📝 Session 12: File Handling & Advanced Loops

🎯 Objectives

- Learn how to read and write files in Python (txt & CSV).
- Understand the `with` statement for safe file handling.
- Explore loop control statements: `break`, `continue`, `pass`.
- Practice nested loops & print fun patterns.
- Build mini-projects (games) using file handling.

📂 1. Introduction to File Handling

File handling allows Python programs to **store and retrieve data** from external files.
This makes data *persistent*, meaning it can be saved beyond program execution.

Common Modes:

- `"r"` → Read (default)
- `"w"` → Write (overwrites file)
- `"a"` → Append
- `"b"` → Binary mode
- `"t"` → Text mode (default)

```
# Reading a file
file = open("example.txt", "w")
file.write("Hello, World!\nThis is file handling in Python.")
file.close()

file = open("example.txt", "r")
print(file.read())
file.close()
```

```
Hello, World!
This is file handling in Python.
```

```
file = open("example.txt", "r")
for line in file:
    print("Line:", line.strip())
file.close()
```

```
Line: Hello, World!
Line: This is file handling in Python.
```

```
file = open("output.txt", "w")  # Overwrites existing content
file.write("First line\n")
file.write("Second line\n")
file.close()
```

```
file = open("output.txt", "a")
file.write("Appended line\n")
file.close()
```

```
import csv

# Writing CSV
with open("students.csv", "w", newline="") as file:
    writer = csv.writer(file)
    writer.writerow(["Name", "Marks"])
    writer.writerow(["Alice", 90])
    writer.writerow(["Bob", 85])

# Reading CSV
with open("students.csv", "r") as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

```
['Name', 'Marks']
['Alice', '90']
['Bob', '85']
```

🔄 Loop Control Statements

- **break** → Exit the loop completely.
- **continue** → Skip the current iteration.
- **pass** → Do nothing (placeholder).

```
# break example
for i in range(5):
    if i == 3:
        break
    print("i =", i)

# continue example
for i in range(5):
    if i == 2:
        continue
    print("i =", i)

# pass example
for i in range(5):
    if i == 4:
        pass
    print("i =", i)
```

Pattern Printing

```
# Square Pattern
n = 5
for i in range(n):
    for j in range(n):
        print("*", end=" ")
    print()

# Right Triangle
for i in range(1, n+1):
    print("*" * i)

# Pyramid
for i in range(1, n+1):
    print(" "*(n-i) + "*"*(2*i-1))

# Inverted Triangle
for i in range(n, 0, -1):
    print("*"*i)
```

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
*
**
***
****
*****
    *
   ***
  *****
 *******
*********
*****
****
***
**
*
```