1.How do you open and write to a file in Python?

You use the built-in `open()` function.

```
file = open("example.txt", "w")   # Open file for writing
file.write("Hello, World!")
file.close()
```

2.What are common file modes?

R-Read -opens file for reading
W-write-create a new file or overwrites existing one
a-append
r+-read and write
x- exlusive create

3.What's the use of .strip()?
`.strip()` removes **whitespace characters** (spaces, tabs, newlines) from the **beginning and end** of a string.

```
text = "  Hello World!  "
print(text.strip())  # Output: "Hello World!"
```

4.How do lists work in Python?

A **list** is an **ordered, mutable (changeable) collection** that can store multiple items of different types.

```
fruits = ["apple", "banana", "cherry"]
fruits[1] = "mango"  # Modify
```

5.What is the difference between append() and insert()?

append(x) --Adds `x` at the **end** of the list --fruits.append("orange")
insert(i, x) ---Adds `x` at **position `i`** --fruits.insert(1, "kiwi")

6.How can you remove elements from a list?

```
fruits.remove("apple")  # Remove by value
fruits.pop(1)          # Remove by index
del fruits[0]          # Delete by index
fruits.clear()         # Remove all elements
```

7.What are context managers (with statement)?

Context managers handle **setup and cleanup** automatically — commonly used for file handling

```
with open("data.txt", "r") as f:
    data = f.read()
# File is automatically closed here
```

8.How do you loop through a file line by line?

```
with open("data.txt", "r") as file:
    for line in file:
        print(line.strip())
```

9.What is a data structure?

A **data structure** is a way of **organizing and storing data** so it can be used efficiently.
Examples in Python:
List – ordered, mutable collection
Tuple – ordered, immutable collection
Set – unordered, unique elements
Dict – key-value pairs
10.What happens if the file doesn't exist?

Opening with **`"r"` (read mode)** → raises a `FileNotFoundError`.
Opening with **`"w"` or `"a"` (write/append mode)** → creates the file automatically.

```
try:
    open("missing.txt", "r")
except FileNotFoundError:
    print("File not found!")
```