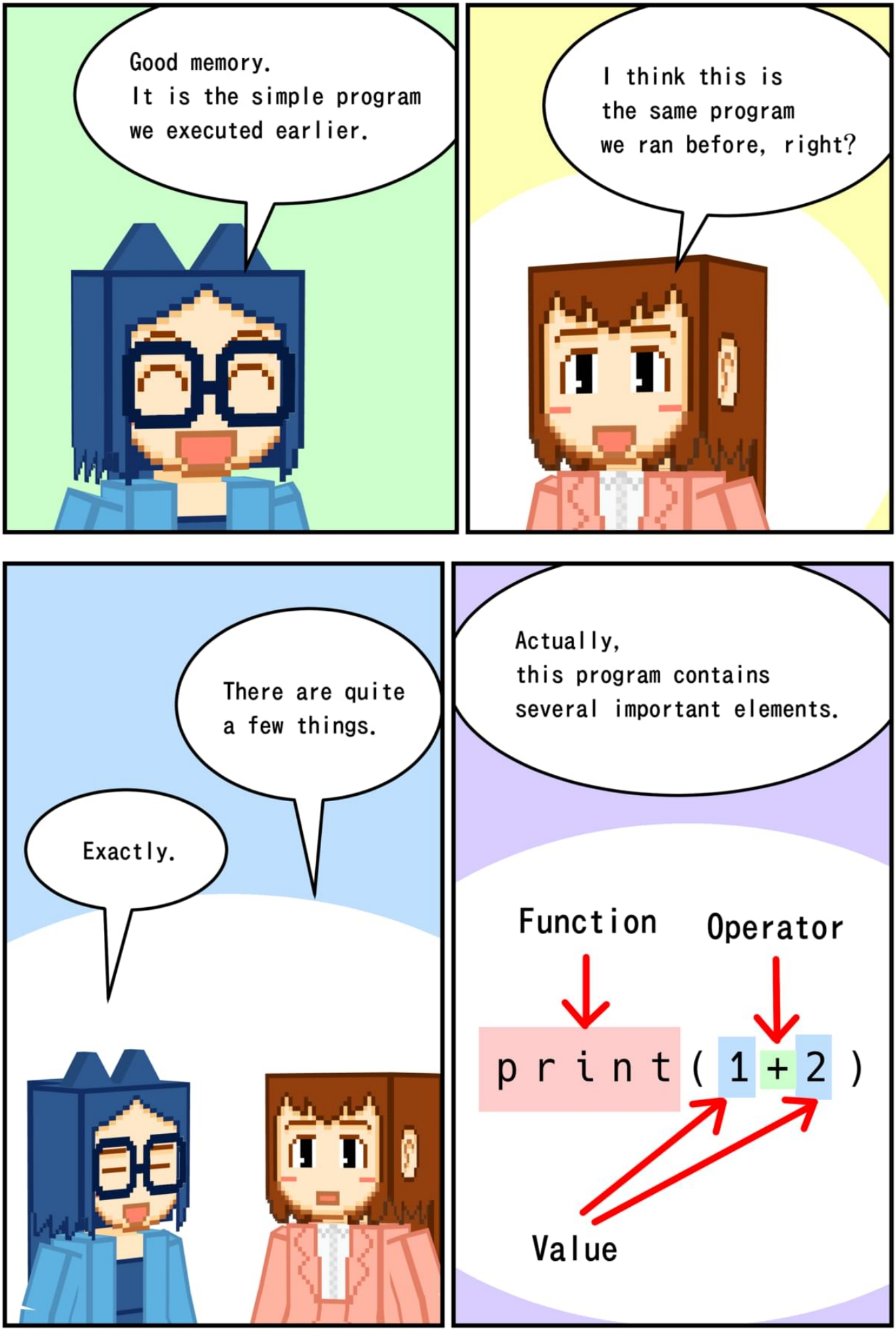
Then, write a short
program in that file.

```
print(1+2)
```

