PYTHON BASICS: FINDING AND FIXING C  DE BUGS

Python Basics: Finding and Fixing Code Bugs

It’s okay to make mistakes! 🤗

1. Syntax Errors
2. Runtime Errors
3. Logic Errors

Python Basics: Finding and Fixing Code Bugs

It’s okay to make mistakes! 🤗

1. Syntax Errors
2. Runtime Errors
3. **Logic Errors**

Python Basics: Finding and Fixing Code Bugs

Python Basics: Finding and Fixing Code Bugs

- **Bugs:** Unexpected behavior
- **Debugging:** Removing bugs
- **Debugger:** Tool to help find and understand bugs

Python Basics: Finding and Fixing Code Bugs

- Learn how to use IDLE's Debug Control window
- Practice debugging on a buggy function

IDLE's Debug Control Window

- Open it by selecting *Debug / Debugger*
- Look for `[DEBUG ON]` next to the prompt

> **Note:** The *Debug* menu item is only accessible in the interactive window.

The Step, Out, and Over Buttons

- **Step:** Execute one line of code and pause before the next one
- **Out:** Continue execution until you reach the end of the current scope, for example until the function you're in returns
- **Over:** Run a function call instead of stepping inside of that scope

Breakpoints, Go, and Quit

- **Breakpoint:** Set a breakpoint by right-clicking (Ctrl-click on a Mac) and selecting *Set Breakpoint*
- **Go:** Pressing *Go* runs all code until the next breakpoint
- **Quit:** Stop your debugging session using *Quit*

A Buggy Program

```
def add_underscores(word):
    new_word = "_"
    for char in word:
        new_word = char + "_"
    return new_word


phrase = "hello"
print(add_underscores(phrase))
```

A Buggy Program

Expected:

```
>>> add_underscores("hello")
"_h_e_l_l_o_"
```

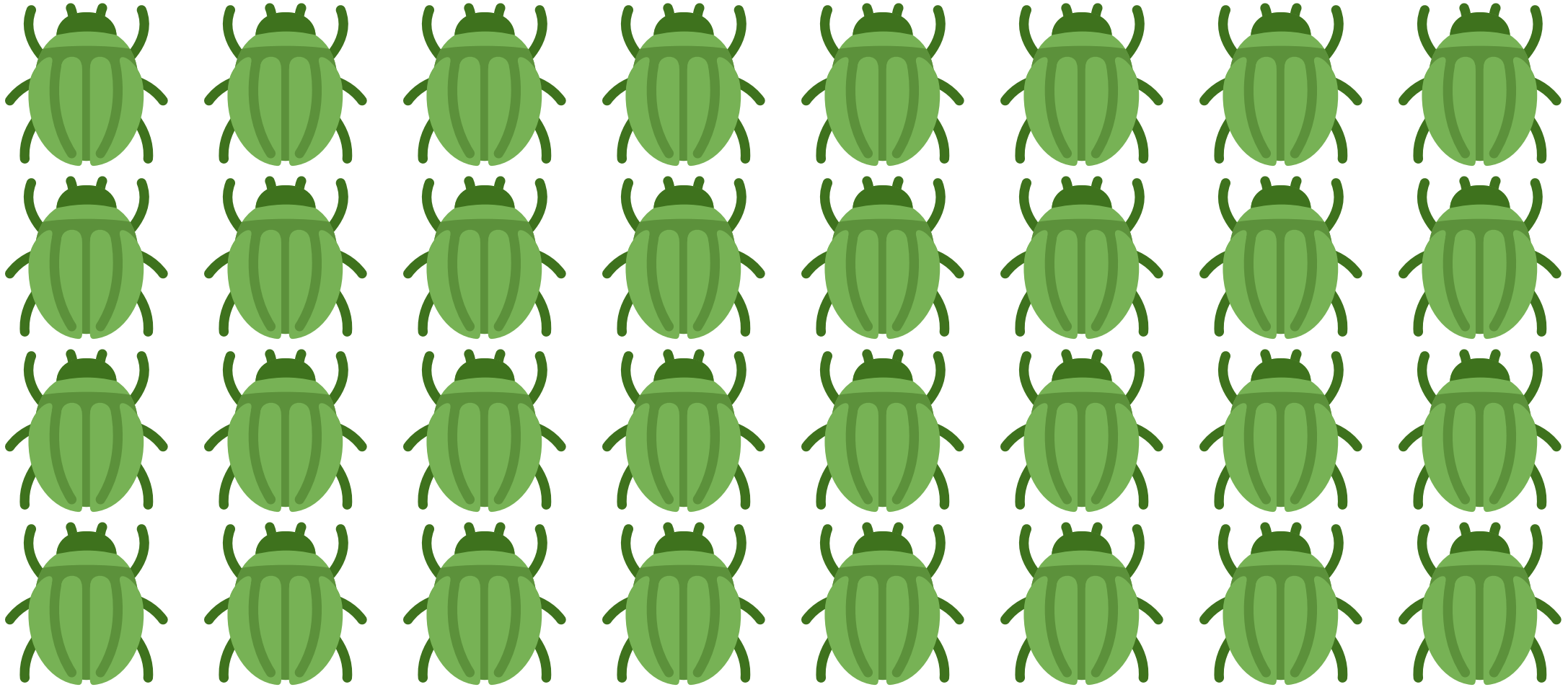
Actual:

```
>>> add_underscores("hello")
"o_"
```

