

 Marwadi University <small>Marwadi Chandarana Group</small>	 NAAC A+	Marwadi University Faculty of Engineering & Technology Department of Information and Communication Technology
Subject: Programming With Python (01CT1309)	Aim: Practical demonstrate to validate PAN card number and Email ID.	
Experiment No: 23	Date:	Enrollment No: 92510133028

Aim: Practical demonstrate to validate PAN card number and Email ID.

IDE:

For this experiment, you only need the built-in **re** library for regular expressions, which comes with Python.

Validating a PAN Card Number

PAN Card Format:

- A PAN (Permanent Account Number) card in India consists of 10 characters.
- The first five characters are uppercase letters (A-Z).
- The next four characters are digits (0-9).
- The last character is an uppercase letter (A-Z).

Regular Expression:

```
^[A-Z]{5}[0-9]{4}[A-Z]{1}$
```

```
import re
```

```
def validate_pan(pan):
    pattern = r'^[A-Z]{5}[0-9]{4}[A-Z]{1}$'
    if re.match(pattern, pan):
        return True
    return False
```

```
# Test the function
```

```
pan_number = input("Enter PAN card number: ")
if validate_pan(pan_number):
```



Subject: Programming With Python (01CT1309)

Aim: Practical demonstrate to validate PAN card number and Email ID.

Experiment No: 23

Date:

Enrollment No: 92510133028

```
print("Valid PAN card number.")
```

else:

```
print("Invalid PAN card number.")
```

Validating an Email ID

```
import re

def validate_pan(pan):
    pattern = r'^[A-Z]{5}[0-9]{4}[A-Z]{1}$'
    if re.match(pattern, pan):
        return True
    return False

# Test the function
pan_number = input("Enter PAN card number: ")
if validate_pan(pan_number):
    print("Valid PAN card number.")
else:
    print("Invalid PAN card number.")

✓ 11.0s
Valid PAN card number.
```

Email Format:

- An email consists of local and domain parts, separated by an @ symbol.
- The local part can include letters, digits, dots, underscores, and hyphens.
- The domain part should consist of a domain name and a top-level domain (TLD), such as .com, .org, etc.

Regular Expression:

[^@\[a-zA-Z0-9._%+-\]+@\[a-zA-Z0-9.-\]+\.\[a-zA-Z\]{2,}\\$_](#)

```
def validate_email(email):
```

```
pattern = r'^@[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$_'
```

```
if re.match(pattern, email):
```

```
    return True
```



Subject: Programming With Python (01CT1309)

Aim: Practical demonstrate to validate PAN card number and Email ID.

Experiment No: 23

Date:

Enrollment No: 92510133028

return False

```
# Test the function
```

```
email_id = input("Enter email ID: ")
```

```
if validate_email(email_id):
```

```
    print("Valid email ID.")
```

```
else:
```

```
    print("Invalid email ID.")
```

```
def validate_email(email):
    pattern = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$
    if re.match(pattern, email):
        return True
    return False

# Test the function
email_id = input("Enter email ID: ")
if validate_email(email_id):
    print("Valid email ID.")
else:
    print("Invalid email ID.")

✓ 15.45
```

```
Valid email ID.
```

Post Lab:

Write a code combine both validations into a single program



Subject: Programming With Python (01CT1309)

Aim: Practical demonstrate to validate PAN card number and Email ID.

Experiment No: 23

Date:

Enrollment No: 92510133028

```
def validate_both():
    # Get both inputs
    pan = input("Enter PAN card number: ")
    email = input("Enter email ID: ")

    # Validate PAN
    if validate_pan(pan):
        print("Valid PAN card number.")
    else:
        print("Invalid PAN card number.")

    # Validate Email
    if validate_email(email):
        print("Valid email ID.")
    else:
        print("Invalid email ID.")

# Run the combined validation
validate_both()
```

✓ 26.8s

```
Valid PAN card number.
Valid email ID.
```

Github : https://github.com/keshvi1234/PWP_experiment