

Day 8 Notes

Hashing & Python Built-in Methods

1. Hashing

- **Never store passwords directly in the database.**
- Hashing converts a password into a **fixed, unreadable value** using a mathematical function.
- The output is called a **hash value**.
- Hashing is **one-way**.
- You **cannot convert hash back to the original password**.

Example:

Password: sriya

Hashed Value: \$2b\$12\$Kx3...

Hashing vs Encoding vs Encryption

Feature	Hashing	Encoding	Encryption
Reversible?	No	Yes	Yes
Purpose	<i>Security (passwords)</i>	<i>Data formatting</i>	<i>Secure communication</i>
Example	<i>bcrypt</i>	<i>Base64</i>	<i>AES</i>

- **Encoding & Encryption** can return to original form.
 - **Hashing cannot** be reversed.
-

bcrypt in Python

```
pip install bcrypt      #install
import bcrypt      #import
password = "sriya"      #hash a password
```

```
password_bytes = password.encode('utf-8')      #Convert string to bytes
salt = bcrypt.gensalt()      #Generate salt
hashed = bcrypt.hashpw(password_bytes, salt)      # Hash password
print(hashed)
```

Salt

- Salt = Random value added before hashing.
 - Prevents attackers from guessing passwords easily.
 - Even the same passwords will generate different hashes due to salt.
-

2. Built-in Methods in Strings

1. `strip()`

Removes spaces from both sides.

```
text = "    Hi    "
print(text.strip())
```

2. `join()`

Joins elements using a string.

```
arr = ["a", "d", "s"]
print("".join(arr))      # ads
```

3. `replace()`

Replaces a word in a string.

```
text = "I like Java"
print(text.replace("Java", "Python"))
```

4. `startswith()`

Checks if the string starts with something.

```
text = "Hello"  
print(text.startswith("He")) # True
```

5. `endswith()`

Checks if string ends with something.

```
text = "Hello"  
print(text.endswith("lo")) # True
```

6. `find()`

Finds the index of a word.

Returns **-1 if not found**.

```
text = "Python"  
print(text.find("th")) # 2
```

7. `isdigit()`

Checks if all characters are digits.

```
print("123".isdigit()) # True
```

8. `isalpha()`

Checks if all characters are alphabets.

```
print("Hello".isalpha()) # True
```

3. Built-in Methods in Dictionary

1. `get()`

Gets value of a key.

```
d = {"name": "Sriya"}  
print(d.get("name"))
```

2. `keys()`

Returns all keys.

```
print(d.keys())
```

3. `values()`

Returns all values.

```
print(d.values())
```

4. `items()`

Returns key-value pairs.

```
print(d.items())
```

5. `update()`

Updates dictionary.

```
d.update({"age": 20})
```

6. `pop()`

Removes specific key.

```
d.pop("age")
```

7. `popitem()`

Removes last inserted key-value pair.

```
d.popitem()
```

8. `clear()`

Removes entire dictionary.

```
d.clear()
```

4. Built-in Methods in Sets

1. `add()`

Adds element to set.

```
s = {1, 2}  
s.add(3)
```

2. `remove()`

Removes element.

⚠ Shows error if element not present.

```
s.remove(2)
```

3. `discard()`

Same as remove but **no error** if element not present.

```
s.discard(5)
```

4. `pop()`

Removes random element.

```
s.pop()
```

5. `clear()`

Removes all elements.

```
s.clear()
```

6. `union()`

Joins two sets and removes duplicates.

```
a = {1,2}
b = {2,3}
print(a.union(b)) # {1,2,3}
```

7. `intersection()`

Returns common elements.

```
print(a.intersection(b)) # {2}
```

8. `difference()`

Returns elements present in first set but not second.

```
print(a.difference(b)) # {1}
```