

# Text Processing Services

The modules described in this chapter provide a wide range of string manipulation operations and other text processing services.

The [codecs](#) module described under [Binary Data Services](#) is also highly relevant to text processing. In addition, see the documentation for Python's built-in string type in [Text Sequence Type — str](#).

- [string](#) — Common string operations
  - [String constants](#)
  - [Custom String Formatting](#)
  - [Format String Syntax](#)
    - [Format Specification Mini-Language](#)
    - [Format examples](#)
  - [Template strings](#)
  - [Helper functions](#)
- [re](#) — Regular expression operations
  - [Regular Expression Syntax](#)
  - [Module Contents](#)
  - [Regular Expression Objects](#)
  - [Match Objects](#)
  - [Regular Expression Examples](#)
    - [Checking for a Pair](#)
    - [Simulating scanf\(\)](#)
    - [search\(\) vs. match\(\)](#)
    - [Making a Phonebook](#)
    - [Text Munging](#)
    - [Finding all Adverbs](#)
    - [Finding all Adverbs and their Positions](#)
    - [Raw String Notation](#)
    - [Writing a Tokenizer](#)
- [difflib](#) — Helpers for computing deltas
  - [SequenceMatcher Objects](#)
  - [SequenceMatcher Examples](#)
  - [Differ Objects](#)
  - [Differ Example](#)
  - [A command-line interface to difflib](#)
- [textwrap](#) — Text wrapping and filling
- [unicodedata](#) — Unicode Database
- [stringprep](#) — Internet String Preparation
- [readline](#) — GNU readline interface
  - [Init file](#)
  - [Line buffer](#)
  - [History file](#)
  - [History list](#)

- [Startup hooks](#)
  - [Completion](#)
  - [Example](#)
- [rlcompleter](#) — Completion function for GNU readline
  - [Completer Objects](#)