# Datetime

## ??????

- %d ??????????????????
- %m ?????????????
- %y ???????????????????????
- %Y ??????????????????
- %H ???????????????24 ????
- %I ???????????????12 ????
- %M ?????????????
- %S ??????????????
- %f ????????????6 ????? 0?
- %A ????????????
- %a ???????????????
- %B ?????????
- %b ???????????????
- %j ????????????????????????
- %U ??????????????????????????????
- %W ??????????????????????????

## Today, Now

```
import datetime


dt_now = datetime.datetime.now()
print(dt_now)
# 2018-02-02 18:31:13.271231


print(type(dt_now))
# <class 'datetime.datetime'>


print(dt_now.year)
# 2018


print(dt_now.hour)
# 18
```

# String to Datetime

- `strptime()` : ?????????????

```
from datetime import datetime


date_str = '09-19-2022'


date_object = datetime.strptime(date_str, '%m-%d-%Y').date()
print(type(date_object))
print(date_object)  # printed in default format


# Output:
# <class 'datetime.date'>
# 2022-09-19
```

```
from datetime import datetime


time_str = '13::55::26'
time_object = datetime.strptime(time_str, '%H::%M::%S').time()
print(type(time_object))
print(time_object)


# Output:
# <class 'datetime.time'>
# 13:55:26
```

```
from datetime import datetime
import locale


locale.setlocale(locale.LC_ALL, 'de_DE')
date_str_de_DE = '16-Dezember-2022 Freitag'  # de_DE locale
datetime_object = datetime.strptime(date_str_de_DE, '%d-%B-%Y %A')
print(type(datetime_object))
print(datetime_object)


# Output:
# <class 'datetime.datetime'>
# 2022-12-16 00:00:00
```

# date

- strftime() : ?????????????

```
import datetime
d = datetime.date(2020,1,1)   # 2020-01-01
```

```
import datetime
today = datetime.date.today()
print(today)                # 2021-10-19
print(today.year)           # 2021
print(today.month)          # 10
print(today.day)            # 19
print(today.weekday())      # 1   ( 因為是星期二, 所以是 1 )
print(today.isoweekday())   # 2   ( 因為是星期二, 所以是 2 )
print(today.isocalendar())  # (2021, 42, 2) ( 第三個數字是星期二, 所以是 2 )
print(today.isoformat())    # 2021-10-19
print(today.ctime())        # Tue Oct 19 00:00:00 2021
print(today.strftime('%Y.%m.%d'))   # 2021.10.19

newDay = today.replace(year=2020)
print(newDay)               # 2020-10-19
```

```
import datetime
d1 = datetime.date(2020, 6, 24)
d2 = datetime.date(2021, 11, 24)
print(abs(d1-d2).days)      # 518
```

# time

```
import datetime
thisTime = datetime.time(12,0,0,1)
print(thisTime)   # 12:00:00.000001
```

```
import datetime
thisTime = datetime.time(14,0,0,1,tzinfo=datetime.timezone(datetime.timedelta(hours=8)))
print(thisTime)             # 14:00:00.000001+08:00
print(thisTime.isoformat())   # 14:00:00.000001+08:00
print(thisTime.tzname())      # UTC+08:00
print( thisTime.strftime('%H:%M:%S'))   # 14:00:00
```

```
newTime = today.replace(hour=20)
print(newTime)              # 20:00:00.000001+08:00
```