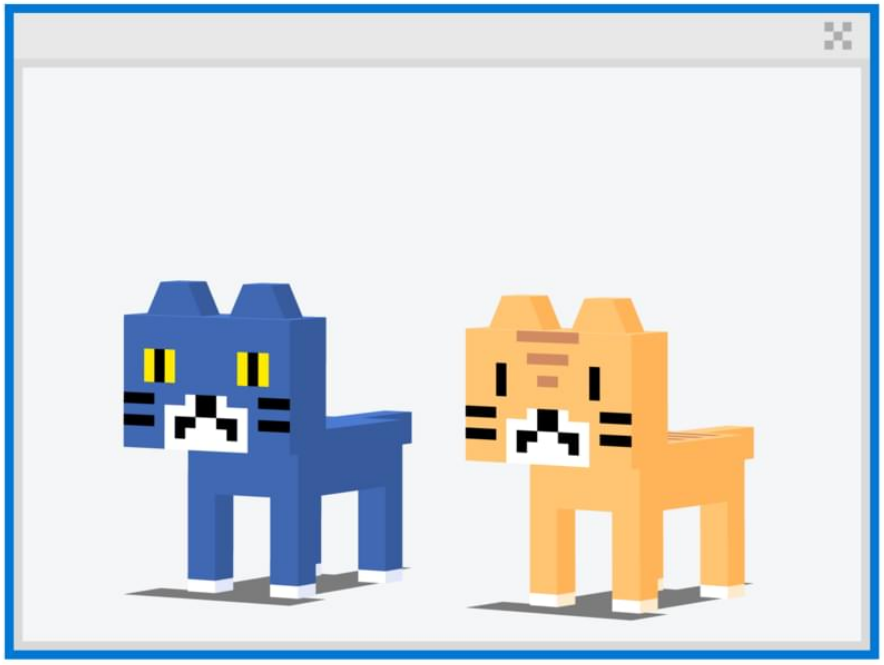
1. Open the folder containing main.py in VSCode.
2. Open the terminal with Ctrl+@.
3. Run the following command:
   python test.py
4. View the output.

```
> 3
```

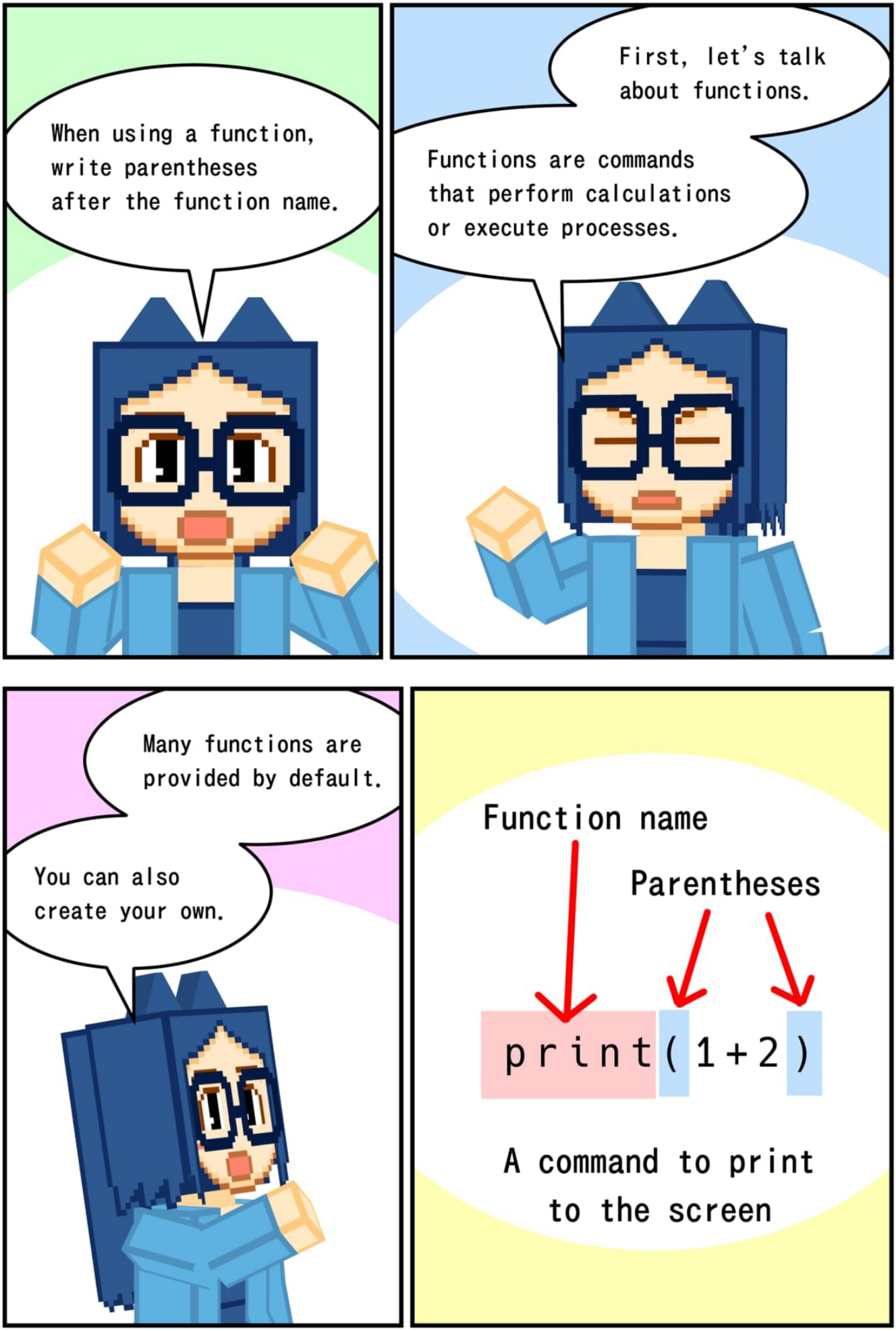
Finally,
run it as
a Python program.


