

### 1 Ví dụ 1

```
# Cho 1 dãy số
# In ra các phần tử phân biệt
lst = [1, 2, 3, 1, 4, 2, 1, 3]
print(set(lst))
```

Kết quả: {1, 2, 3, 4}

### 2 Ví dụ 2

```
# Cho 1 văn bản
# In ra các từ đơn phân biệt trong văn bản
```

```
st = 'nhiệt độ hiện tại là 15 độ C, đêm nay nhiệt độ sẽ giảm xu  
ống 9-10 độ C'  
words = st.split()  
print(set(words))
```

### 3 Ví dụ 3

```
# Cho danh sách các thời điểm khách hàng đến cửa hàng
# In ra các khung giờ có khách hàng
timestamps = ['06:45', '06:50', '07:05', '07:15', '08:05',  
              '11:45', '11:50', '11:55', '12:15', '12:30', '13:05',  
              '18:45', '18:50', '19:15', '19:30', '19:45'  
              ]

# TODO : 06-07, 07-08, 08-09, 11-12, 12-13, 13-14, 18-19, 19-20
hours = set()
for timestamp in timestamps:
    hour, minute = timestamp.split(':')
    hours.add(hour)

for hour in sorted(hours):
    print(f'{hour}-{int(hour)+1:02d}')
```

#### 4 Ví dụ 4

```
# Nhập vào phát âm tiếng Việt của 1 số từ 1-10
# In ra số đó
# VD: năm -> 5
```

```
phat_am_so = input('Phát âm của số:')
table = {
    'một': 1,
    'hai': 2,
    'ba': 3,
    'bốn': 4,
    'năm': 5,
    'sáu': 6,
    'bảy': 7,
    'tám': 8,
    'chín': 9,
    'mười': 10
}
print(table.get(phat_am_so.lower(), 'không tồn tại'))
```

#### 5 Ví dụ 5

```
# cho 1 dãy số
# In ra số lần xuất hiện của các số phân biệt
lst = [1, 2, 3, 1, 2, 5, 6, 7, 2, 3, 10000]
counts = {}
for x in lst:
    counts[x] = counts.get(x, 0) + 1

for x in counts:
    print(x, ': ', counts[x])
```

#### 6 Ví dụ 6

```
studentList = [
```

```

        {'studentNumber': '10001', 'name': 'Nguyen Van A', 'email':
'nva@nhatnghe.com'},
        {'studentNumber': '10002', 'name': 'Nguyen Van B', 'email':
'nvb@nhatnghe.com'},
        {'studentNumber': '10003', 'name': 'Nguyen Van C', 'email':
'nvc@nhatnghe.com'},
    ]

```

```

for st in studentList:
    print(st['name'])

```

# Yêu cầu: Nhập vào mã học sinh --  
 > in ra tên & email của học sinh đó

```
code = input('Mã học sinh:')
```

```
table = {}
```

```

for st in studentList:
    key = st['studentNumber']
    value = st
    table[key] = value

```

```
student = table.get(code)
```

```

if student != None:
    print(student['name'], student['email'])
else:
    print('Mã ko tồn tại')

```

## 7 Ví dụ

# Nhập vào mã --> in ra tên & giá

```

productList = [
    {'code': 'IPX', 'name': 'IPhone X', 'price': 12.5},
    {'code': 'IP11', 'name': 'IPhone 11', 'price': 17.5},
    {'code': 'IP12', 'name': 'IPhone 12', 'price': 25},
]

```

```
code = input('Mã sản phẩm:')
```

```
table = {}
```

```

for p in productList : table[p['code']] = p

product = table.get(code)
if product:
    print(product['name'], product['price'])
else:
    print('Không tồn tại')

```

## 8 Thống kê doanh số

# Cho danh sách đơn hàng của một cửa hàng  
 # In ra doanh số của mỗi loại sản phẩm

```

orderList = [
    {'productCode': 'IPX', 'qty': 1},
    {'productCode': 'IP11', 'qty': 2},
    {'productCode': 'IPX', 'qty': 1},
    {'productCode': 'IP11', 'qty': 1},
    {'productCode': 'IP12', 'qty': 2},
    {'productCode': 'IP11', 'qty': 1},
]

```

```

productList = [
    {'code': 'IPX', 'price': 11.5},
    {'code': 'IP11', 'price': 17.5},
    {'code': 'IP12', 'price': 25},
]

```

```

price_table = {}
for p in productList:
    price_table[p['code']] = p['price']

```

```

revenues = {}

```

```

for order in orderList:
    code = order['productCode']
    qty = order['qty']
    price = price_table[code]

```

```
revenues[code] = revenues.get(code, 0) + price*qty
```

```
print(revenues)
```

## 9 Tính điểm chuẩn đầu vào

Tính điểm chuẩn thi vào 1 trường ĐH

```
marks = [  
    20, 21, 20, 21, 22, 23, 20,  
    17, 16, 15, 20, 21, 23, 21,  
    20, 21, 24, 27, 20, 28, 26,  
    25, 24, 23, 23, 20, 21, 24,  
    22, 14, 13, 12, 14, 15, 16
```

```
]
```

```
qty = 10
```

```
counts = {}
```

```
for mark in marks:
```

```
    counts[mark] = counts.get(mark, 0) + 1
```

```
print(counts)
```

```
count_items = counts.items()
```

```
count_items = sorted(count_items, reverse=True)
```

```
print(count_items)
```

# TODO: Di tu cao xuống thấp, lấy dư chỉ tiêu thì dừng lại

```
acc = 0
```

```
thresh = 0
```

```
for mark, count in count_items:
```

```
    if acc + count > qty:
```

```
        break
```

```
    acc += count
```

```
    thresh = mark
```

```
print(thresh, acc)
```

