Python Programming
Unit 04 – Lecture 03 Notes
Event Handling and Input Validation

Tofik Ali

February 14, 2026

Contents

1 Lecture Overview

GUIs are event-driven: the program waits for user actions (clicks, typing) and responds using callback functions. Good GUI programs also validate inputs so that:

- the app does not crash,
- stored data remains consistent,
- and the user gets clear feedback.

2 Core Concepts

2.1 Callbacks with `command=`

The simplest event handling in Tkinter is attaching a callback to a Button:

```
def submit():
    print("Submitted")

tk.Button(root, text="Submit", command=submit).pack()
```

2.2 Binding Events with `bind`

`bind` attaches a function to a specific event sequence.

```
def on_key_release(event):
    print("Key:", event.keysym)

entry.bind("<KeyRelease>", on_key_release)
```

Common event sequences:

- `<Button-1>` left mouse click
- `<Return>` Enter key
- `<KeyRelease>` after a key is released
- `<Control-s>` Ctrl+S shortcut

2.3 Validation: What and When

What to validate:

- required fields (must not be empty),
- numeric fields (age must be a number),
- ranges (age 0–120),
- patterns (email).

When to validate:

- on submit (simple and reliable),
- while typing (better user experience, but more code).

2.4 Email Validation (Simple Regex)

For beginner apps, a simple pattern is often enough.

```
import re
EMAIL = r"^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$"
```

This checks format only. Real-world email validation is more complex.

2.5 User Feedback

Good feedback:

- explains what is wrong,
- tells the user how to fix it,
- does not expose technical details.

3 Demo Walkthrough

File: `demo/event_validation_demo.py`
What to observe:

- Button callback used for submission.
- Live validation using `bind` on input fields.
- Message boxes for clear feedback.
- Data is "accepted" only when all fields are valid.

4 Interactive Checkpoints (with Solutions)

Checkpoint 1 Solution

Question: When use `command=` vs `bind()`?
Answer:

- `command=` is best for button clicks (no event object needed).
- `bind()` is best for key/mouse events and shortcuts (event object needed).

Checkpoint 2 Solution

Question: Why is validation important?
Answer: It prevents wrong data, avoids crashes, and improves user experience.

5 Practice Exercises (with Solutions)

Exercise 1: Age Validation Rule

Task: Accept age only if it is an integer between 0 and 120.
Solution:

```python
age_text = age_var.get().strip()
if not age_text.isdigit():
    error = "Age must be an integer."
else:
    age = int(age_text)
    if not (0 <= age <= 120):
        error = "Age must be between 0 and 120."
```

Exercise 2: Keyboard Shortcut

Task: Bind Ctrl+L to clear the form.

Solution (idea):

```
def clear(event=None):
    name_var.set("")
    email_var.set("")
    age_var.set("")

root.bind("<Control-l>", clear)
```

6 Exit Question (with Solution)

Question: Write one validation rule for Age input.

Example answer: Age must be a non-negative integer (or in a range 0–120).