Good memory.
It is the simple program we executed earlier.
I think this is the same program we ran before, right?
There are quite a few things.
Exactly.
Actually, this program contains several important elements.
Function
Operator
print (1 + 2)
Value


These are things I've never heard of in my life.
Starting next time, we will explain functions, values, and operators.
I'll do my best to keep up!
Programming concepts are a bit different from the real world.



3-4: Functions


When using a function, write parentheses after the function name.
First, let's talk about functions.
Functions are commands that perform calculations or execute processes.
Many functions are provided by default.
You can also create your own.
Function name
Parentheses
print(1+2)
A command to print to the screen

One argument

```
print(1)
```

No arguments

```
print()
```

Some functions allow you to write values or expressions inside the parentheses.

These are called arguments and provide information to the function.

Each function has a fixed number of arguments it can accept.

Multiple arguments separated by commas

```
print(1, 2, 3)
```

Example:
A function that returns
the largest value
among the arguments.

```
max(1, 2, 3)
```

↓

```
3
```

Also, many functions
return a value
as a result of their
calculation or process.

This returned value is
called the return value.

```
print(max(1, 2, 3))
```

↓

```
max(1, 2, 3)
```
returns
```
3
```

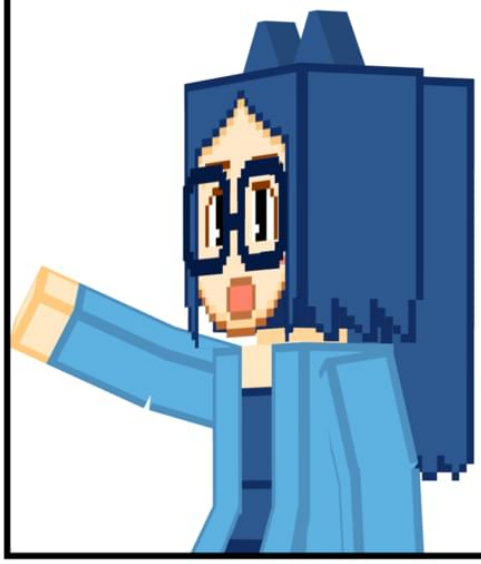
↓

Then, it becomes
```
print(3)
```

↓

"3" is displayed

When combining
functions,
it works like this.



