Data Types

The modules described in this chapter provide a variety of specialized data types such as dates and times, fixed-type arrays, heap queues, double-ended queues, and enumerations.

Python also provides some built-in data types, in particular, `dict`, `list`, `set` and `frozenset`, and `tuple`. The `str` class is used to hold Unicode strings, and the `bytes` and `bytearray` classes are used to hold binary data.

The following modules are documented in this chapter:

- `datetime` — Basic date and time types
  - Aware and Naive Objects
  - Constants
  - Available Types
    - Common Properties
    - Determining if an Object is Aware or Naive
  - `timedelta` Objects
    - Examples of usage: `timedelta`
  - `date` Objects
    - Examples of Usage: `date`
  - `datetime` Objects
    - Examples of Usage: `datetime`
  - `time` Objects
    - Examples of Usage: `time`
  - `tzinfo` Objects
  - `timezone` Objects
  - `strftime()` and `strptime()` Behavior
    - `strftime()` and `strptime()` Format Codes
    - Technical Detail
- `zoneinfo` — IANA time zone support
  - Using `ZoneInfo`
  - Data sources
    - Configuring the data sources
      - Compile-time configuration
      - Environment configuration
      - Runtime configuration
  - The `ZoneInfo` class
    - String representations
    - Pickle serialization
  - Functions
  - Globals
  - Exceptions and warnings
- `calendar` — General calendar-related functions
- `collections` — Container datatypes
  - `ChainMap` objects
    - `ChainMap` Examples and Recipes
  - `Counter` objects
  - `deque` objects
    - `deque` Recipes

        - defaultdict Examples
    - namedtuple() Factory Function for Tuples with Named Fields
    - OrderedDict objects
        - OrderedDict Examples and Recipes
    - UserDict objects
    - UserList objects
    - UserString objects
- collections.abc — Abstract Base Classes for Containers
    - Collections Abstract Base Classes
    - Collections Abstract Base Classes – Detailed Descriptions
    - Examples and Recipes
- heapq — Heap queue algorithm
    - Basic Examples
    - Priority Queue Implementation Notes
    - Theory
- bisect — Array bisection algorithm
    - Performance Notes
    - Searching Sorted Lists
    - Examples
- array — Efficient arrays of numeric values
- weakref — Weak references
    - Weak Reference Objects
    - Example
    - Finalizer Objects
    - Comparing finalizers with __del__() methods
- types — Dynamic type creation and names for built-in types
    - Dynamic Type Creation
    - Standard Interpreter Types
    - Additional Utility Classes and Functions
    - Coroutine Utility Functions
- copy — Shallow and deep copy operations
- pprint — Data pretty printer
    - PrettyPrinter Objects
    - Example
- reprlib — Alternate repr() implementation
    - Repr Objects
    - Subclassing Repr Objects
- enum — Support for enumerations
    - Module Contents
    - Creating an Enum
    - Programmatic access to enumeration members and their attributes
    - Duplicating enum members and values
    - Ensuring unique enumeration values
    - Using automatic values
    - Iteration
    - Comparisons
    - Allowed members and attributes of enumerations
    - Restricted Enum subclassing
    - Pickling
    - Functional API

        - IntFlag
        - Flag
        - Others
    - When to use `__new__()` vs. `__init__()`
    - Interesting examples
        - Omitting values
            - Using `auto`
            - Using `object`
            - Using a descriptive string
            - Using a custom `__new__()`
        - OrderedEnum
        - DuplicateFreeEnum
        - Planet
        - TimePeriod
    - How are Enums different?
        - Enum Classes
        - Enum Members (aka instances)
        - Finer Points
            - Supported `__dunder__` names
            - Supported `_sunder_` names
            - _Private__names
            - `Enum` member type
            - Boolean value of `Enum` classes and members
            - `Enum` classes with methods
            - Combining members of `Flag`
- `graphlib` — Functionality to operate with graph-like structures
    - Exceptions

"