Debugging Steps

1. **Guess** which section may contain the bug.
2. **Set a breakpoint** and **inspect** the code by stepping through it with your debugger.
3. **Identify** a possible error and **make a change**.
4. **Repeat** steps 1–3 as needed until the code works.

No Debugger? 😢



Print Debugging

An alternative way to find bugs

PYTHON BASICS: FINDING AND FIXING C  DE BUGS

PYTHON BASICS: FINDING AND FIXING CODE BUGS

Finding and Fixing Code Bugs

1. Learn how to use IDLE's Debug Control window
2. Practice debugging on a buggy function

IDLE's Debug Control Window

- Open with *Debug / Debugger* from the menu of the interactive window
- Watch for `[DEBUG ON]`

IDLE's Debug Control Window

- **Buttons:** *Step*, *Out*, *Over*, *Go*, and *Quit*
- **Checkboxes:** *Stack*, *Locals*, *Globals*, and *Source*
- **Panels:** *Stack*, *Locals*, *Globals*

The 4 Debugging Steps

1. Guess where the bug is located.
2. Set a breakpoint and inspect the code.
3. Identify the error and attempt to fix it.
4. Repeat steps 1–3 until the error is fixed.

A Buggy Program 

Expected:

```
>>> add_underscores("hello")
"_h_e_l_l_o_"
```

Actual:

```
>>> add_underscores("hello")
"o_"
```

A ~~Buggy~~ Program 🎉

Expected:

```
>>> add_underscores("hello")
"_h_e_l_l_o_"
```

Actual:

