

売上データ

	佐藤		2,000
	山田		1,500
	高橋		3,000
	佐藤		2,500



辞書型

値

キー

佐藤

	佐藤		2,000
	佐藤		2,500

山田

	山田		1,500
	山田		3,500

高橋

	高橋		3,000
	高橋		4,500

リスト型

```
li=[value1, value2, value3]  
print(li[1])
```



value2

数字を指定して値を取り出す

辞書型

```
li=["key1":value1, "key2":value2, "key3":value3]  
print(li["key2"])
```



value2

キーを指定して値を取り出す

```
# ダミーデータ
```

```
dummy_data = [  
    ["佐藤", 400],  
    ["佐藤", 900],  
    ["佐藤", 300],  
    ["山田", 500],  
    ["山田", 200]  
]
```

```
# データを顧客名で辞書型に分割 --- (*1)
```

```
users = {}
```

```
for row in dummy_data:
```

```
    name, value = row # 変数に分ける
```

```
    # 顧客名が初出ならリストを初期化 --- (*2)
```

```
    if name not in users:
```

```
        users[name] = []
```

```
    # 顧客名をキーに値にデータを追記 --- (*3)
```

```
    users[name].append(row)
```

```
# 顧客ごとに集計 --- (*4)
```

```
for name, rows in users.items():
```

```
    print(rows) # --- 集計対象データを表示
```

```
    # 顧客の購入金額を合計 --- (*5)
```

```
    total = 0
```

```
    for row in rows:
```

```
        total += row[1]
```

```
    print(name, total) # 結果を表示
```

```
["佐藤", 400],  
 ["佐藤", 900],  
 ["佐藤", 300],  
 ["山田", 500],  
 ["山田", 200]
```

name, value = row

user(name)にappendでデータを追記

実行結果

```
[['佐藤', 400], ['佐藤', 900], ['佐藤', 300]]  
佐藤 1600  
[['山田', 500], ['山田', 200]]  
山田 700
```

	A	B	C	D	E	F
1	2020-11-06 0:00:00	E商店	醤油	20	1500	30000
2	2020-11-11 0:00:00	F商店	ソース	2	1500	3000
3	2020-11-20 0:00:00	E商店	味噌	8	950	7600
4	2020-11-21 0:00:00	F商店	醤油	10	1500	15000
5	2022-11-01 0:00:00	A商店	ミカン	2	2500	5000
6	2022-11-02 0:00:00	B商店	リンゴ	1	1500	1500
7	2022-11-03 0:00:00	C商店	ミカン	3	2500	7500
8	2022-11-15 0:00:00	D商店	ミカン	4	2500	10000
9	2022-11-16 0:00:00	A商店	リンゴ	2	1500	3000
10	2022-11-17 0:00:00	D商店	ブドウ	2	3000	6000
11	2022-11-22 0:00:00	A商店	ブドウ	2	3000	6000
12	2020-11-03 0:00:00	E商店	豆腐	4	1000	4000
13	2020-11-05 0:00:00	F商店	豆腐	1	1000	1000
14	2020-11-05 0:00:00	H商店	湯葉	1	2000	2000
15	2020-11-07 0:00:00	E商店	大豆セット	2	2000	4000
16	2020-11-15 0:00:00	F商店	小豆	1	1000	1000
17	2020-11-16 0:00:00	H商店	豆腐	4	1000	4000

↓ 商店毎に配列

```
[datetime.datetime(2020, 11, 6, 0, 0), 'E商店', '醤油', 20, 1500, 30000]
[datetime.datetime(2020, 11, 20, 0, 0), 'E商店', '味噌', 8, 950, 7600]
[datetime.datetime(2020, 11, 3, 0, 0), 'E商店', '豆腐', 4, 1000, 4000]
[datetime.datetime(2020, 11, 7, 0, 0), 'E商店', '大豆セット', 2, 2000, 4000]
[datetime.datetime(2020, 11, 11, 0, 0), 'F商店', 'ソース', 2, 1500, 3000]
[datetime.datetime(2020, 11, 21, 0, 0), 'F商店', '醤油', 10, 1500, 15000]
[datetime.datetime(2020, 11, 5, 0, 0), 'F商店', '豆腐', 1, 1000, 1000]
[datetime.datetime(2020, 11, 15, 0, 0), 'F商店', '小豆', 1, 1000, 1000]
[datetime.datetime(2020, 11, 20, 0, 0), 'F商店', '小豆', 5, 1000, 5000]
```

