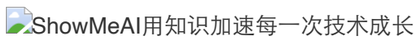
图解 Python 编程(27) | 时间和日期

2021-11-09 1122 0 工具教程 python 编程语言



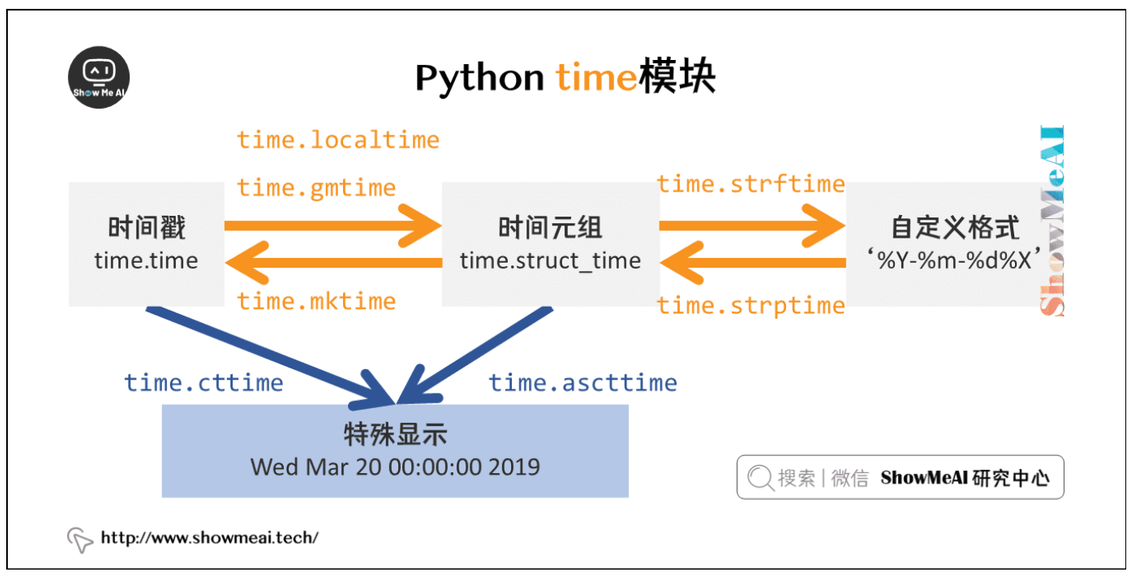
作者：韩信子@ShowMeAI
教程地址：https://www.showmeai.tech/tutorials/56
本文地址：https://www.showmeai.tech/article-detail/90

Python日期与时间

在Python的开发过程中，我们经常要处理时间类型的数据，Python内置了 `time` ， `datetime` 等标准库，帮助我们对时间型进行处理，在本节内容中，我们将围绕这两个标准库，详细展开介绍常用方法。

time模块

在Python中， `time` 模块主要用于将时间戳转换为具体的日期时间，但 `time` 模块表示日期时间的对象结构简单，不适合进行复杂的操作和表示。



模块用法

`time` 模块中只有 `time.struct_time` 一个类：

`struct_time` 是一个转换秒数得到的结构化的时间对象，可以通过下标或属性名称获取对象的年月日时分秒等属性。调用 `gmtime()` ， `localtime()` ， `strptime()` 等方法可得到 `struct_time` 实例。

```
>>> st = time.localtime()
>>> st
time.struct_time(tm_year=2021, tm_mon=10, tm_mday=27, tm_hour=19, tm_min=27, tm_sec=31,
tm_wday=2, tm_yday=300, tm_isdst=0)
>>> st.tm_mon
10
>>> st[1]
10
```

```
# 在struct_time和字符串之间进行转换
>>> time.strftime('%H:%M:%S')
'19:10:37'

>>> time.strptime("30 Nov 00", "%d %b %y")
time.struct_time(tm_year=2000, tm_mon=11, tm_mday=30, tm_hour=0, tm_min=0,
                 tm_sec=0, tm_wday=3, tm_yday=335, tm_isdst=-1)
```

```
import time

# 格式化成2021-10-27 19:56:36形式
print(time.strftime("%Y-%m-%d %H:%M:%S", time.localtime()))

# 格式化成Sat Mar 28 22:24:24 2016形式
print(time.strftime("%a %b %d %H:%M:%S %Y", time.localtime()))

# 将格式字符串转换为时间戳
a = "Wed Oct 27 19:56:36 2021"
print(time.mktime(time.strptime(a,"%a %b %d %H:%M:%S %Y")))
```