1. Open a folder in VSCode.
2. Create "main.py" in the folder.
3. Write the program in "main.py".

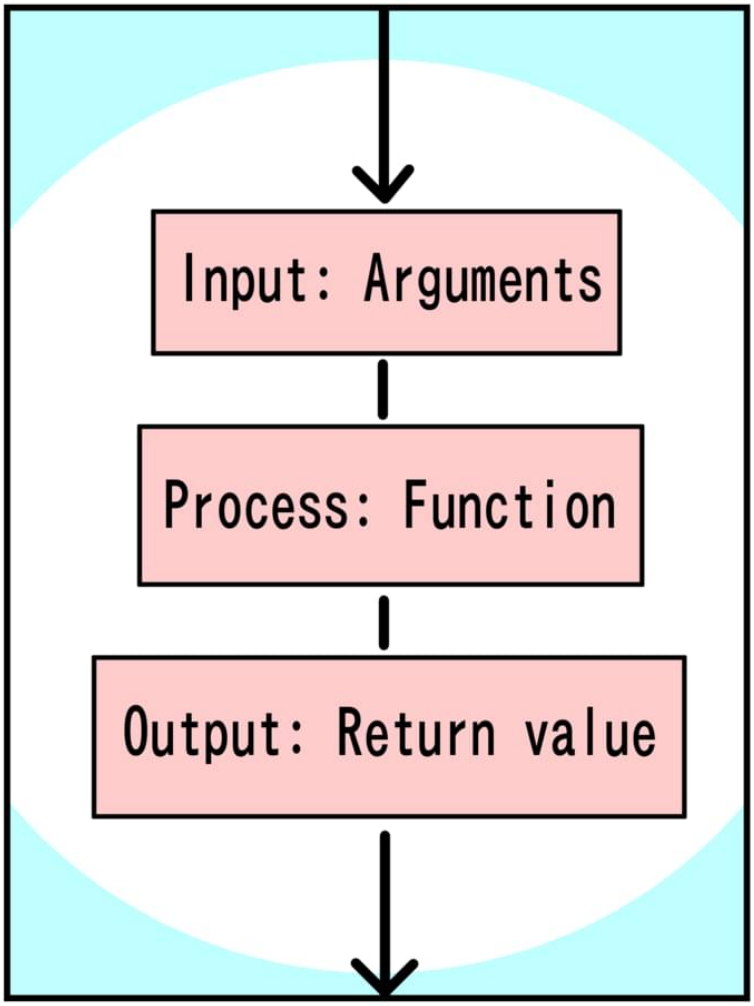
main.py

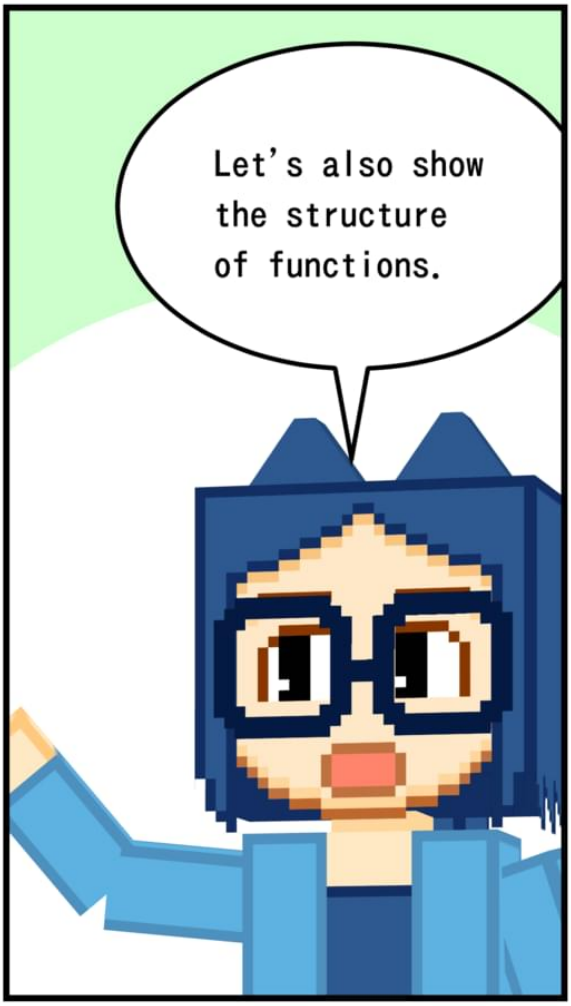
```
print(max(1, 2, 3))
```

4. Open the terminal with Ctrl+@.
5. Run "python main.py".
