Learn Python Programming from Scratch

Topic: String Operations in Python

1. What are String Operations?

String operations are methods and techniques used to manipulate, analyze, and work with text data in Python. Strings are sequences of characters, and Python provides powerful built-in methods to process them efficiently.

String operations enable you to:

- Modify text (uppercase, lowercase, replace)
- Extract information (find, count, slice)
- Validate data (check if numeric, alphabetic)
- Format output (alignment, padding)
- Split and join text (parse data, create sentences)

2. Why String Operations are Important

String operations are fundamental because:

- **Text Processing**: Most programs work with text data
- **Data Cleaning**: Clean and normalize user input
- **User Interface**: Format output for better presentation
- **Data Analysis**: Parse and extract information from text
- **File Processing**: Read and manipulate file contents
- **Web Development**: Handle URLs, forms, and HTML content

3. Categories of String Operations

Basic Operations:

- Concatenation (`+`), Repetition (`*`), Length (`len()`)

Case Conversion:

- `upper()` , `lower()` , `title()` , `capitalize()` , `swapcase()`

Search and Replace:

- `find()` , `index()` , `count()` , `replace()` , `startswith()` , `endswith()`

Splitting and Joining:

- `split()` , `rsplit()` , `splitlines()` , `join()` , `partition()`

Formatting and Alignment:

- `center()`, `ljust()`, `rjust()`, `zfill()`, `format()`, f-strings

Validation and Testing:

- `isdigit()`, `isalpha()`, `isalnum()`, `isspace()`, `isupper()`, `islower()`

Whitespace Handling:

- `strip()`, `lstrip()`, `rstrip()`

In [1]:

```python
# Comprehensive String Operations Examples

print("=== BASIC STRING OPERATIONS ===")

# Sample strings for demonstration
text = "Python Programming"
user_input = "  Hello World  "
email = "user@example.com"
sentence = "The quick brown fox jumps over the lazy dog"

print(f"Original text: '{text}'")
print(f"Length: {len(text)}")
print(f"Upper case: {text.upper()}")
print(f"Lower case: {text.lower()}")
print(f"Title case: {text.title()}")
print(f"Swapped case: {text.swapcase()}")

print("\n=== SEARCH AND REPLACE ===")
print(f"Find 'Python': {text.find('Python')}")
print(f"Count 'o': {text.count('o')}")
print(f"Replace 'Python' with 'Java': {text.replace('Python', 'Java')}")
print(f"Starts with 'Python': {text.startswith('Python')}")
print(f"Ends with 'ing': {text.endswith('ing')}")

print("\n=== SPLITTING AND JOINING ===")
words = sentence.split()
print(f"Split sentence: {words}")
print(f"Join with '-': {'-'.join(words)}")
print(f"Split email: {email.split('@')}")

print("\n=== STRING FORMATTING ===")
name = "Alice"
age = 30
score = 95.6
print(f"f-string: Hello {name}, you are {age} years old")
print(f"Format method: Score: {score:.1f}%")
print(f"Center alignment: '{name.center(20, '*')}'")

print("\n=== VALIDATION ===")
test_strings = ["123", "abc", "123abc", "   ", "ABC"]
for s in test_strings:
    print(f"'{s}' -> digit: {s.isdigit()}, alpha: {s.isalpha()}, alnum: {s.isalnu

print("\n=== WHITESPACE HANDLING ===")
print(f"Original: '{user_input}'")
print(f"Strip: '{user_input.strip()}'")
print(f"Left strip: '{user_input.lstrip()}'")
print(f"Right strip: '{user_input.rstrip()}'")

# Quick practical example
def clean_and_format_name(name):
    """Clean and format a person's name."""
    return name.strip().title()

names = ["  alice johnson  ", "BOB SMITH", "charlie brown"]
print(f"\nCleaned names: {[clean_and_format_name(name) for name in names]}")

print("\n=== ADVANCED STRING TECHNIQUES ===")
# String slicing
print(f"First 6 chars: '{text[:6]}'")
print(f"Last 11 chars: '{text[-11:]}'")
print(f"Every 2nd char: '{text[::2]}'")
print(f"Reverse string: '{text[::-1]}'")

# Multiple operations chained
processed = sentence.lower().replace('the', 'a').split()
print(f"Chained operations: {processed[:5]}...")  # Show first 5 words
```

```
=== BASIC STRING OPERATIONS ===
Original text: 'Python Programming'
Length: 18
Upper case: PYTHON PROGRAMMING
Lower case: python programming
Title case: Python Programming
Swapped case: pYTHON pROGRAMMING

=== SEARCH AND REPLACE ===
Find 'Python': 0
Count 'o': 2
Replace 'Python' with 'Java': Java Programming
Starts with 'Python': True
Ends with 'ing': True

=== SPLITTING AND JOINING ===
Split sentence: ['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'd
og']
Join with '-': The-quick-brown-fox-jumps-over-the-lazy-dog
Split email: ['user', 'example.com']

=== STRING FORMATTING ===
f-string: Hello Alice, you are 30 years old
Format method: Score: 95.6%
Center alignment: '*******Alice********'

=== VALIDATION ===
'123' -> digit: True, alpha: False, alnum: True
'abc' -> digit: False, alpha: True, alnum: True
'123abc' -> digit: False, alpha: False, alnum: True
'   ' -> digit: False, alpha: False, alnum: False
'ABC' -> digit: False, alpha: True, alnum: True

=== WHITESPACE HANDLING ===
Original: '  Hello World  '
Strip: 'Hello World'
Left strip: 'Hello World  '
Right strip: '  Hello World'

Cleaned names: ['Alice Johnson', 'Bob Smith', 'Charlie Brown']

=== ADVANCED STRING TECHNIQUES ===
First 6 chars: 'Python'
Last 11 chars: 'Programming'
Every 2nd char: 'Pto rgamn'
Reverse string: 'gnimmargorP nohtyP'
Chained operations: ['a', 'quick', 'brown', 'fox', 'jumps']...
```

