Question 3: Fix the Bug (Default Parameter Pitfall)

Problem Statement

In Python, using a mutable object (like a list) as a default argument can lead to unexpected behavior because the default value is evaluated only once at the time of function definition, not every time the function is called.

Given Code (The Bug)

```
def save_error(error, errors=[]):
    errors.append(error)
    return errors
```

Note: The default list `errors=[]` is shared across all function calls where an explicit list is not provided.

The Solution

Corrected Function (Fixed Version)

To fix this, we use `None` as the default value and initialize the list inside the function body.

```
def save_error(error, errors=None):
    if errors is None:
        errors = []
    errors.append(error)
    return errors
```

Calling the Corrected Function

```
print(save_error("E1"))
print(save_error("E2"))
print(save_error("E3"))
```

Expected Output

Because a new list is created for each call where `errors` is `None`, the output is:

```
['E1']
['E2']
['E3']
```

Explanation

- **Avoiding Pitfalls:** Using `None` avoids the *default mutable argument pitfall* because `None` is immutable.
- **Correct Behavior:** This approach ensures the function behaves correctly on repeated calls, as it guarantees a fresh list is instantiated whenever the argument is omitted.