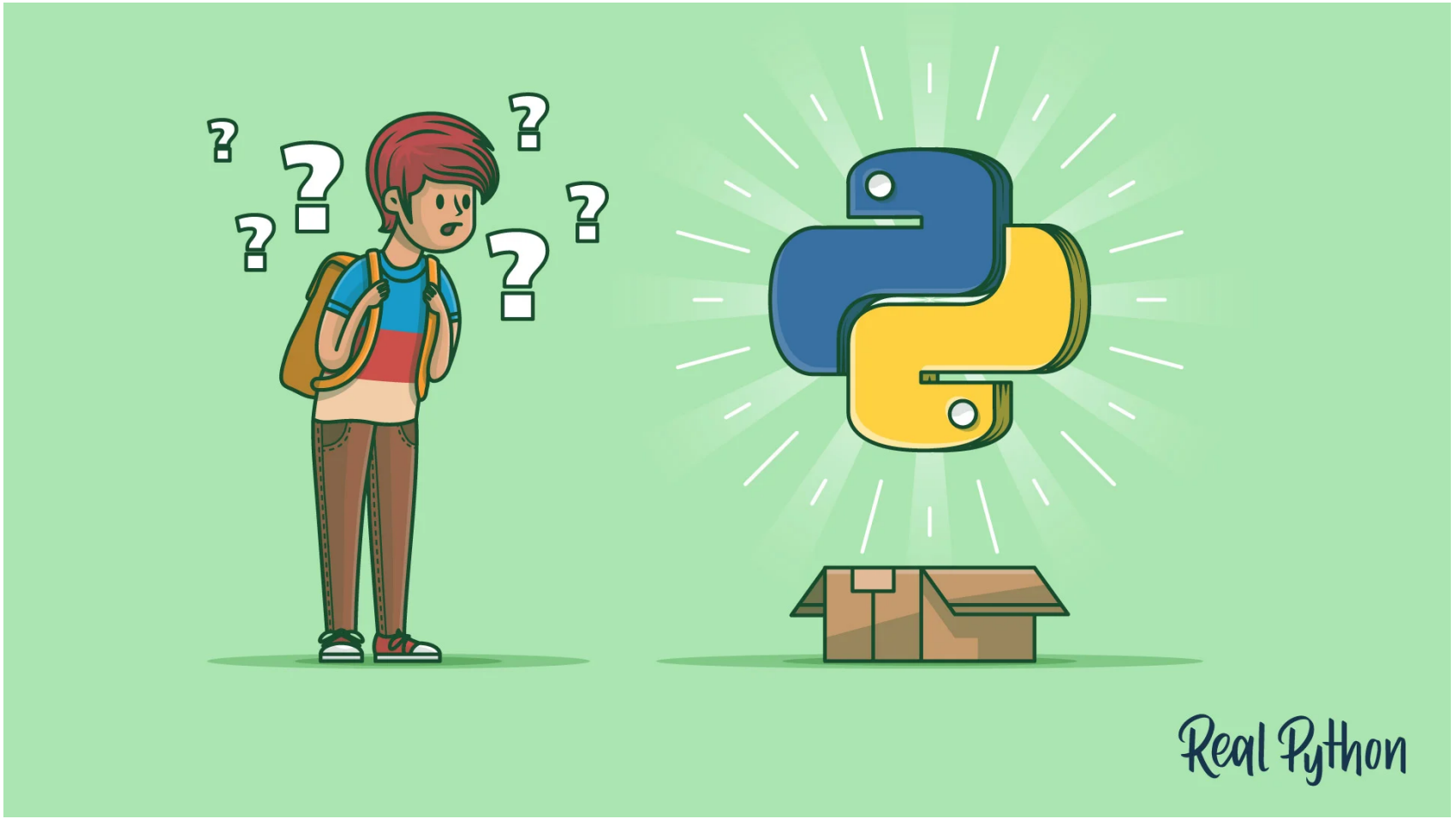
```
>>> add_underscores("hello")
"_h_e_l_l_o_"
```

Print Debugging

Add `print()` calls to inspect variables at different states of your program.