# datetime

- `datetime.datetime`?????????????
- `datetime.date`???
- `datetime.time`???
- `datetime.timedelta`????????

```
import datetime
thisTime = datetime.datetime(2020,1,1,20,20,20,20)
print(thisTime)    # 2020-01-01 20:20:20.000020
```

```
import datetime
print(datetime.datetime.today())    # 2021-10-19 06:15:46.022925
print(datetime.datetime.now(tz=datetime.timezone(datetime.timedelta(hours=8))))
# 2021-10-19 14:15:46.027982+08:00
print(datetime.datetime.utcnow())   # 2021-10-19 06:15:46.028630
```

```
import datetime
now = datetime.datetime.now(tz=datetime.timezone(datetime.timedelta(hours=8)))
print(now)              # 2021-10-19 14:25:46.962975+08:00
print(now.date())       # 2021-10-19
print(now.time())       # 14:25:46.962975
print(now.tzname())     # UTC+08:00
print(now.weekday())    # 1
print(now.isoweekday())   # 2
print(now.isocalendar())  # (2021, 42, 2)
print(now.isoformat())    # 2021-10-19 14:25:46.962975+08:00
print(now.ctime())       # Tue Oct 19 14:48:38 2021
print(now.strftime('%Y/%m/%d %H:%M:%S'))  # 2021/10/19 14:48:38
print(now.timetuple())    # time.struct_time(tm_year=2021, tm_mon=10, tm_mday=19, tm_hour=16,
tm_min=8, tm_sec=6, tm_wday=1, tm_yday=292, tm_isdst=-1)
```

# timedelta

??/????

```
import datetime
today = datetime.datetime.now()
yesterday = today - datetime.timedelta(days=1)
tomorrow = today + datetime.timedelta(days=1)
nextweek = today + datetime.timedelta(weeks=1)
print(today)       # 2021-10-19 07:01:22.669886
print(yesterday)   # 2021-10-18 07:01:22.669886
print(tomorrow)    # 2021-10-20 07:01:22.669886
print(nextweek)    # 2021-10-26 07:01:22.669886
```

# Timezone

```
import datetime
tzone = datetime.timezone(datetime.timedelta(hours=8))
now = datetime.datetime.now(tz=tzone)
print(now)    # 2021-10-19 15:07:51.128092+08:00
```

```
from datetime import datetime, timezone

# Get the current time in UTC
utc_time = datetime.now(timezone.utc)

print(utc_time)
```

```
from datetime import datetime
import pytz

timezone = pytz.timezone("America/New_York")

current_time_in_timezone = datetime.now(timezone)

print(current_time_in_timezone)
```

# Sleep

```
import time

time.sleep(5) # Pauses the code for 5 seconds
```

# Timestamp

### Get Current Time in Milliseconds

```
milliseconds_since_epoch = time.time() * 1000
```

### Get Current Timestamp

```
current_timestamp = time.time()

print(current_timestamp)
```

### Timestamp to a human-readable date

```
timestamp = time.time()

readable_date = datetime.fromtimestamp(timestamp)

print(readable_date)
```

## Time Diff.

```
time1 = datetime.now()

# ... some operations ...

time2 = datetime.now()

difference = time2 - time1

print(difference)
```

```
start_time = time.time()

# ... some operations ...

end_time = time.time()

elapsed_time = end_time - start_time

print(f"Time elapsed: {elapsed_time} seconds")
```

# ????????

```
import datetime

def dow(date):
    dateobj = datetime.datetime.strptime(date, r"%Y-%m-%d")
    return dateobj.strftime("%A")

date_str = "2024-12-11"
print(dow(date_str))  # Output: Wednesday
```

???????

```
import datetime
from datetime import date

def add_year(date_obj):
  try:
    new_date_obj = date_obj.replace(year = date_obj.year + 1)
  except ValueError:
    # This gets executed when the above method fails,
    # which means that we're making a Leap Year calculation
    new_date_obj = date_obj.replace(year = date_obj.year + 4)
  return new_date_obj

def next_date(date_string):
  # Convert the argument from string to date object
  date_obj = datetime.datetime.strptime(date_string, r"%Y-%m-%d")
  next_date_obj = add_year(date_obj)
  #print("DEBUG", next_date_obj)

  # Convert the datetime object to string,
  # in the format of "yyyy-mm-dd"
  next_date_string = next_date_obj.strftime("%Y-%m-%d")
  return next_date_string

today = date.today()  # Get today's date
#print("DEBUG Today: ", today)
print(next_date(str(today)))
# Should return a year from today, unless today is Leap Day
```

```
print(next_date("2021-01-01")) # Should return 2022-01-01
print(next_date("2020-02-29")) # Should return 2024-02-29
```

## Resources

- ? Python datetime (strftime, strptime) ???????????????? | From-Locals

---

Revision #19
Created 7 June 2023 10:42:11 by Admin
Updated 11 December 2024 14:51:54 by Admin