Lecture Notes

**Unit 2: **

2.1 Strings

A string is a sequence of characters enclosed in quotes. In Python, strings can be enclosed in single ('), double ("), or triple quotes (''', """).

Strings in Python are immutable, meaning once a string is created, its characters cannot be changed.

2.1.1 Creating and Storing Strings

- Single and Double Quotes:

```
str1 = 'Hello'
str2 = "World"
```

- Triple Quotes: Used for multi-line strings or strings that contain both single and double quotes.

```
str3 = '''This is a multi-line string'''
```

- Empty Strings: An empty string is created by using two quotes with nothing in between.

```
empty_str = ""
```

2.1.2 Basic String Operations

- Concatenation: Joining two or more strings using the + operator.

```
greeting = str1 + " " + str2
# Output: "Hello World"
```

- Repetition: Repeating a string multiple times using the * operator.

```
repeated_str = str1 * 3  # Output: "HelloHelloHello"
```

- Length: Finding the length of a string using the len() function.

```
length = len(greeting)  # Output: 11
```

2.1.4 Accessing Characters in a String by Index Number

- Positive Indexing: Index starts from 0 on the left side.

```
first_char = greeting[0]
# Output: 'H'
```

- Negative Indexing: Index starts from -1 on the right side.

```
last_char = greeting[-1]  # Output: 'd'
```

Index Out of Range: Accessing an index that doesn't exist will result in an IndexError.

2.1.5 String Slicing and Joining

- Slicing: Extracting a substring from a string using the : operator.

```
sub_str = greeting[0:5]  # Output: 'Hello'
sub_str = greeting[6:]   # Output: 'World'
sub_str = greeting[:5]   # Output: 'Hello'
```

- Step in Slicing: Define the step size for slicing.

```
step_str = greeting[0:11:2]  # Output: 'HloWrd'
```

- Joining: Combining elements of a sequence into a single string using a separator.

```
words = ['Hello', 'World']
joined_str = " ".join(words)  # Output: 'Hello World'
```

2.1.5 String Methods

Common Methods:

- upper(): Converts all characters in the string to uppercase.

```
greeting.upper()  # Output: 'HELLO WORLD'
```

- lower(): Converts all characters in the string to lowercase.

```
greeting.lower()  # Output: 'hello world'
```

- strip(): Removes leading and trailing whitespaces.

```
" Hello ".strip()  # Output: 'Hello'
```

- replace(): Replaces a substring with another substring.

```
greeting.replace('World', 'Python')  # Output: 'Hello Python'
```

- split(): Splits a string into a list of substrings based on a delimiter.

```
greeting.split(' ')  # Output: ['Hello', 'World']
```

- find(): Returns the index of the first occurrence of a substring.

```
greeting.find('World')  # Output: 6
```

2.1.6 Formatting Strings

- Using format() method:

```
name = "Alice"
age = 25
intro = "My name is {} and I am {} years old.".format(name, age)
# Output: 'My name is Alice and I am 25 years old.'
```

- F-Strings (Python 3.6+):

```
intro = f"My name is {name} and I am {age} years old."
# Output: 'My name is Alice and I am 25 years old.'
```

- Percentage (%) Formatting:

```
intro = "My name is %s and I am %d years old." % (name, age)
# Output: 'My name is Alice and I am 25 years old.'
```