

# ماژول datetime

پایتون دارای ماژول datetime است که به شما در مدیریت برچسب زمانی در کدتان کمک می‌کند. مقادیر زمان با استفاده از کلاس time نمایش داده می‌شوند. زمان‌ها ویژگی‌هایی برای ساعت، دقیقه، ثانیه و میکروثانیه دارند. همچنین می‌توانند شامل اطلاعات منطقه زمانی نیز باشند. آرگومان‌هایی برای ایجاد یک نمونه زمان اختیاری هستند، اما پیش‌فرض 0 احتمالاً همان چیزی نیست که می‌خواهید.

## time

بیاید نگاهی به این بیندازیم که چگونه می‌توانیم اطلاعات زمان را از ماژول datetime استخراج کنیم. ما می‌توانیم با مشخص کردن `datetime.time(hour,minute,second,microsecond)` یک برچسب زمانی ایجاد کنیم.

In [4]:

```
1 import datetime
2
3 t = datetime.time(4, 20, 1)
4
5 print(t)
6 print('Hour :', t.hour)
7 print('Minute :', t.minute)
8 print('Second :', t.second)
9 print('Microsecond :', t.microsecond)
10 print('tzinfo', t.tzinfo)
```

```
04:20:01
Hour : 4
Minute : 20
Second : 1
Microsecond : 0
tzinfo None
```

توجه: یک نمونه زمان فقط مقادیر زمانی را نگه می‌دارد و شامل تاریخی که با زمان مرتبط است نیست.

همچنین می‌توانیم حداقل و حداکثر مقادیری که یک زمان در روز می‌تواند داشته باشد را در ماژول بررسی کنیم:

In [5]:

```
1 print('Earliest :', datetime.time.min)
2 print('Latest :', datetime.time.max)
3 print('Resolution:', datetime.time.resolution)
```

```
Earliest : 00:00:00
Latest : 23:59:59.999999
Resolution: 0:00:00.000001
```

ویژگی‌های کلاس min و max بازتابی از محدوده معتبر زمانی در یک روز است.

## تاریخ‌ها

ماژول `datetime` (همانطور که شما حدس می‌زنید) همچنین به ما امکان می‌دهد با برچسب زمانی تاریخ کار کنیم. مقادیر تاریخ تقویم با استفاده از کلاس `date` نمایش داده می‌شوند. نمونه‌ها ویژگی‌هایی برای سال، ماه و روز دارند. بسیار آسان است که با استفاده از روش کلاس `today()` یک تاریخی که نماینده تاریخ امروز است، ایجاد کنیم.

In [10]:

```
1 today = datetime.date.today()
2
3 print(today)
4 print('ctime', today.ctime())
5 print('tuple', today.timetuple())
6 print('ordinal:', today.toordinal())
7 print('Year :', today.year)
8 print('Month:', today.month)
9 print('Day :', today.day)
```

2023-07-06

ctime Thu Jul 6 00:00:00 2023

tuple time.struct\_time(tm\_year=2023, tm\_mon=7, tm\_mday=6, tm\_hour=0, tm\_min=0, tm\_sec=0, tm\_wday=3, tm\_yday=187, tm\_isdst=-1)

ordinal: 738707

Year : 2023

Month: 7

Day : 6

همانند زمان، محدوده مقادیر تاریخی پشتیبانی شده را می‌توان با استفاده از ویژگی‌های `max` و `min` تعیین کرد.

In [11]:

```
1 print('Earliest :', datetime.date.min)
2 print('Latest :', datetime.date.max)
3 print('Resolution:', datetime.date.resolution)
```

Earliest : 0001-01-01

Latest : 9999-12-31

Resolution: 1 day, 0:00:00

یک روش دیگر برای ایجاد نمونه‌های تاریخ جدید استفاده از روش `replace()` در یک تاریخ موجود است. به عنوان مثال، شما می‌توانید سال را تغییر دهید و روز و ماه را حفظ کنید.

In [13]:

```
1 d1 = datetime.date(2023, 3, 11)
2
3 print('d1 :', d1)
4
5 d2 = d1.replace(year = 1998)
6
7 print('d2 :', d2)
```

d1 : 2023-03-11

d2 : 1998-03-11

# ریاضیات

ما می‌توانیم عملیات حسابی را بر روی اشیاء تاریخ انجام دهیم تا تفاوت‌های زمانی را بررسی کنیم. به عنوان مثال:

In [14]:

1	d1
---	----

Out[14]:

```
datetime.date(2023, 3, 11)
```

In [15]:

1	d2
---	----

Out[15]:

```
datetime.date(1998, 3, 11)
```

In [16]:

1	d1-d2
---	-------

Out[16]:

```
datetime.timedelta(days=9131)
```

این ما را به تفاوت در روزها بین دو تاریخ می‌رساند. می‌توانید از روش `timedelta` برای مشخص کردن واحدهای زمانی مختلف (روزها، دقایق، ساعت‌ها و غیره) استفاده کنید.

عالی! اکنون باید یک درک اولیه از نحوه استفاده از ماژول `datetime` در پایتون برای کار با برچسب‌های زمانی در کدتان داشته باشید!