6. "3" is displayed in the terminal.

Enter the command and execute it.

```
python main.py
> 3 <-- displayed
```

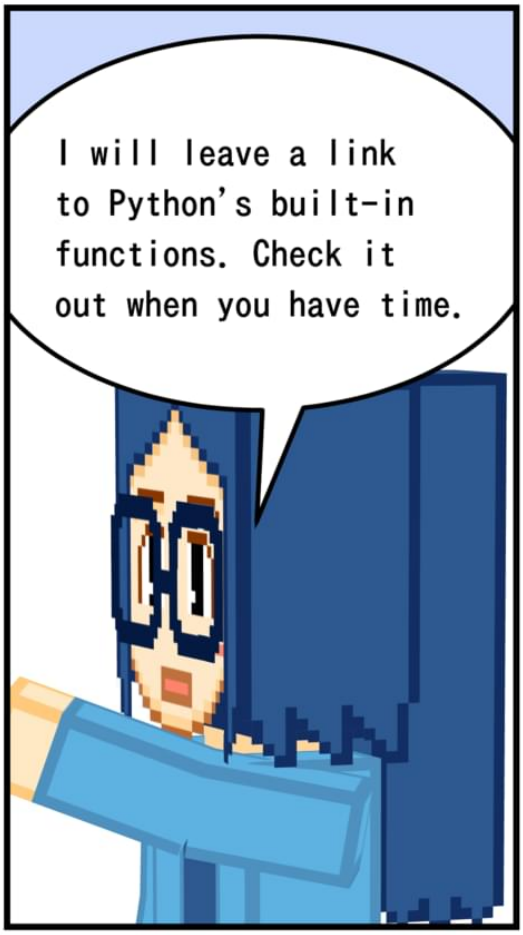


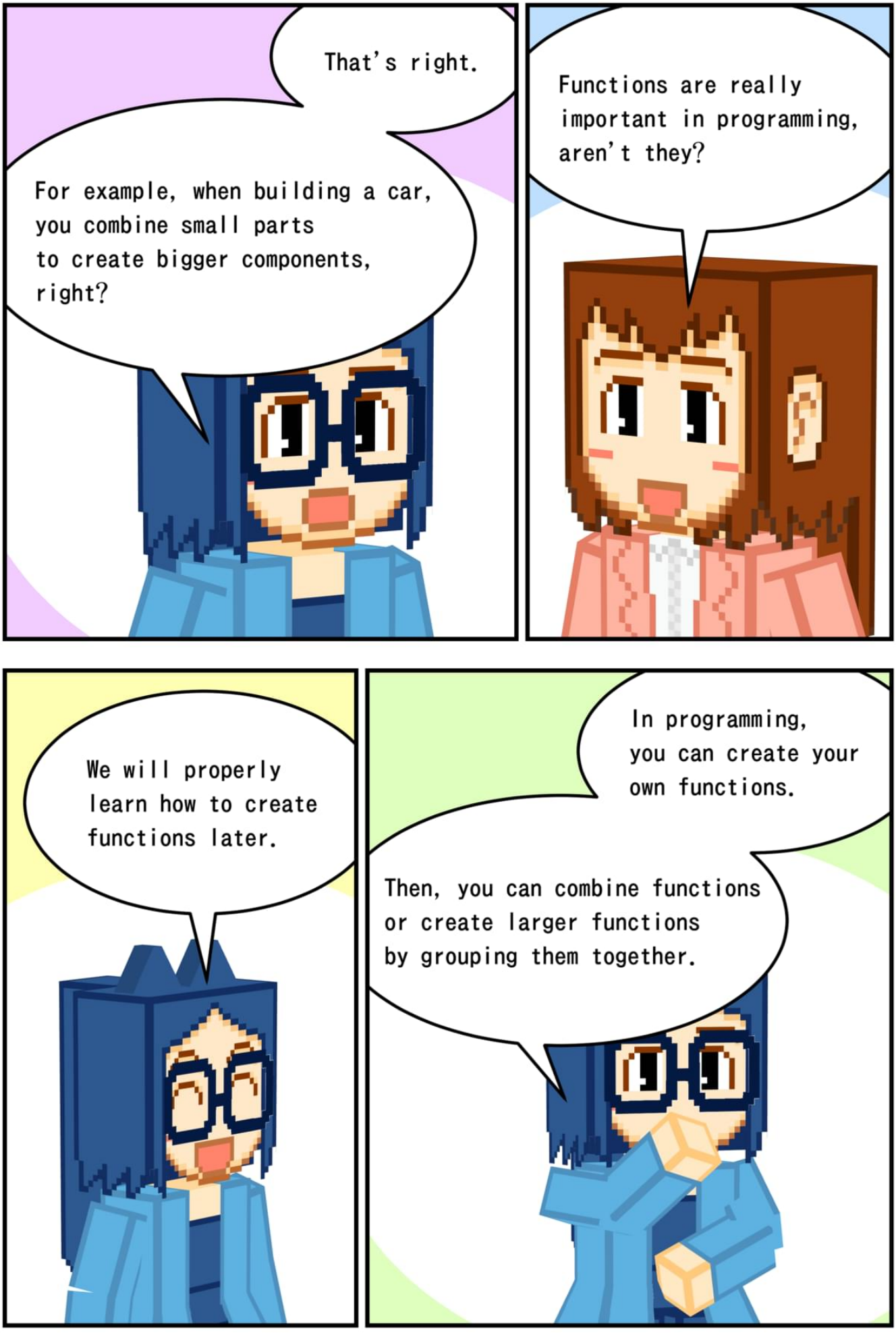


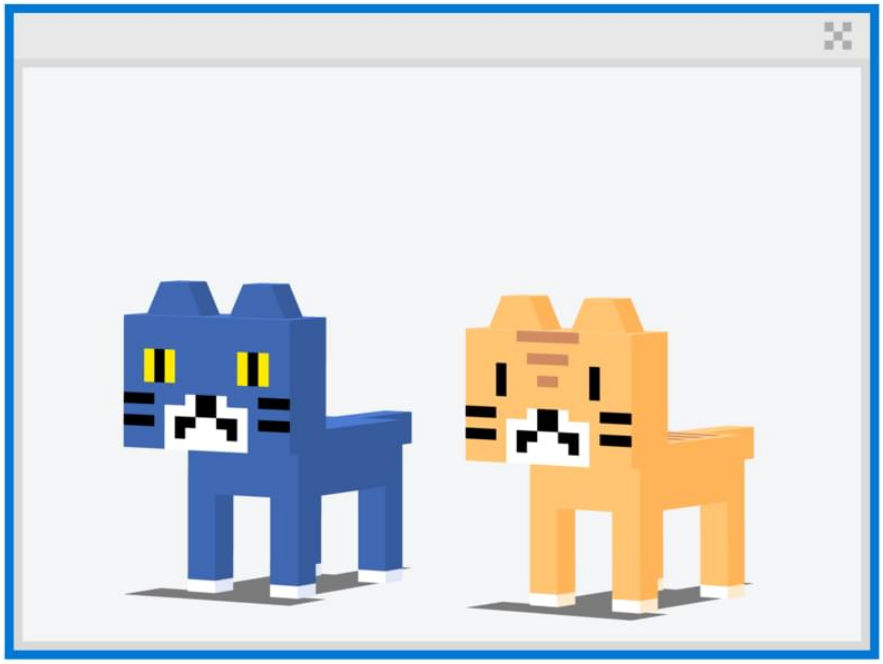
Built-in Functions
— Python documentation

