

15章 日付・時刻の処理を理解しよう

15章 日付・時刻の処理を理解しよう

| Pythonで日付や時刻を取得する方法を学びます

本章の目標

- Pythonで日付や時刻を取得する方法を知ること
- datetimeモジュールとtimedelta関数を使って日時の差を比較したり、計算したりする方法を知ること

15章 なぜ日時の処理を学ぶのか

| プログラミングでは日付や時刻を取得・計算する機会が多いです

- 学習記録アプリ
- スケジュール管理アプリ
- 日時の操作がメインになることも多い

15章 日時を取得する2つの方法

| Pythonで日時を取得する主な方法は2つあります

モジュール	用途
timeモジュール	シンプルな時刻取得・経過時間計算
datetimeモジュール	日時の取得・比較・計算

モジュール=日時を扱うための便利な部品（次章で詳しく解説）

| 現在の時刻を取得できます

```
# timeモジュールをインポート
import time

# 現在時刻をUNIXタイムスタンプで取得
print(time.time())

# 現在時刻を日時のフォーマットで取得
print(time.strftime("%Y年%m月%d日%H時%M分%S秒", time.localtime()))
```

15章 日時のフォーマット

| さまざまな表示形式で日時を表示できます

- 「2015年3月19日」
- 「2015-03-19」
- 「2015/03/19 12:30:00」

strftime関数を使ってフォーマットを指定します

15章 フォーマット文字一覧

日時のフォーマットを指定する特殊な文字

文字	意味	例
%Y	年 (4桁)	2015、2022
%m	月 (01~12)	01、12
%d	日 (01~31)	01、31
%H	時 (00~23)	00、23
%M	分 (00~59)	00、59
%S	秒 (00~59)	00、59

Visual Studio Codeで日本時間を使う場合

```
!rm /etc/localtime  
!ln -s /usr/share/zoneinfo/Asia/Tokyo /etc/localtime  
!date
```

- タイムゾーン=同じ標準時を使う地域のこと
- 日本はAsia/Tokyo

15章 タイムゾーンの設定



The screenshot shows a terminal window with a light gray background. On the left side, there is a vertical toolbar with a green checkmark icon and a play button icon. Below the toolbar, the text "0 秒" is displayed. The main area of the terminal contains three commands in blue text:
!rm /etc/localtime
!ln -s /usr/share/zoneinfo/Asia/Tokyo /etc/localtime
!date
The command "!date" is partially cut off at the bottom. At the bottom of the terminal window, the system date and time are shown in gray text: "Sat Sep 24 23:02:24 JST 2022".

15章 timeモジュールで日時を取得

| フォーマットを指定して現在の日時を取得

```
# timeモジュールをインポート
import time

# 現在の日時を指定したフォーマットで出力する
print(time.strftime("%Y年%m月%d日%H時%M分%S秒", time.localtime()))
```

15章 timeモジュールで日時を取得

✓
0
秒



```
# timeモジュールをインポート
import time

# 現在の日時を指定したフォーマットで出力する
print(time.strftime("%Y年%m月%d日%H時%M分%S秒", time.localtime()))
```

2022年09月24日23時46分54秒

1970年1月1日0時0分0秒からの経過秒数

- 例：「1426690800」「1652662122」
- インターネットの世界で共通の整数型の値として扱える形式

```
import time  
  
# 現在時刻をUNIXタイムスタンプで取得  
print(time.time())
```

15章 UNIXタイムスタンプ

✓
0 秒



```
# timeモジュールをインポート  
import time  
  
# 現在時刻をUNIXタイムスタンプで取得  
print(time.time())
```

1664031600.9133644

15章 datetimeモジュール

| 日時の取得・比較・計算ができる便利なクラスが用意されています

クラス名	用途
datetime.datetime	日付と時刻
datetime.date	日付
datetime.time	時刻
datetime.timedelta	時間差・経過時間

15章 datetimeクラスの主な関数

| 日時を取得・変換するための関数

関数名	用途
now()	現在時刻を取得
strptime(文字列, フォーマット)	文字列をdatetimeに変換
strftime(フォーマット)	datetimeから文字列に変換

15章 datetimeモジュールの使い方

日時を取得・変換する

```
import datetime

# 現在の日時を取得する
now = datetime.datetime.now()

# 指定した日付を取得する
date = datetime.datetime.strptime("2015-03-19", "%Y-%m-%d")

# 指定した日時を取得する
date_time = datetime.datetime.strptime("2015-03-19 12:15:30", "%Y-%m-%d %H:%M:%S")
```

15章 datetimeモジュールを使ってみよう

日時を特定のフォーマットで出力する

```
import datetime

# 指定した日時を取得する
date_time = datetime.datetime.strptime("2015-03-19 12:15:30", "%Y-%m-%d %H:%M:%S")

# 取得した日付date_timeの日時を特定のフォーマットで出力する
print(date_time.strftime("%Y年%m月%d日%H時%M分%S秒"))
```