売上合計

```
E商店 45600
F商店 40000
A商店 14000
B商店 10500
C商店 19100
D商店 34000
H商店 6000
G商店 33000
I商店 5700
```

```
import openpyxl as excel, json
# 入出力ファイルを指定
in_file = 'C:/Users/
out_file = 'C:/Users/

# メイン処理 --- (*1)
def split_list():
    # Excelシートのデータを顧客ごとに分ける --- (*2)
    users = read_and_split(in_file)
    # お客ごとにデータを集計する --- (*3)
    result = {}
    for name, rows in users.items():
        result[name] = calc_user(rows)
        print(name, result[name]['total'])
    # ファイルに結果を保存 --- (*4)
    with open(out_file, "wt") as fp:
        json.dump(result, fp)
```

```
# 入力ファイルを読んで顧客ごとに分割 --- (*5)
def read_and_split(in_file):
    users = {} # 辞書型の変数を初期化
    # ブックを開いてシートの全行を読む --- (*6)
    sheet = excel.load_workbook(in_file).active
    for row in sheet.iter_rows():
        # シートの一行を取り出してリストに変換 --- (*7)
        values = [col.value for col in row]
        name = values[1] # 顧客名を取り出す
        # 初出ならリスト初期化
        if name not in users: users[name] = []
        # データを顧客毎に分ける --- (*8)
        users[name].append(values)
    return users
```

strftimeメソッドで
日付を月／日に変換

←JSON形式でファイル保管

```
# 顧客一人分のデータを集計する --- (*9)
def calc_user(rows):
    total = 0
    items = [] # 請求の明細用
    # 集計処理を行う --- (*10)
    for row in rows:
        # 請求書に必要な項目だけ抽出して請求書明細の形式で追加
        date, _, item, cnt, price, _ = row
        date_s = date.strftime('%m/%d') # --- (*11)
        items.append([date_s, item, cnt, price])
        # 合計金額を計算 --- (*12)
        total += cnt * price
    # 集計結果を辞書形式で返す
    return {'items': items, 'total': total}

if __name__ == "__main__":
    split_list() # メイン処理を実行
```

```
[datetime.datetime(2020, 11, 6, 0, 0), 'E商店', '醤油', 20, 1500, 30000]
[datetime.datetime(2020, 11, 20, 0, 0), 'E商店', '味噌', 8, 950, 7600]
[datetime.datetime(2020, 11, 3, 0, 0), 'E商店', '豆腐', 4, 1000, 4000]
[datetime.datetime(2020, 11, 7, 0, 0), 'E商店', '大豆セット', 2, 2000, 4000]
[datetime.datetime(2020, 11, 11, 0, 0), 'F商店', 'ソース', 2, 1500, 3000]
[datetime.datetime(2020, 11, 21, 0, 0), 'F商店', '醤油', 10, 1500, 15000]
[datetime.datetime(2020, 11, 5, 0, 0), 'F商店', '豆腐', 1, 1000, 1000]
[datetime.datetime(2020, 11, 15, 0, 0), 'F商店', '小豆', 1, 1000, 1000]
[datetime.datetime(2020, 11, 20, 0, 0), 'F商店', '小豆', 5, 1000, 5000]
```

ユーザー毎の配列になっている

実行結果

E商店 45600
F商店 40000
A商店 14000
B商店 10500
C商店 19100
D商店 34000
H商店 6000
G商店 33000
I商店 5700