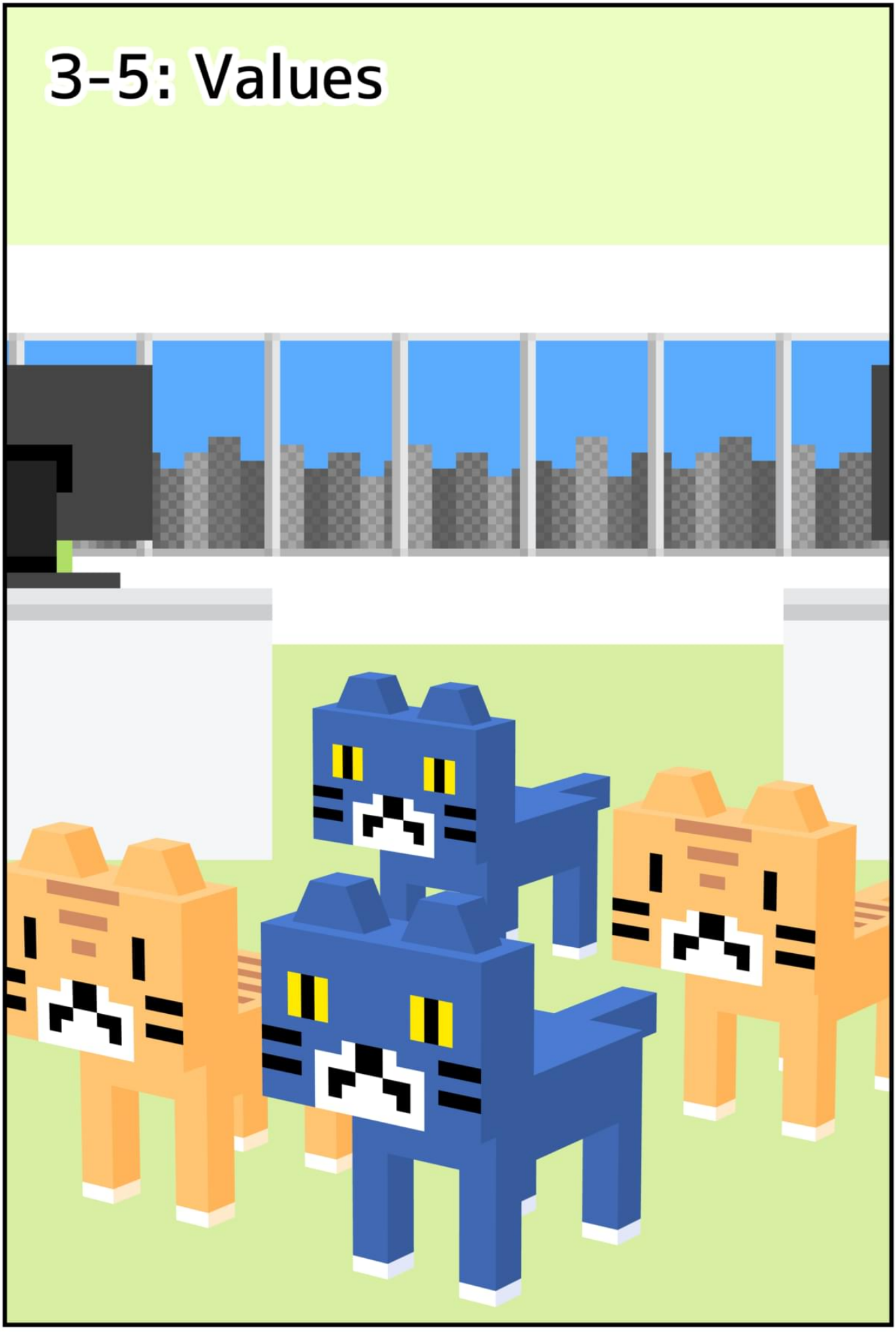
https://docs.python.org/3/library/functions.html

Built-in functions:
Functions that are available by default.




That's right.
For example, when building a car, you combine small parts to create bigger components, right?
Functions are really important in programming, aren't they?
We will properly learn how to create functions later.
In programming, you can create your own functions.
Then, you can combine functions or create larger functions by grouping them together.




