

# *Week 9*

# *Lecture 17*



# Files

## Persistent storage

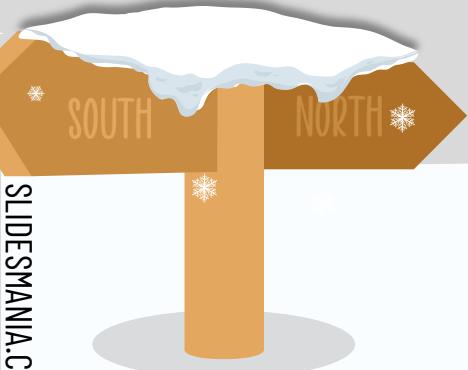
Data stored in a file is said to persist. A database is an example of persistent storage.

## File actions

We can create files, add to files, or read from files using Python.

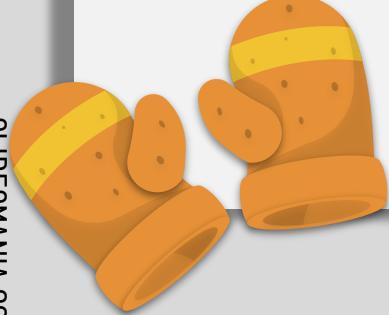
## Text files

Text files store characters and can be easily read by humans.





W



# Writing to a file

```
newfile = open("famousfive.txt", "w")  
newfile.close()
```

This code creates a file called famousfive.txt in the project's folder.

'w' defines the type of file access  
'w' means write to the file





W

## Using print() to write to the file

```
output = open("famousfive.txt", "w")
print("Julian", file=output)
print("Dick", file=output)
print("Anne", file=output)
print("George", file=output)
output.close()
```

`close()` closes the connection to the file.

Any attempt to write to the file after the file connection has closed will cause an error.

`print("Julian")` prints to the console.

But we can specify that the output must go elsewhere.



W



```
with open("famousfive.txt", "w") as output:  
    print("Julian", file=output)  
    print("Dick", file=output)  
    print("Anne", file=output)  
    print("George", file=output)
```

The first line creates a file called "famousfive.txt" and creates a connection to that file called `output`.

All indented code can access `output`.

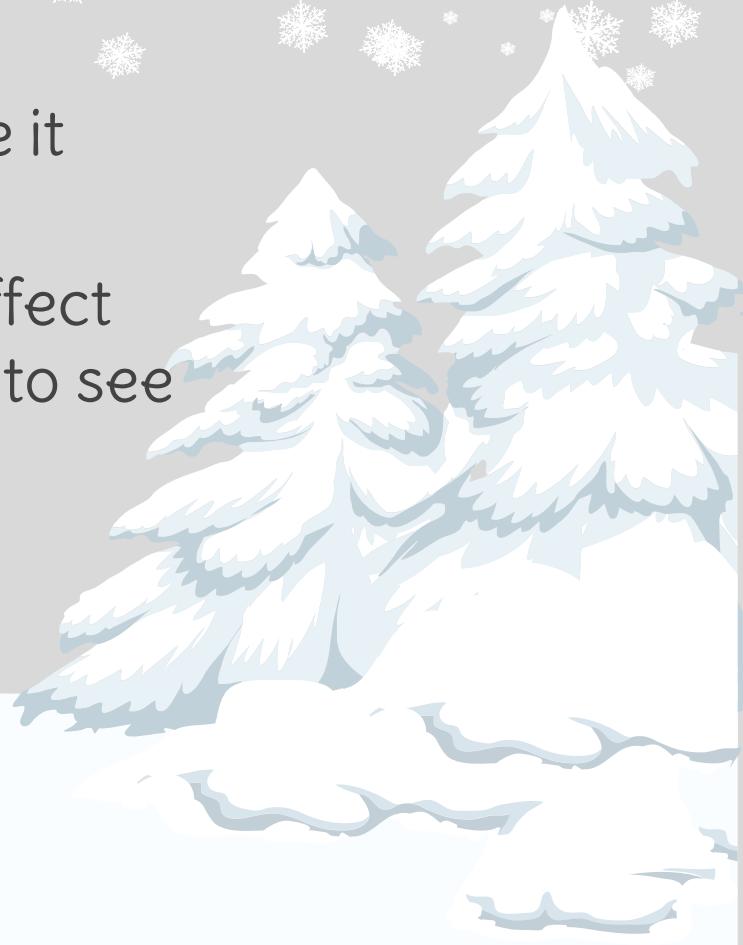
The `print()` statements print the names to the file.

