

Bài 7: Tuples và Sets - So sánh các cấu trúc dữ liệu

W27 - Thứ 5 - Buổi sáng (09:00-12:00)

Mục tiêu:

- Hiểu về Tuples và cách sử dụng
- Hiểu về Sets và các thao tác cơ bản
- So sánh và lựa chọn cấu trúc dữ liệu phù hợp
- Áp dụng vào các bài tập thực tế

1. TUPLES - Bộ dữ liệu không thay đổi

1.1 Tuple là gì?

- Tuple là một collection có thứ tự và **không thể thay đổi** (immutable)
- Được định nghĩa bằng dấu ngoặc tròn () hoặc không cần ngoặc
- Cho phép trùng lặp phần tử
- Thường dùng để lưu trữ dữ liệu không thay đổi như tọa độ, ngày tháng

```
In [ ]: # Tạo tuple cơ bản
toa_do = (10, 20)
mau_sac = ("đỏ", "xanh", "vàng")
so_hon_hop = (1, "hello", 3.14, True)

print("Tọa độ:", toa_do)
print("Màu sắc:", mau_sac)
print("Số hỗn hợp:", so_hon_hop)
print("Kiểu dữ liệu:", type(toa_do))
```

1.2 Truy cập phần tử trong Tuple

```
In [ ]: # Truy cập bằng index
thong_tin_sv = ("Nguyễn Văn A", 20, "CNTT", 8.5)

print("Tên:", thong_tin_sv[0])
print("Tuổi:", thong_tin_sv[1])
print("Ngành:", thong_tin_sv[2])
print("Điểm TB:", thong_tin_sv[3])

# Slicing
print("Thông tin cơ bản:", thong_tin_sv[:2])
print("Thông tin học tập:", thong_tin_sv[2:])
```

1.3 Phương thức và thao tác với Tuple

```
In [ ]: # Các phương thức cơ bản
diem_so = (8, 7, 9, 8, 6, 8, 10)

print("Số lượng phần tử:", len(diem_so))
print("Đếm số điểm 8:", diem_so.count(8))
print("Vị trí điểm 9:", diem_so.index(9))
print("Điểm cao nhất:", max(diem_so))
print("Điểm thấp nhất:", min(diem_so))
print("Tổng điểm:", sum(diem_so))
```

1.4 Unpacking Tuple

```
In [1]: # Unpacking - gán từng phần tử vào biến riêng
hoc_sinh = ("Trần Thị B", 18, "Lý")

# Cách 1: Unpacking đầy đủ
ten, tuoi, mon_hoc = hoc_sinh
print(f"Tên: {ten}, Tuổi: {tuoi}, Môn: {mon_hoc}")

# Cách 2: Unpacking với *
diem_thi = (8, 7, 9, 6, 10)
diem_dau, *diem_giua, diem_cuoi = diem_thi
print(f"Điểm đầu: {diem_dau}")
print(f"Điểm giữa: {diem_giua}")
print(f"Điểm cuối: {diem_cuoi}")
```

Tên: Trần Thị B, Tuổi: 18, Môn: Lý

Điểm đầu: 8

Điểm giữa: [7, 9]

Điểm cuối: 10



TODO 1: Thực hành Tuple

Tạo một chương trình quản lý thông tin sản phẩm bằng tuple:

```
In [ ]: # TODO 1: Hoàn thành code dưới đây
# Tạo tuple chứa thông tin sản phẩm (tên, giá, số Lượng, danh mục)
san_pham_1 = ("Laptop Dell", 15000000, 5, "Điện tử")
san_pham_2 = ("Bàn phím", 500000, 20, "Phụ kiện")
san_pham_3 = ("Chuột", 200000, 15, "Phụ kiện")

# TODO: Unpacking thông tin sản phẩm 1
# ten_sp, gia_sp, so_luong_sp, danh_muc_sp = ?

# TODO: In ra thông tin sản phẩm theo định dạng đẹp
# print(f"Sản phẩm: {ten_sp}")
# print(f"Giá: {gia_sp:,} VNĐ")
# print(f"Còn Lại: {so_luong_sp} cái")
# print(f"Danh mục: {danh_muc_sp}")
```

```
# TODO: Tính tổng giá trị kho hàng của sản phẩm 1
# tong_gia_tri = ?
# print(f"Tổng giá trị: {tong_gia_tri:}, VNĐ")
```

2. SETS - Tập hợp không trùng lặp

2.1 Set là gì?

- Set là một collection **không có thứ tự** và **không trùng lặp**
- Được định nghĩa bằng dấu ngoặc nhọn {} hoặc hàm set()
- Hỗ trợ các phép toán tập hợp: hợp, giao, hiệu
- Thường dùng để loại bỏ phần tử trùng lặp và kiểm tra thành viên

```
In [ ]: # Tạo set cơ bản
mau_sac = {"đỏ", "xanh", "vàng", "đỏ"} # "đỏ" sẽ chỉ xuất hiện 1 lần
so_chan = {2, 4, 6, 8, 10}
set_rong = set()

print("Màu sắc:", mau_sac)
print("Số chẵn:", so_chan)
print("Set rỗng:", set_rong)
print("Kiểu dữ liệu:", type(mau_sac))

# Tạo set từ List (Loại bỏ trùng lặp)
diem_so = [8, 7, 9, 8, 6, 8, 10, 7]
diem_khong_trung = set(diem_so)
print("Điểm gốc:", diem_so)
print("Điểm không trùng:", diem_khong_trung)
```