3-5: Values

Numbers

1234

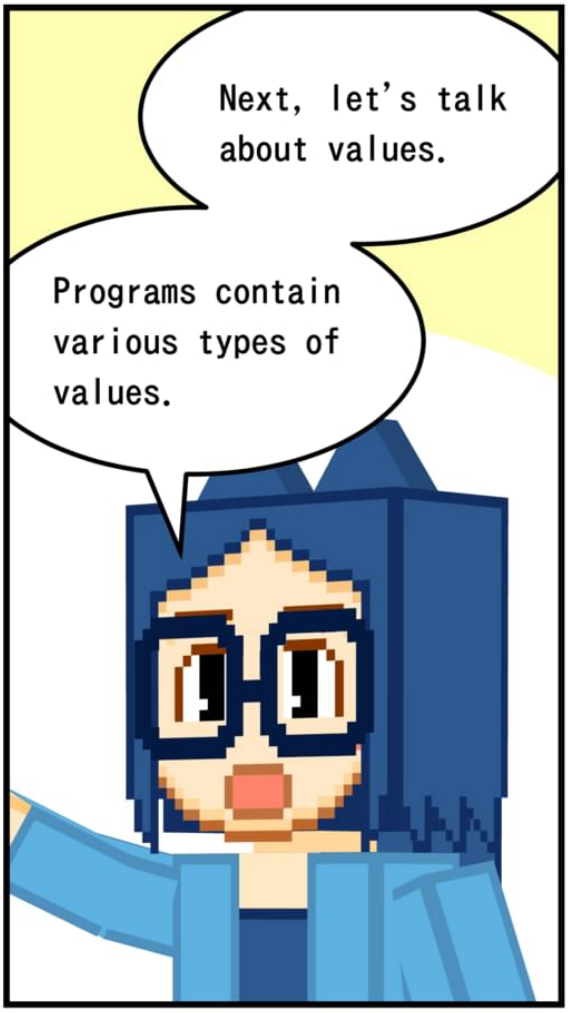
Text (strings)

Abcdef

Boolean values

True or False

There are other types as well.



Numbers

Integers

1234　0　-1234

Numbers without
a decimal point

Floating-point numbers

12.34　0.0　-12.34

Numbers with a decimal point

First, numbers.

There are integers and floating-point numbers, and they are treated separately.

Why are they treated separately?

Because the way they are stored in the computer is different.

That's why they are distinguished in programming.

Text (string)

'My Cat.'

"Your Cat!"

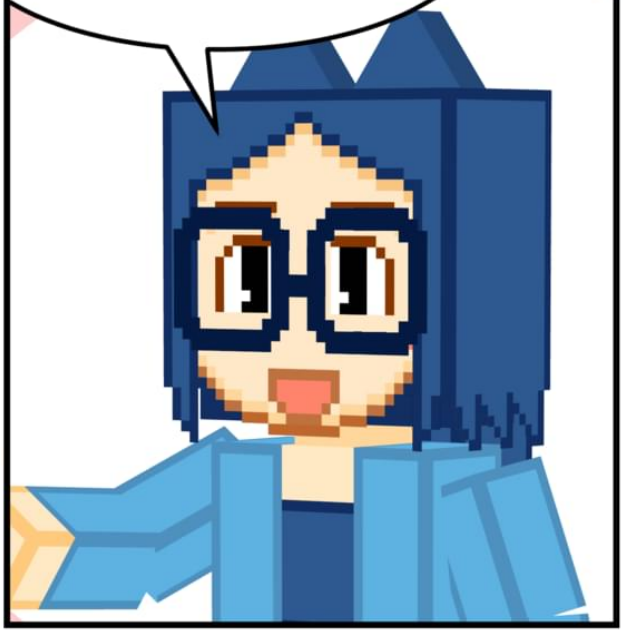
'A'

"z"

You can use single or double quotation marks.

Next, text, also called strings.

Strings are enclosed in quotation marks.



So, a string can have one or more characters, or even none at all!

Strings can also be empty, which is called an empty string.

''

""


'1234'
"12.34"
There are some things to be careful about with strings.
This is text, not a number.
Because it is enclosed in quotation marks.
That means it is not treated as a number.
Why?

Line break: \n

```
"The cat's name is\nTora"
```

The backslash itself: \\

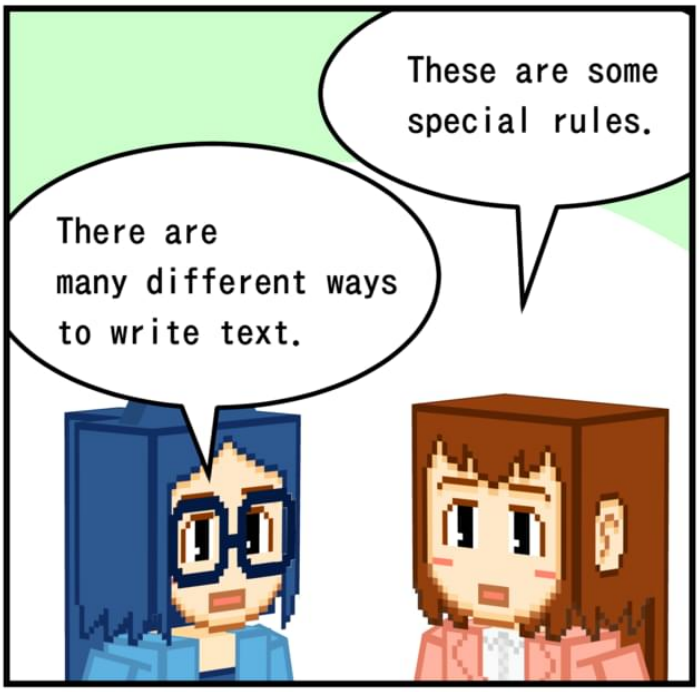
```
'work\\document.txt'
```

To use quotation marks inside a string, use "\".

```
'The important thing is \'heart\''
"The important thing is \"heart\""
```