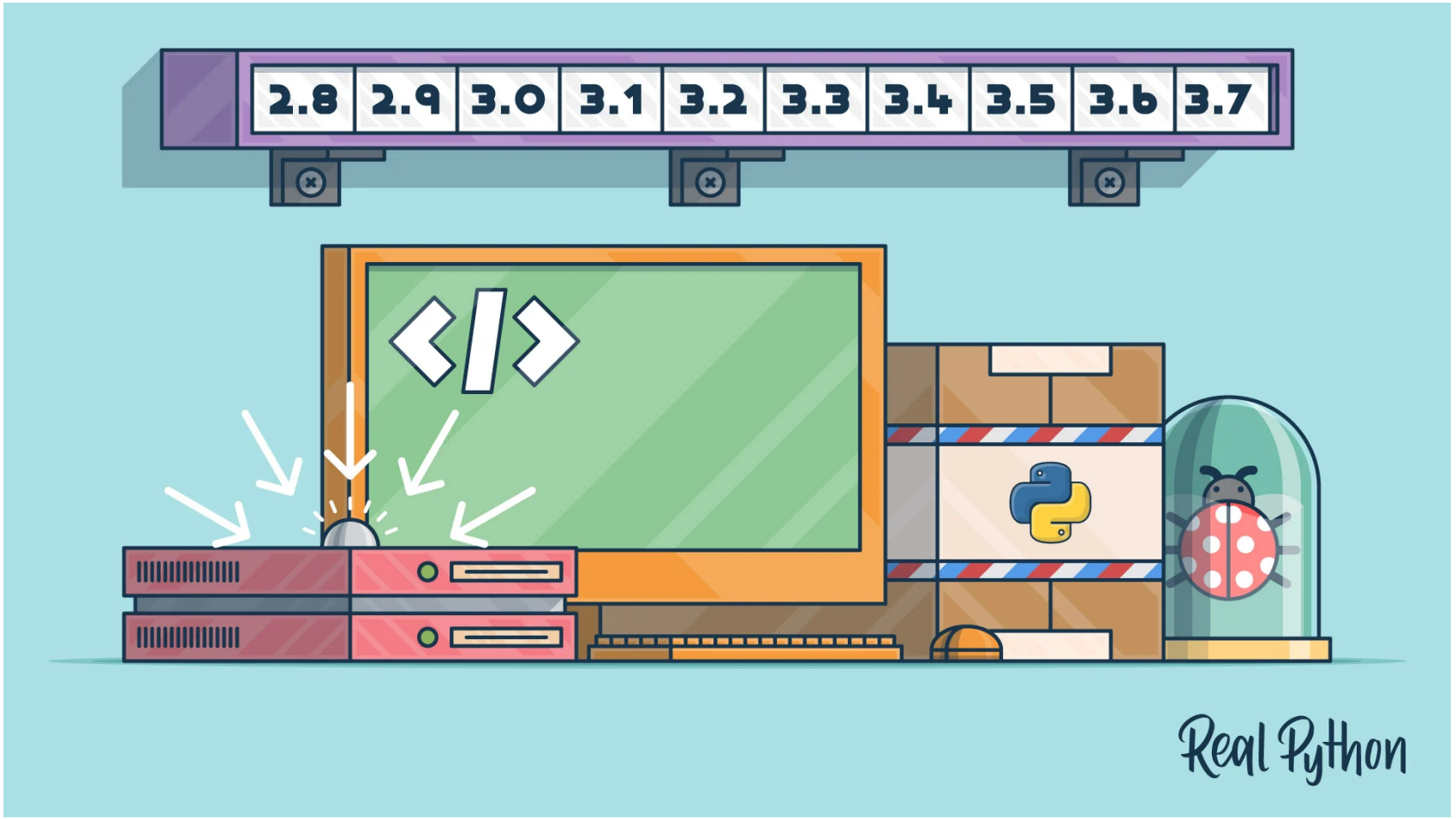
- ➕ Can use it in systems with limited resources (IoT devices)
- ➖ More code to write
- ➖ Need to run the whole program
- ➖ Need to remember to remove the `print()` calls afterwards

