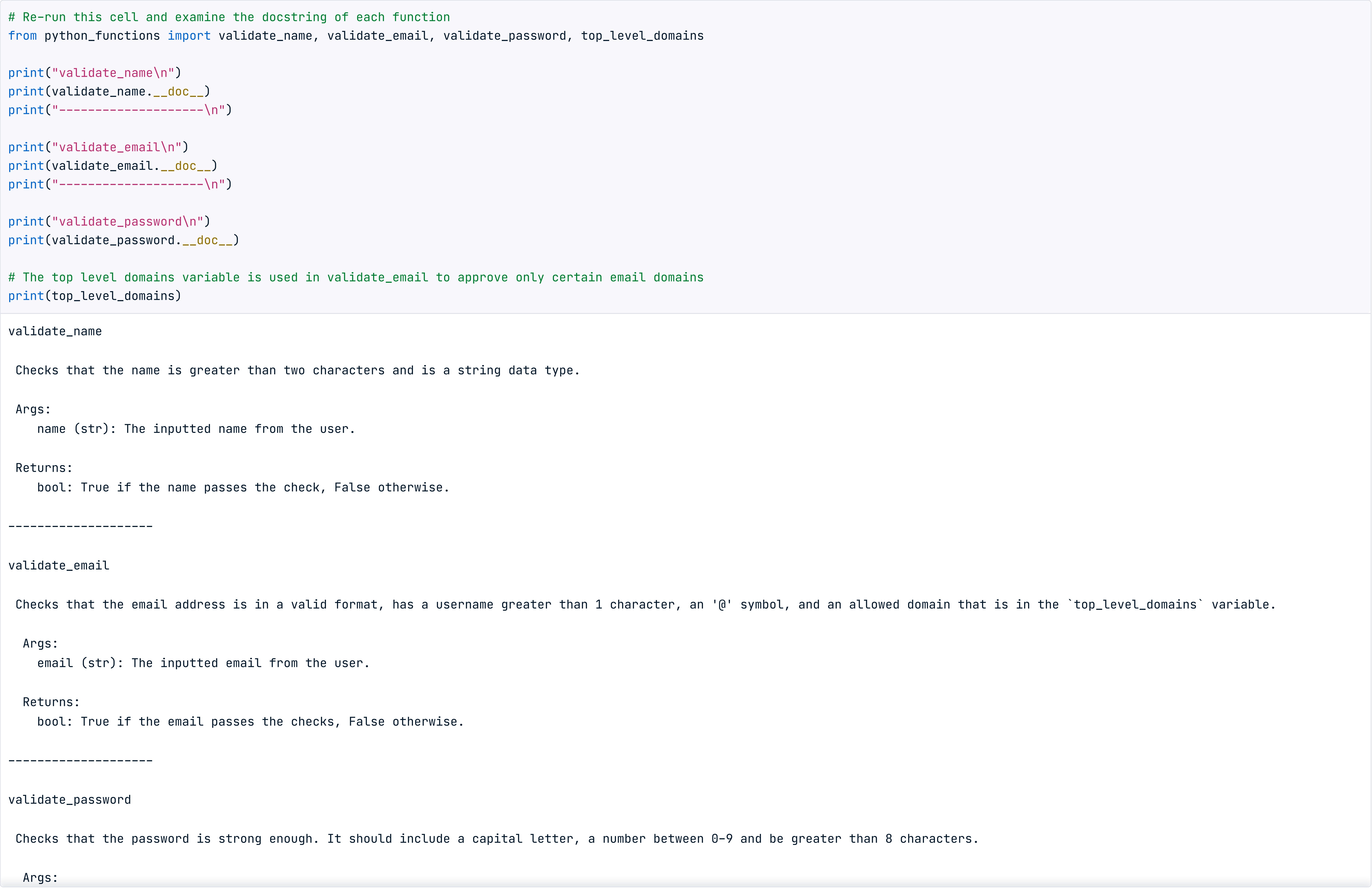
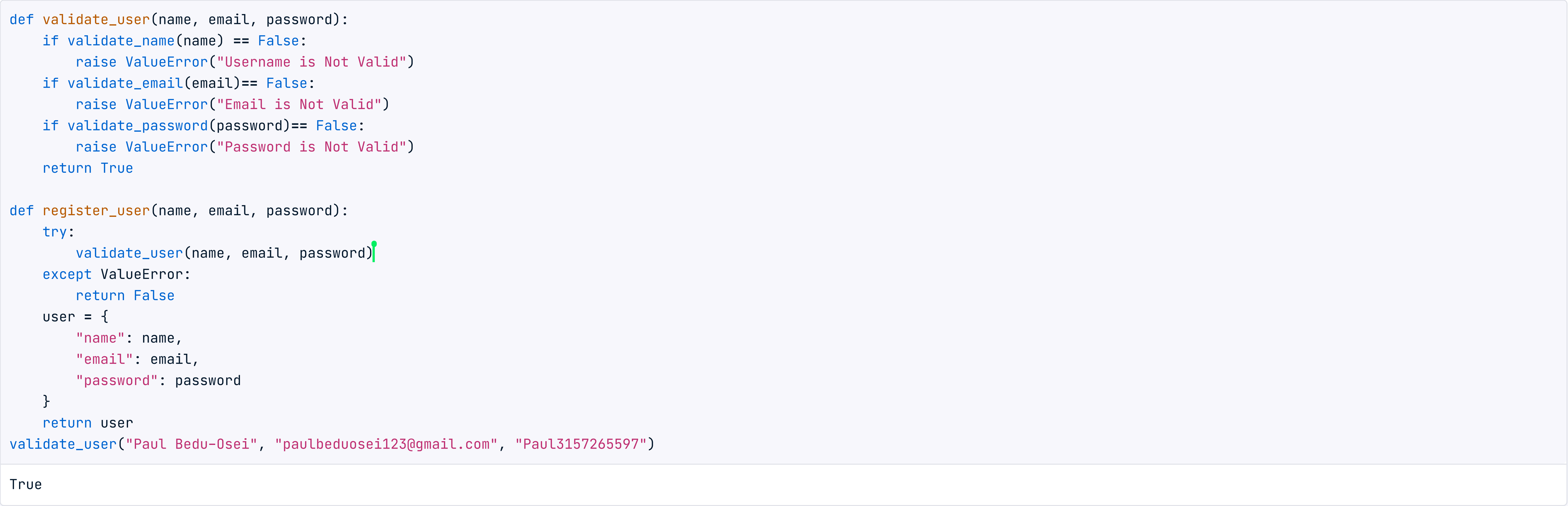
You are a junior developer working in a small start-up. Your managers have asked you to develop a new account registration system for a mobile app. The system must validate user input on the sign-up form before creating an account.

The previous junior developer wrote some helper functions that validate the name, email, and password. Use these functions to register users, store their data, and implement some error handling! These have been imported into the workspace for you. They will be a great help to you when registering the user, but first you have to understand what the function does! Inspect the docstrings of each of the helper functions: `validate_name`, `validate_email` and `validate_password`.



```python
# Re-run this cell and examine the docstring of each function
from python_functions import validate_name, validate_email, validate_password, top_level_domains

print("validate_name\n")
print(validate_name.__doc__)
print("--------------------\n")

print("validate_email\n")
print(validate_email.__doc__)
print("--------------------\n")

print("validate_password\n")
print(validate_password.__doc__)

# The top level domains variable is used in validate_email to approve only certain email domains
print(top_level_domains)
```

```
validate_name

 Checks that the name is greater than two characters and is a string data type.

 Args:
    name (str): The inputted name from the user.

 Returns:
    bool: True if the name passes the check, False otherwise.

--------------------

validate_email

 Checks that the email address is in a valid format, has a username greater than 1 character, an '@' symbol, and an allowed domain that is in the `top_level_domains` variable.

  Args:
    email (str): The inputted email from the user.

  Returns:
    bool: True if the email passes the checks, False otherwise.

--------------------

validate_password

 Checks that the password is strong enough. It should include a capital letter, a number between 0-9 and be greater than 8 characters.

  Args:
```



```python
def validate_user(name, email, password):
    if validate_name(name) == False:
        raise ValueError("Username is Not Valid")
    if validate_email(email)== False:
        raise ValueError("Email is Not Valid")
    if validate_password(password)== False:
        raise ValueError("Password is Not Valid")
    return True

def register_user(name, email, password):
    try:
        validate_user(name, email, password)
    except ValueError:
        return False
    user = {
        "name": name,
        "email": email,
        "password": password
    }
    return user
validate_user("Paul Bedu-Osei", "paulbeduosei123@gmail.com", "Paul3157265597")
```

```
True
```