The connection is closed outside of this indented section.

## Alternative using `with`

# A note on closing files!

- ✿ Leaving a file open can slow down the program because it uses up RAM which impacts performance.
- ✿ In most cases, changes written to a file do not go into effect until *after* the file is closed, so closing the file allows you to see the newly edited/created file.
- ✿ In theory you could run out of connections.
- ✿ It is bad programming practice to leave a file open.



# File access modes

`open(filename, access_option)`

w

- Writing to a file – note: any previous content in the file is deleted

a

- Appending or adding to the end of a file

r

- Reading from a file





a

## Adding to a file

```
f = open("famousfive.txt", "a")
print("Timmy the dog.", file=f)
f.close()
```

Either creates "famousfive.txt" and writes "Timmy the dog" to it or adds it to the end of the file.

# Gathering and storing information

Open a file

Ask how many employees?

In a loop

Get name

Get ID number

Get department

Write name, ID, and department to a file

How could this process go wrong?

```
number_of_employees = int(input("Enter number of employees: "))

with open("employees.txt", 'w') as employee_file:
    # For each employee ask user for details and write these to the file
    for i in range(number_of_employees):
        print(f"Enter details for employee # {i+1}")
        name = input("Name: ")
        id_number = input("ID number: ")
        dept = input("Department: ")
        print(f"{name} {id_number} {dept}", file=employee_file)
```



# Gathering and storing information

Open a file

Ask how many employees?

In a loop

Get name

Get ID number

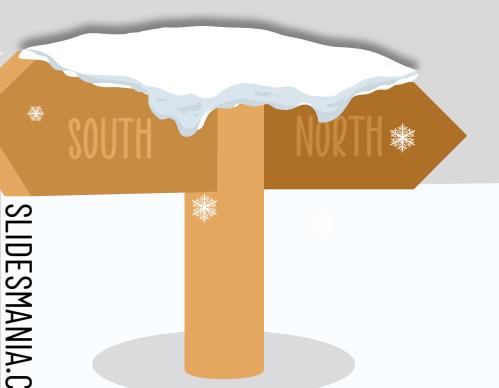
Get department

Write name, ID, and department to a file

1. The user might not enter a positive integer for `number_of_employees`
2. It might not be possible to create the file

```
number_of_employees = int(input("Enter number of employees: "))

with open("employees.txt", 'w') as employee_file:
    # For each employee ask user for details and write these to the file
    for i in range(number_of_employees):
        print(f"Enter details for employee # {i+1}")
        name = input("Name: ")
        id_number = input("ID number: ")
        dept = input("Department: ")
        print(f"{name} {id_number} {dept}", file=employee_file)
```



# Only positive integers

```
while True:  
    try:  
        number_of_employees = int(input("Enter number of employees: "))  
        if number_of_employees > 0:  
            break  
    except ValueError:  
        print("The number of employees must be a positive, whole number. Please try again.")  
  
with open("employees.txt", 'w') as employee_file:  
    # For each employee ask user for details and write these to the file  
    for i in range(number_of_employees):  
        print(f"Enter details for employee # {i+1}")  
        name = input("Name: ")  
        id_number = input("ID number: ")  
        dept = input("Department: ")  
        print(f"{name} {id_number} {dept}", file=employee_file)
```



# Handle file write errors

```
while True:  
    try:  
        number_of_employees = int(input("Enter number of employees: "))  
        if number_of_employees > 0:  
            break  
    except ValueError:  
        print("The number of employees must be a positive, whole number. Please try again.")  
  
try:  
    with open("employees.txt", 'w') as employee_file:  
        # For each employee ask user for details and write these to the file  
        for i in range(number_of_employees):  
            print(f"Enter details for employee # {i+1}")  
            name = input("Name: ")  
            id_number = input("ID number: ")  
            dept = input("Department: ")  
            print(f"{name} {id_number} {dept}", file=employee_file)  
    except FileNotFoundError:  
        print("Sorry, I couldn't create a file there.")
```