# Bài tập về Kiểu ngày tháng

## 10 VD1

# In ra ngày tháng năm của ngày trong tuần tới có cùng thứ với ngày hôm nay

```
from datetime import date, timedelta
d = date.today()
d += timedelta(days=7)
print(d.strftime('%d/%m/%Y'))
```

## 11 VD2

# Một người mua mặt hàng vào hôm nay  
# Nhập vào 1 số N : số ngày vận chuyển  
# In ra ngày dự kiến nhận hàng

```
from datetime import date, timedelta

N = int(input('Số ngày vận chuyển :'))
d = date.today()
d += timedelta(days=N)
print(d.strftime('%d/%m/%Y'))
```

## 12 VD3

# Một người mua mặt hàng vào hôm nay  
# Nhập vào 1 số N : số ngày vận chuyển  
# T7, CN: nghỉ làm việc  
# In ra ngày dự kiến nhận hàng

```
# Hint : Dùng hàm d.weekday()
from datetime import date, timedelta
N = int(input('Số ngày vận chuyển :'))
d = date.today()

while N > 0:
    d += timedelta(days=1)
    if d.weekday() < 5:
        N -= 1
```

```
print(d)
```

### 13 VD

```
# Nhập vào 1 năm (2000-2099)
```

```
# In ra ngày chủ nhật cuối cùng của năm
```

```
from datetime import date, timedelta
```

```
y = int(input('Năm:'))
```

```
d = date(y, 12, 31) # 31/12/y
```

```
while d.weekday() != 6:
```

```
    d -= timedelta(days=1)
```

```
print(d)
```

### 14 Xe Bus

```
# Tuyến xe bus bắt đầu chạy 05:30
```

```
# Tần suất : 15 phút/chuyến
```

```
# In ra thời gian chuyển bus tiếp theo
```

```
from datetime import datetime, timedelta
```

```
t = datetime.now()
```

```
print(t)
```

```
#TODO: Tao duoc t0 = 05:30 today
```

```
t0 = datetime(t.year, t.month, t.day, 5, 30, 0)
```

```
while t0 < t:
```

```
    t0 += timedelta(minutes=15)
```

```
print(t0.strftime('%H:%M %d/%m/%Y'))
```

```
import calendar
```

```
# create a plain text calendar
```

```
c = calendar.TextCalendar(calendar.SUNDAY)
```

```
str = c.formatmonth(2021, 3, 0, 0)
```

```

print(str)

# create an HTML formatted calendar
hc = calendar.HTMLCalendar(calendar.SUNDAY)
str = hc.formatmonth(2021, 3)
print(str)

# loop over the days of a month
# zeroes mean that the day of the week is in an overlapping month
for i in c.itermonthdays(2021, 8):
    print(i)

# The Calendar module provides useful utilities for the given locale,
# such as the names of days and months in both full and abbreviated forms
for name in calendar.month_name:
    print(name)

for day in calendar.day_name:
    print(day)

# Calculate days based on a rule: For example, consider
# a team meeting on the first Friday of every month.
# To figure out what days that would be for each month,
# we can use this script:
print("Team meetings will be on:")
for m in range(1, 13):
    # returns an array of weeks that represent the month
    cal = calendar.monthcalendar(2017, m)
    # The first Friday has to be within the first two weeks
    weekone = cal[0]
    weektwo = cal[1]

    if weekone[calendar.FRIDAY] != 0:

```



```

        meetday = weekone[calendar.FRIDAY]
    else:
        # if the first friday isn't in the first week, it must
        be in the second
        meetday = weektwo[calendar.FRIDAY]

    print("%10s %2d" % (calendar.month_name[m], meetday))

```

## 15 Ví dụ tổng hợp

```

import calendar

# create a plain text calendar
c = calendar.TextCalendar(calendar.SUNDAY)
str = c.formatmonth(2021, 3, 0, 0)
print(str)

# create an HTML formatted calendar
hc = calendar.HTMLCalendar(calendar.SUNDAY)
str = hc.formatmonth(2021, 3)
print(str)

# loop over the days of a month
# zeroes mean that the day of the week is in an overlapping month
for i in c.itermonthdays(2021, 8):
    print(i)

# The Calendar module provides useful utilities for the given locale,
# such as the names of days and months in both full and abbreviated forms
for name in calendar.month_name:
    print(name)

for day in calendar.day_name:
    print(day)

```

```

# Calculate days based on a rule: For example, consider
# a team meeting on the first Friday of every month.
# To figure out what days that would be for each month,
# we can use this script:
print("Team meetings will be on:")
for m in range(1, 13):
    # returns an array of weeks that represent the month
    cal = calendar.monthcalendar(2017, m)
    # The first Friday has to be within the first two weeks
    weekone = cal[0]
    weektwo = cal[1]

    if weekone[calendar.FRIDAY] != 0:
        meetday = weekone[calendar.FRIDAY]
    else:
        # if the first friday isn't in the first week, it must
        # be in the second
        meetday = weektwo[calendar.FRIDAY]

    print("%10s %2d" % (calendar.month_name[m], meetday))

```

## Hàm

```

# function with default value for an argument
def power(num, x=1):
    result = 1;
    for i in range(x):
        result = result * num
    return result

#function with variable number of arguments
def multi_add(*args):
    result = 0;
    for x in args:

```

```
    result = result + x  
    return result
```

```
print (power(2))  
print (power(2,3))  
print (power(x=3, num=2))  
print (multi_add(4,5,10,4))
```