ikSaan.com

Working with Strings in Python

Strings are one of the most commonly used data types in Python. A string is a sequence of characters enclosed in either single, double, or triple quotes.

String Creation

You can create strings using single ('), double ("), or triple quotes (''' or """).

```python
single_quote_str = 'Hello'
double_quote_str = "World"
triple_quote_str = '''This is a multiline string.'''
```

ikSaan.com

Working with Strings in Python

Accessing Characters in a String

You can access individual characters of a string using indexing. The index starts at 0 for the first character and goes up.

```python
my_string = "Hello"
print(my_string[0])  # Output: H
print(my_string[4])  # Output: o
```

ikSaan.com

Working with Strings in Python

String Slicing

```python
my_string = "Hello, World"
print(my_string[0:5])  # Output: Hello
print(my_string[7:])   # Output: World
print(my_string[:5])   # Output: Hello
print(my_string[::2])  # Output: Hlo ol
```

ikSaan.com

Working with Strings in Python

String Concatenation and Repetition

You can concatenate two strings using the + operator and repeat strings using the * operator.

```python
greeting = "Hello" + " " + "World"
print(greeting)  # Output: Hello World

repeat_str = "Hi " * 3
print(repeat_str)  # Output: Hi Hi Hi
```

ikSaan.com

Working with Strings in Python

String Methods

Python provides various built-in string methods to manipulate and analyze strings.

- **lower(): Converts all characters to lowercase.**
- **upper(): Converts all characters to uppercase.**
- **capitalize(): Capitalizes the first character.**
- **title(): Capitalizes the first letter of each word.**

```python
my_string = "hello world"
print(my_string.upper())  # Output: HELLO WORLD
print(my_string.capitalize())  # Output: Hello world
print(my_string.title())  # Output: Hello World
```

ikSaan.com

Working with Strings in Python

Stripping Whitespace

- **strip(): Removes leading and trailing spaces.**
- **lstrip(): Removes leading spaces.**
- **rstrip(): Removes trailing spaces.**

```python
my_string = "   Hello   "
print(my_string.strip())  # Output: Hello
```

ikSaan.com

Working with Strings in Python

Finding Substrings

- **find(): Returns the index of the first occurrence of a substring. Returns -1 if not found.**
- **count(): Returns the number of occurrences of a substring.**
- **startswith(), endswith(): Check if a string starts or ends with a certain substring.**

```python
my_string = "hello world"
print(my_string.find("world"))  # Output: 6
print(my_string.count("l"))     # Output: 3
print(my_string.startswith("he"))  # Output: True
```

ikSaan.com

Working with Strings in Python

Replacing Substrings

replace(): Replaces occurrences of a substring with another substring.

```python
my_string = "hello world"
new_string = my_string.replace("world", "Python")
print(new_string)  # Output: hello Python
```

ikSaan.com

Working with Strings in Python

String Formatting

Python offers several ways to format strings.

f-Strings (Formatted String Literals)
Introduced in Python 3.6, f-strings allow you to embed expressions inside string literals using curly braces {}.

python

```python
name = "Alice"
age = 25
print(f"My name is {name} and I am {age} years old.")
```

ikSaan.com

Working with Strings in Python

format() Method

Another way to format strings is using the format() method.

```python
name = "Bob"
age = 30
print("My name is {} and I am {} years old.".format(name, age))
```

ikSaan.com

Working with Strings in Python

Splitting and Joining Strings

Splitting Strings

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

```python
my_string = "apple, banana, cherry"
fruits = my_string.split(", ")
print(fruits)  # Output: ['apple', 'banana', 'cherry']
```

ikSaan.com

Working with Strings in Python

Joining Strings

join(): Joins elements of a list into a single string, with a delimiter between them.

```python
fruits = ['apple', 'banana', 'cherry']
fruit_string = ", ".join(fruits)
print(fruit_string)  # Output: apple, banana, cherry
```

ikSaan.com

Working with Strings in Python

Escape Sequences

In Python, escape sequences are used to include special characters in a string.

Common escape sequences:

- **\n: Newline**
- **\t: Tab**
- **\': Single quote**
- **\": Double quote**
- **\\: Backslash**

```python
print("Hello\nWorld")  # Output: Hello (newline) World
print("This is a tab:\tTab!")  # Output: This is a tab:    Tab!
```

ikSaan.com

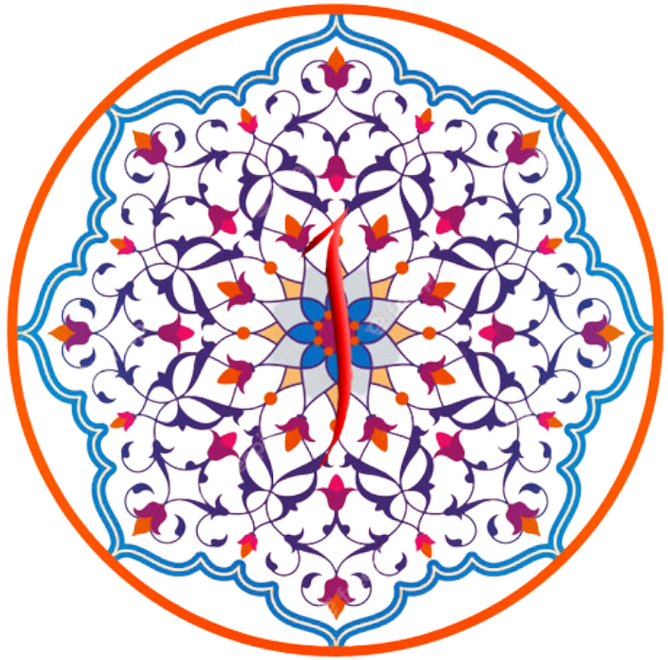
Working with Strings in Python

Raw Strings

A raw string ignores escape sequences. It's created by prefixing the string with r.

```python
raw_string = r"C:\Users\Name"
print(raw_string)  # Output: C:\Users\Name
```

ikSaan.com



ikSaan.com

Strings are a fundamental part of Python, and mastering them will significantly improve your coding skills. Would you like to explore more advanced string topics like regular expressions, or work with string data in a specific project?