Python Basics + Libraries + REST API (Detailed Notes)

Dictionaries

- Store key-value pairs (mutable, ordered in Python 3.7+).
- Keys are unique; values can be anything.
- Fast lookup by key.

Example:

```
person = {"name": "Sakshi", "age": 22}
person["city"] = "Delhi"
print(person["name"])
```

Lists

- Ordered, mutable collection of items.
- Indexed access, allows duplicates.

Example:

```
fruits = ["apple", "banana"]
fruits.append("mango")
print(fruits[0])
```

Functions

- Reusable blocks of code.
- Support parameters, return values, default arguments.

Example:

```
def greet(name="Guest"):
return f"Hello, {name}!"
```

Loops

- For loop: iterate over sequence.
- While loop: run while condition is True.

Example:

```
for i in range(3): print(i)
count=0
while count<3:
print(count)
count+=1
```

requests Library

- Send HTTP requests (GET, POST, PUT, DELETE).
- Used for REST APIs.

Example GET:

```
res = requests.get("https://api.example.com")
print(res.json())
```

Example POST:

```
res = requests.post(url, json={"key":"value"})
```

uuid Library

- Generate Universally Unique Identifiers.
- Versions: v1 (time), v4 (random), v5 (deterministic).

Example:

```
import uuid
print(uuid.uuid4())
```

random Library

- Generate pseudo-random numbers and choices.

Example:

```
random.randint(1,10)
random.choice(["A","B","C"])
random.shuffle(list)
```

time Library

- Work with time functions.

Example:

```
time.time() # timestamp
time.sleep(2) # pause
time.strftime("%Y-%m-%d")
```

json Library

- Work with JSON data.

Example:

```
data={"name":"Sakshi"}
s=json.dumps(data)
obj=json.loads(s)
```

REST API Basics

- Communication over HTTP.
- Methods:

GET → retrieve data
POST → create data
PUT/PATCH → update
DELETE → remove

Example GET:

```
res=requests.get("https://jsonplaceholder.typicode.com/posts/1")
print(res.json())
```

Example POST:

```
data={"title":"Hello","body":"World"}
res=requests.post(url,json=data)
print(res.json())
```