datetime模块

`datetime` 模块支持日期和时间的运算，它提供了一些用于操作日期和时间的类。该模块的绝大部分功能都围绕着以下 4 个类（以及另外两个关于时区的类）的方法和属性来实现。

date 类及用法

date 类表示日期类型。

支持的操作符：

- 支持与另一 date 对象进行 ==，≤，<，≥，> 等比较操作。
- 支持与 timedelta 对象进行加减操作，结果依然为 date 对象。
- 支持与另一 date 对象进行相减操作，得到 timedelta 对象。
- 支持哈希。

代码示例：

```
# 传入日期对应的年月日参数，实例化date类
>>> from datetime import date
>>> date(2021, 10, 29)
datetime.date(2021, 10, 29)

# 可以通过时间戳获得时间
>>> date.fromtimestamp(time.time())
datetime.date(2021, 10, 29)

>>> d2 = date(2021, 10, 29)
>>> d1 = date(2021, 10, 27)
>>> d2 > d1
True
>>> d2 - d1
datetime.timedelta(days=2)
```

time 类及用法

time 类表示时间（时分秒）类型。

支持的操作符

- 支持与另一 time 对象进行 ==，≤，<，≥，> 等比较操作。
- 支持哈希。

代码示例

```
>>> from datetime import time
>>> t = time.fromisoformat('19:32:10')
>>> t.strftime('%Hh %Mm %Ss')
'19h 32m 10s'

>>> t = time(hour=19, minute=27, second=55)
>>> t.isoformat()
'19:27:55'
```

datetime 类及用法

datetime 类表示包含日期时分的时间类型，可以视作 date 和 time 实例的组合体，因此同时具备了两种对象的大部分方法和属性。

支持的操作符

- datetime 支持与 date 进行相等比较，但结果一定为 False ，除此之外只支持与另一 datetime 对象执行 ==，≤，<，≥，> 等比较操作。
- 支持与 timedelta 相加，结果为 datetime；支持与 timedelta 对象进行加减，结果依然为 datetime 对象，与另一 datetime 对象进行相减，得到 timedelta 对象。
- 同样支持哈希。

代码示例

```
>>> from datetime import datetime
>>> datetime(year=2021, month=10, day=29)
datetime.datetime(2021, 10, 29, 0, 0)

>>> datetime.now()
datetime.datetime(2021, 10, 29, 14, 51, 18, 731235)

>>> datetime.fromisoformat('2021-10-29 16:09:32')
datetime.datetime(2021, 10, 29, 16, 9, 32)

>>> dt = datetime.now()
>>> dt.timestamp()
1635317544.682565

>>> dt.date()
datetime.date(2021, 10, 29)
```

timedelta

timedelta 类对象表示两个 datetime 对象之间的差异。

支持的操作符

- 只支持与另一 timedelta 进行比较，进行 ==，≤，<，≥，> 等比较操作。
- timedelta 对象支持支持加减操作，datetime 与 timedelta 相加或相减仍然返回 datetime 。
- timedelta 还支持乘除模除等操作符。