Tab character: \t

```
'Animal\tCat\tTora'
```



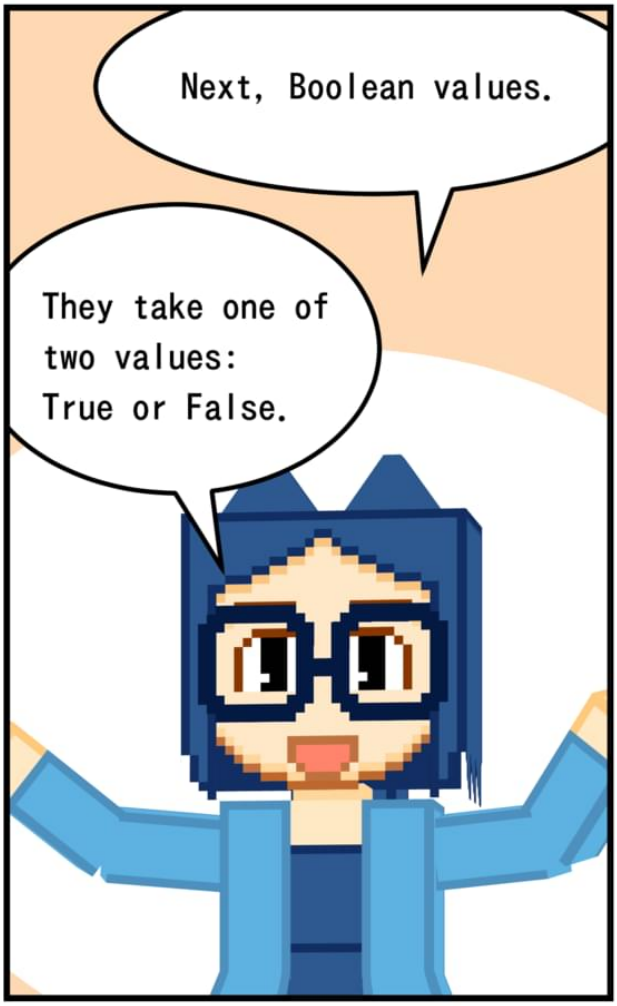
Boolean values

True

Yes, OK, valid, etc.

False

No, invalid, etc.



Boolean values are used for such decisions.

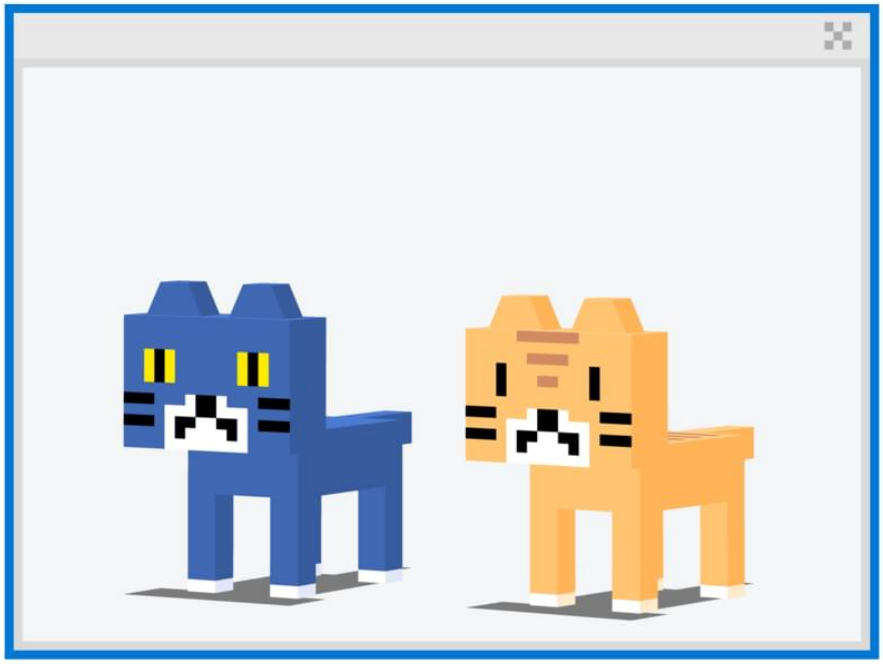
Process this

True →

False →

Process that


Thank you!
That's it for today.
Hehehe
The road ahead is long...
Now you can go play with the office cat.



3-6: Operators