Key Takeaways

- **Strings are immutable** - operations return new strings, don't modify originals
- **Case methods**: `upper()`, `lower()`, `title()`, `capitalize()`, `swapcase()`
- **Search methods**: `find()`, `index()`, `count()`, `startswith()`, `endswith()`
- **Splitting**: `split()` creates lists, `join()` combines them back

- **Validation**: `isdigit()`, `isalpha()`, `isalnum()` for data checking
- **Whitespace**: `strip()`, `lstrip()`, `rstrip()` for cleaning input
- **Slicing**: `string[start:end:step]` for extracting parts
- **Formatting**: f-strings `f"{var}"` are the modern preferred method

Common String Operations Reference

Operation	Method	Example	Result
Length	`len(s)`	`len("hello")`	`5`
Upper case	`s.upper()`	`"hello".upper()`	`"HELLO"`
Find substring	`s.find(sub)`	`"hello".find("ll")`	`2`
Replace	`s.replace(old, new)`	`"hello".replace("l", "x")`	`"hexxo"`
Split	`s.split()`	`"a b c".split()`	`['a', 'b', 'c']`
Strip whitespace	`s.strip()`	`" hi ".strip()`	`"hi"`
Check digits	`s.isdigit()`	`"123".isdigit()`	`True`

Practice Exercises

1. **Text Processor**: Create a function that cleans and formats user input
2. **Word Counter**: Count words, characters, and sentences in text
3. **Email Validator**: Check if a string looks like a valid email
4. **Password Checker**: Validate password strength using string methods
5. **Text Analyzer**: Find most common words and characters in text

---

Course Information

Learn Python Programming from Scratch

Author: Prakash Ukhalkar

Topic: Python Fundamentals - String Operations

Built with ❤️ for the Python learning community