- 支持哈希。
- `timedelta` 是有符号的，支持 `abs()` 函数，可返回两个 `datetime` 之间的绝对间隔。

代码示例

```
>>> from datetime import timedelta
>>> timedelta(days=2)
datetime.timedelta(days=2)

>>> dt1 = datetime.now()
>>> dt2 = datetime.now()
>>> dt2 -dt1
datetime.timedelta(seconds=4, microseconds=476390)

>>> d = timedelta(minutes=3, seconds=35)
>>> d.total_seconds()
215.0
```

一键运行所有代码

图解Python编程系列 配套的所有代码，可前往ShowMeAI 官方 **GitHub**，下载后即可在本地 Python 环境中运行。能访问 Google 的宝宝也可以直接借助 Google Colab一键运行与交互学习！

下载Python要点速查表

Awesome cheatsheets | **ShowMeAI速查表大全** 系列包含『编程语言』『AI技能知识』『数据科学工具库』『AI垂直领域工具库』四个板块，追平到工具库当前最新版本，并跑通了所有代码。点击 **官网** 或 **GitHub** 获取~

👇 **Python 速查表（部分）**

拓展参考资料

- Python教程 - Python3文档
- Python教程 - 廖雪峰的官方网站

ShowMeAI图解Python编程系列推荐（要点速查版）

- ShowMeAI 图解 Python 编程(1) | 介绍
- ShowMeAI 图解 Python 编程(2) | 安装与环境配置
- ShowMeAI 图解 Python 编程(3) | 基础语法
- ShowMeAI 图解 Python 编程(4) | 基础数据类型
- ShowMeAI 图解 Python 编程(5) | 运算符
- ShowMeAI 图解 Python 编程(6) | 条件控制与if语句
- ShowMeAI 图解 Python 编程(7) | 循环语句
- ShowMeAI 图解 Python 编程(8) | while循环
- ShowMeAI 图解 Python 编程(9) | for循环
- ShowMeAI 图解 Python 编程(10) | break语句
- ShowMeAI 图解 Python 编程(11) | continue语句
- ShowMeAI 图解 Python 编程(12) | pass语句
- ShowMeAI 图解 Python 编程(13) | 字符串及操作
- ShowMeAI 图解 Python 编程(14) | 列表
- ShowMeAI 图解 Python 编程(15) | 元组
- ShowMeAI 图解 Python 编程(16) | 字典
- ShowMeAI 图解 Python 编程(17) | 集合
- ShowMeAI 图解 Python 编程(18) | 函数
- ShowMeAI 图解 Python 编程(19) | 迭代器与生成器
- ShowMeAI 图解 Python 编程(20) | 数据结构
- ShowMeAI 图解 Python 编程(21) | 模块
- ShowMeAI 图解 Python 编程(22) | 文件读写
- ShowMeAI 图解 Python 编程(23) | 文件与目录操作
- ShowMeAI 图解 Python 编程(24) | 错误与异常处理
- ShowMeAI 图解 Python 编程(25) | 面向对象编程
- ShowMeAI 图解 Python 编程(26) | 命名空间与作用域
- ShowMeAI 图解 Python 编程(27) | 时间和日期

ShowMeAI系列教程精选推荐

- 大厂技术实现：推荐与广告计算解决方案
- 大厂技术实现：计算机视觉解决方案
- 大厂技术实现：自然语言处理行业解决方案
- 图解Python编程：从入门到精通系列教程
- 图解数据分析：从入门到精通系列教程
- 图解AI数学基础：从入门到精通系列教程
- 图解大数据技术：从入门到精通系列教程
- 图解机器学习算法：从入门到精通系列教程
- 机器学习实战：手把手教你玩转机器学习系列
- 深度学习教程：吴恩达专项课程 · 全套笔记解读

- 自然语言处理教程：斯坦福CS224n课程 · 课程带学与全套笔记解读
- 深度学习与计算机视觉教程：斯坦福CS231n · 全套笔记解读

图解 Python 编程(27) | 时间和日期

〈 上一篇
图解 Python 编程(26) | 命名空间与作用域

下一篇 〉
图解Python编程：从入门到精通系列教程