

# 数据类型

本章所描述的模块提供了许多专门的数据类型，如日期和时间、固定类型的数组、堆队列、双端队列、以及枚举。

Python也提供一些内置数据类型，特别是，[dict](#)、[list](#)、[set](#)、[frozenset](#)、以及[tuple](#)。[str](#) 这个类是用来存储Unicode字符串的，而[bytes](#) 和[bytearray](#) 这两个类是用来存储二进制数据的。

本章包含以下模块的文档：

- [datetime](#) --- 基本日期和时间类型
  - 感知型对象和简单型对象
  - 常量
  - 有效的类型
    - 通用的特征属性
    - 确定一个对象是感知型还是简单型
  - [timedelta](#) 类对象
    - [timedelta](#) 用法示例
  - [date](#) 对象
    - [date](#) 用法示例
  - [datetime](#) 对象
    - 用法示例: [datetime](#)
  - [time](#) 对象
    - 用法示例: [time](#)
  - [tzinfo](#) 对象
  - [timezone](#) 对象
  - [strftime\(\)](#) 和 [strptime\(\)](#) 的行为
    - [strftime\(\)](#) 和 [strptime\(\)](#) 格式码
    - 技术细节
- [zoneinfo](#) --- IANA 时区支持
  - 使用 [ZoneInfo](#)
  - 数据源
    - 配置数据源
      - 编译时配置
      - 环境配置
      - 运行时配置
  - [ZoneInfo](#) 类
    - 字符串表示
    - 封存序列化
  - 函数
  - 全局变量

- 异常与警告
- `calendar` --- 通用日历相关函数
  - 命令行用法
- `collections` --- 容器数据类型
  - `ChainMap` 对象
    - `ChainMap` 例子和方法
  - `Counter` 对象
  - `deque` 对象
    - `deque` 用法
  - `defaultdict` 对象
    - `defaultdict` 例子
  - `namedtuple()` 命名元组的工厂函数
  - `OrderedDict` 对象
    - `OrderedDict` 例子和用法
  - `UserDict` 对象
  - `UserList` 对象
  - `UserString` 对象
- `collections.abc` --- 容器的抽象基类
  - 容器抽象基类
  - 多项集抽象基类 -- 详细描述
  - 例子和配方
- `heapq` --- 堆队列算法
  - 基本示例
  - 优先队列实现说明
  - 理论
- `bisect` --- 数组二分算法
  - 性能说明
  - 搜索有序列表
  - 例子
- `array` --- 高效的数字值数组
- `weakref` --- 弱引用
  - 弱引用对象
  - 示例
  - 终结器对象
  - 比较终结器与 `__del__()` 方法
- `types` --- 动态类型创建和内置类型名称
  - 动态类型创建
  - 标准解释器类型
  - 附加工具类和函数
  - 协程工具函数
- `copy` --- 浅层及深层拷贝操作
- `pprint` --- 数据美化输出
  - 函数

- PrettyPrinter 对象
- 示例
- reprlib --- 替代性 repr() 实现
  - Repr 对象
  - 子类化 Repr 对象
- enum --- 对枚举的支持
  - 模块内容
  - 数据类型
    - 支持的 \_\_dunder\_\_ 名称
    - 支持的 \_\_sunder\_\_ 名称
  - 工具与装饰器
  - 备注
- graphlib --- 操作类似图的结构的功能
  - 异常