# os Modul

```
import os
for i in  range(1, 11):

    Ramiz = f"day_{i}"
    if not os.path.exists(Ramiz):

        os.makedirs(Ramiz)

        print(f"The Folder is Creted :{Ramiz}")
    else:
        print(f"The Folder is Allrasy exist :")
```

#logging module 1.

```
import logging
logging.basicConfig(level=logging.DEBUG)
logging.debug("This is a Debug Message:")
logging.info("This  is a info message:")
logging.error("This is a Error Message:")
logging.critical("This is a Critical Message:")
logging.warning("This is a Warning message:")
```

# Loging module 2 -> (File):

```
import logging

logging.basicConfig(
    filename = "loging_log.log",
    level=logging.DEBUG,
    format="%(asctime)s - %(levelname)s - %(message)s -"
)
logging.debug("This is a Debug :")
logging.info("This is a Info :")
logging.warning("This is a Waring:")
logging.error("This is a Error:")
logging.critical("This is a Critical:")
```

*# Render Module :*

```
import re
My_phonenumber  = "My phone number is  9641988821"
patten  = r"\d+"
matches = re.findall(patten , My_phonenumber)
print(matches)
```

```
# Json Module :
import json
Student = {
   "name": "Ramiz",
   "age": 22,
   "course": "python",
   "skill": ["Python", "ML", "AI"],
   "active": True
}

# Write JSON data to a file
with open("Student.json", "w") as f:
   json.dump(Student, f, indent=4)

# Read JSON data from the file
with open("Student.json", "r") as f:
   data = json.load(f)

print("Name:", data["name"])
print("Skills:", ", ".join(data["skill"]))


# Jeson Modul 2 :

import json
```

```
# Existing data (List of students)
students = [
  {"name": "Ali", "age": 21, "course": "Python"},
  {"name": "Sara", "age": 22, "course": "AI"},
  {"name": "Ramiz", "age": 23, "course": "Data Science"}
]

# Write to JSON file
with open("students.json", "w") as f:
  json.dump(students, f, indent=4)

# Read the JSON file
with open("students.json", "r") as f:
  data = json.load(f)

# Add a new student
new_student = {"name": "Zara", "age": 20, "course": "Web Development"}
data.append(new_student)
# Write updated data back to JSON file
with open("students.json", "w") as f:
  json.dump(data, f, indent=4)
# Print all student names
print("All Students:")
for student in data:
  print("-", student["name"])   -------- End !
```

# Jeson Modul 3

```
import json
Student = {"naem": "Ramiz", 'Age': 21, "Course": "Ai"}

json_string = json.dumps(Student, indent= 4)
print(json_string)
```

# json 4

```
import json

json_data ='{"name": "Ramiz", "Age": 21}'

Student = json.loads(json_data)

print(Student)
```

# Timer 1

```
import  time
print("Current Timestamp:", time.time())
print("Wait for  2  seconds.....")
time.sleep(2)
print("Done!")
local_time = time.localtime()
print("Local Time:", time.strftime("%Y -%m-%d %H: %M: %S", local_time))
```

*# Date or Time 1*

```
import datetime
now =  datetime.datetime.now()
print("Current date and time", now)
```

# Date or Time 2

```
import datetime

today = datetime.date.today()

print("Today 's Date ", today)
print("Year:", today.year)
print("Year:", today.month)
print("Day:", today.day)
```

# Date or Time 3

```
import  datetime

now = datetime.datetime.now()

formatted_date = now.strftime("%d/%m/%Y")
```

```
print("Formatted  date:", formatted_date )

formatted_date =now.strftime("%I: %M %p")
print("Formatted Time:",  formatted_date)



# Date or time 4

from datetime import  datetime

custom_dt = datetime(2025, 12, 25, 10, 30)

print("Custome DateTime:", custom_dt)

print("Formatted", custom_dt.strftime("%A, %d %B %Y, %I:%M:%p"))



# Date or Time 5

from  datetime import  datetime,  timedelta

now = datetime.now()

print("Current Time:", now)
```

```
future_date  = now + timedelta(days =7)


print("After 7 days is :", future_date)
past_date  =now  - timedelta(days=10)
print("10 Days Ago :", past_date)
```