Additional Resources



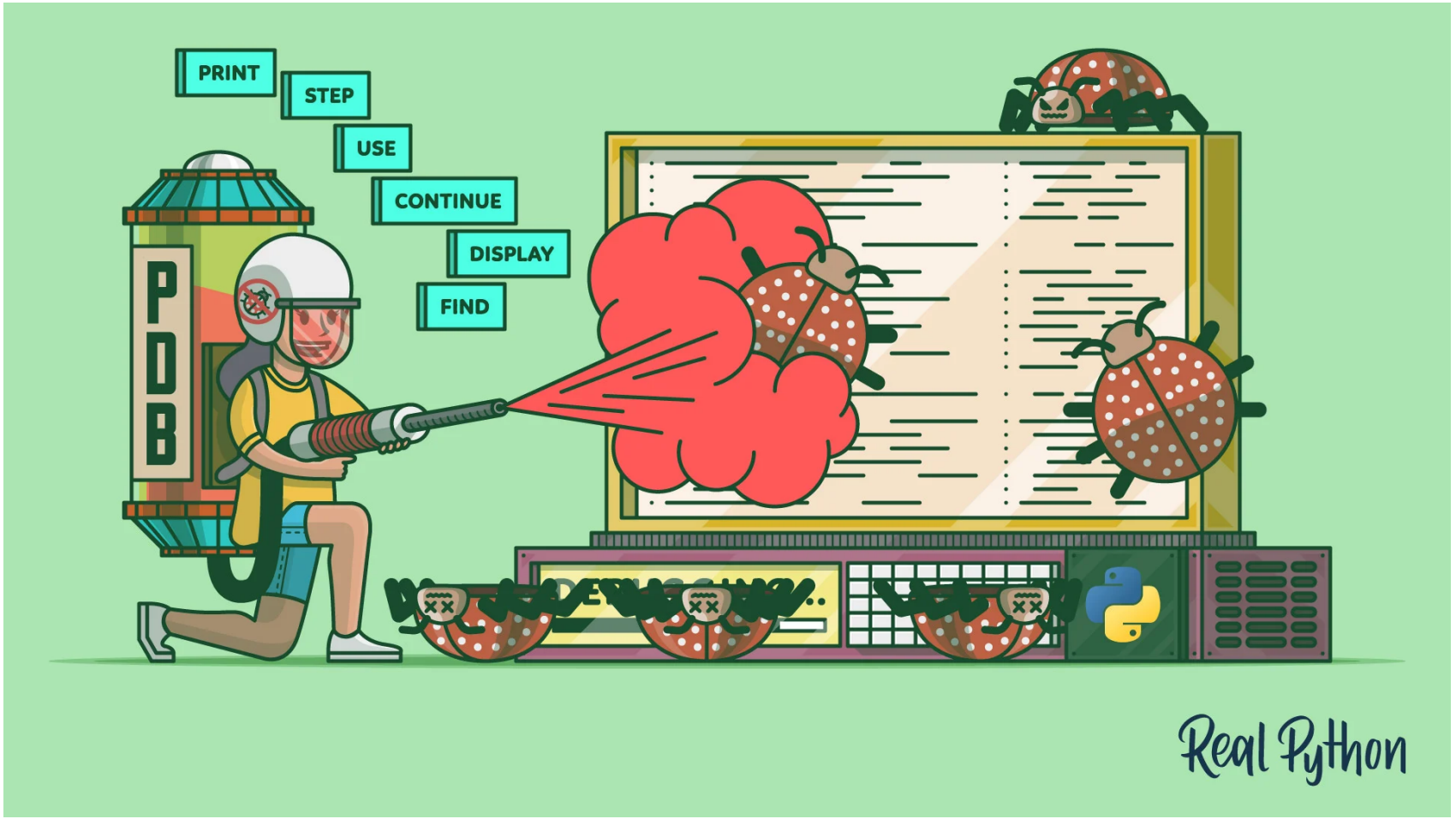
https://realpython.com/quizzes/pybasics-debugging/

Additional Resources



https://realpython.com/courses/python-debugging-pdb/

Additional Resources



https://realpython.com/python-debugging-pdb/

PYTHON BASICS: FINDING AND FIXING CODE BUGS