**Functional**

def validate\_login\_functional(username, password):

def is\_valid\_email(email):

return '@' in email and '.' in email.split('@')[1]

if not username or not password:

return False

if not is\_valid\_email(username):

return False

if len(password) <= 8:

return False

return True

def login\_page\_functional():

username = "testuser@example.com"

password = "testpassword"

if validate\_login\_functional(username, password):

print("Login successful!")

else:

print("Login failed. Please check your input.")

login\_page\_functional()

**OOP**

class LoginValidator:

def \_\_init\_\_(self, username, password):

self.username = username

self.password = password

def validate\_login(self):

if not self.username or not self.password:

return False

if not self.is\_valid\_email():

return False

if len(self.password) <= 8:

return False

return True

def is\_valid\_email(self):

return '@' in self.username and '.' in self.username.split('@')[1]

def login\_page\_oop():

username = "testuser@gmail.com"

password = "apassword"

validator = LoginValidator(username, password)

if validator.validate\_login():

print("Login successful!")

else:

print("Login failed. Please check your input.")

login\_page\_oop()

**Procedural**

def validate\_login\_procedural(username, password):

def is\_valid\_email(email):

return '@' in email and '.' in email.split('@')[1]

if not username or not password:

return False

if not is\_valid\_email(username):

return False

if len(password) <= 8:

return False

return True

def login\_page\_procedural():

username = "testuser@gmail.com"

password = "apassword"

if validate\_login\_procedural(username, password):

print("Login successful!")

else:

print("Login failed. Please check your input.")

login\_page\_procedural()