15章 datetimeモジュールを使ってみよう

✓
0
秒

```
# datetimeモジュールをインポート
import datetime

# 指定した日時を取得する
date_time = datetime.datetime.strptime("2015-03-19 12:15:30", "%Y-%m-%d %H:%M:%S")

# 取得した日付date_timeの日時を特定のフォーマットで出力する
print(date_time.strftime("%Y年%m月%d日%H時%M分%S秒"))
```

2015年03月19日12時15分30秒

15章 年月日時分秒を個別に取り出す

| 取得した日付の各要素にアクセスできます

```
import datetime

date_time = datetime.datetime.strptime("2015-03-19 12:15:30", "%Y-%m-%d %H:%M:%S")

print(date_time.year)      # 年
print(date_time.month)     # 月
print(date_time.day)       # 日
print(date_time.hour)      # 時
print(date_time.minute)    # 分
print(date_time.second)    # 秒
```

15章 年月日時分秒を個別に取り出す

```
✓ 0 秒
▶ # datetimeモジュールをインポート
import datetime

# 指定した日時を取得する
date_time = datetime.datetime.strptime("2015-03-19 12:15:30", "%Y-%m-%d %H:%M:%S")

print(date_time)

# 年,月,日,時,分,秒だけをそれぞれ取り出すことも可能です
print(date_time.year)
print(date_time.month)
print(date_time.day)
print(date_time.hour)
print(date_time.minute)
print(date_time.second)
```

```
2015-03-19 12:15:30
2015
3
19
12
15
30
```

15章 timedeltaクラス

| 2つの日時の差を比較できます

属性	内容
days	日数
seconds	秒数（日数分含まず）
microseconds	マイクロ秒
total_seconds	トータルの秒数

15章 2つの日時の差を比較する

datetimeオブジェクト同士を引き算するとtimedeltaオブジェクトが得られます

```
import datetime

now = datetime.datetime.now()
date_time = datetime.datetime.strptime("2015-03-19 12:15:30", "%Y-%m-%d %H:%M:%S")

# 2つのdatetimeの差を取得
interval = now - date_time

print(f"総日数は{interval.days}日です。")
```

15章 2つの日時の差を比較する

✓ 0 秒

```
# datetimeモジュールをインポート
import datetime

# 現在時刻を取得する
now = datetime.datetime.now()

# 指定した日時を取得する
date_time = datetime.datetime.strptime("2015-03-19 12:15:30", "%Y-%m-%d %H:%M:%S")

# 2つのdatetimeの差を取得し、変数intervalに代入する(interval=間隔)
interval = now - date_time

# 取得した日時の差を特定のフォーマットで出力する
print(f"総日数は{interval.days}日です。")
```

総日数は2746日です。

15章 日時を加算・減算する

| **timedeltaオブジェクトを使って日付を計算できます**

```
import datetime

now = datetime.datetime.now()

# 1年間の日数差を作成
td_1y = datetime.timedelta(days=365)

# 現在の日時に1年を加算
add = now + td_1y

# 3日間の日数差を作成
td_3d = datetime.timedelta(days=3)

# 現在の日時から3日を減算
sub = now - td_3d
```

15章 日時を加算・減算する

✓
0
秒

```
# datetimeモジュールをインポート
import datetime

# 現在時刻を取得する
now = datetime.datetime.now()

# 1年間の日数差を変数td_1yに代入する
td_1y = datetime.timedelta(days=365)

# 現在の日時に1年を加算し、変数addに代入する
add = now + td_1y

# 3日間の日数差を変数td_3dに代入する
td_3d = datetime.timedelta(days=3)

# 現在の日時から3日を減算し、変数subに代入する (sub=「減算」を意味するsubtractionの略)
sub = now - td_3d

# 加算・減算した日時を特定のフォーマットで出力する
print(add.strftime("現在から1年後は%Y年%m月%d日%H時%M分%S秒です。"))
print(sub.strftime("現在から3日前は%Y年%m月%d日%H時%M分%S秒です。")))
```

現在から1年後は2024年04月13日05時40分27秒です。
現在から3日前は2023年04月11日05時40分27秒です。

15章 timeモジュールとdatetimeモジュールの使い分け

| 用途によって使い分けます

用途	使うモジュール
現在の日時を取得	どちらでもOK
プログラムの実行時間計算	timeモジュール
日時の比較や計算	datetimeモジュール

| 本章では以下の内容を学習しました

timeモジュールとdatetimeモジュール

- timeモジュール：現在時刻、UNIXタイムスタンプを取得
- datetimeモジュール：日時の取得・比較・計算が可能
- フォーマット文字を使ってさまざまな形式で日時を表示できる

| 本章では以下の内容を学習しました

timedeltaクラス

- 2つの日時の差を比較できる
- 日付の加算・減算が可能

属性	内容
days	日数
seconds	秒数
total_seconds	トータルの秒数