2.2 Thao tác cơ bản với Set

```
In [1]: # Thêm và xóa phần tử
hoa_qua = {"táo", "cam", "xoài"}
print("Ban đầu:", hoa_qua)

# Thêm phần tử
hoa_qua.add("dưa hấu")
print("Sau khi thêm dưa hấu:", hoa_qua)

# Thêm nhiều phần tử
hoa_qua.update(["nho", "lê", "táo"]) # "táo" đã có nên không thêm
print("Sau khi thêm nhiều:", hoa_qua)

# Xóa phần tử
hoa_qua.remove("cam") # Lỗi nếu không tồn tại
hoa_qua.discard("cherry") # Không lỗi nếu không tồn tại
print("Sau khi xóa:", hoa_qua)

# Kiểm tra thành viên
print("Có táo không?", "táo" in hoa_qua)
print("Có cam không?", "cam" in hoa_qua)
```

```

Ban đầu: {'xoài', 'táo', 'cam'}
Sau khi thêm dừa hấu: {'xoài', 'táo', 'dừa hấu', 'cam'}
Sau khi thêm nhiều: {'cam', 'dừa hấu', 'lê', 'nho', 'xoài', 'táo'}
Sau khi xóa: {'dừa hấu', 'lê', 'nho', 'xoài', 'táo'}
Có táo không? True
Có cam không? False

```

```
Out[1]: 5
```

2.3 Phép toán tập hợp

```

In [4]: # Định nghĩa các tập hợp
lop_10a = {"An", "Bình", "Chi", "Dung", "Em"}
lop_10b = {"Chi", "Dung", "Phong", "Giang", "Hoa"}
clb_bong_da = {"An", "Chi", "Phong", "Khoa"}

print("Lớp 10A:", lop_10a)
print("Lớp 10B:", lop_10b)
print("CLB bóng đá:", clb_bong_da)
print()

# Phép HỢP (Union) - Tất cả học sinh
tat_ca_hs = lop_10a | lop_10b # hoặc lop_10a.union(lop_10b)
print("Tất cả học sinh:", tat_ca_hs)

# Phép GIAO (Intersection) - Học sinh học cả 2 lớp
hs_chung = lop_10a & lop_10b # hoặc lop_10a.intersection(lop_10b)
print("Học sinh học cả 2 lớp:", hs_chung)

# Phép HIỆU (Difference) - Học sinh chỉ ở Lớp 10A
chi_10a = lop_10a - lop_10b # hoặc lop_10a.difference(lop_10b)
print("Chỉ ở lớp 10A:", chi_10a)

# Phép HIỆU ĐỐI XỨNG - Học sinh không học chung
khong_chung = lop_10a ^ lop_10b # hoặc lop_10a.symmetric_difference(lop_10b)
print("Không học chung:", khong_chung)

```

```
Lớp 10A: {'An', 'Dung', 'Chi', 'Bình', 'Em'}
```

```
Lớp 10B: {'Phong', 'Giang', 'Hoa', 'Dung', 'Chi'}
```

```
CLB bóng đá: {'An', 'Phong', 'Chi', 'Khoa'}
```

```
Tất cả học sinh: {'An', 'Giang', 'Hoa', 'Dung', 'Chi', 'Bình', 'Em', 'Phong'}
```

```
Học sinh học cả 2 lớp: {'Chi', 'Dung'}
```

```
Chỉ ở lớp 10A: {'Bình', 'An', 'Em'}
```

```
Không học chung: {'An', 'Phong', 'Giang', 'Hoa', 'Bình', 'Em'}
```



TODO 2: Thực hành Set

Quản lý danh sách học sinh tham gia các hoạt động:

```

In [ ]: # TODO 2: Hoàn thành code dưới đây
# Danh sách học sinh tham gia các hoạt động
hoc_sinh_toan = {"An", "Bình", "Chi", "Dung", "Em", "Phong"}
hoc_sinh_ly = {"Bình", "Chi", "Giang", "Hoa", "Khoa"}
hoc_sinh_hoa = {"An", "Chi", "Em", "Giang", "Khoa", "Long"}

```

```

print("Học sinh học Toán:", hoc_sinh_toan)
print("Học sinh học Lý:", hoc_sinh_ly)
print("Học sinh học Hóa:", hoc_sinh_hoa)
print()

# TODO: Tìm học sinh học cả 3 môn
# hoc_ca_3_mon = ?
# print("Học sinh học cả 3 môn:", hoc_ca_3_mon)

# TODO: Tìm học sinh chỉ học Toán (không học Lý và Hóa)
# chi_hoc_toan = ?
# print("Chỉ học Toán:", chi_hoc_toan)

# TODO: Tìm học sinh học ít nhất 1 môn khoa học (Lý hoặc Hóa)
# hoc_khoa_hoc = ?
# print("Học ít nhất 1 môn khoa học:", hoc_khoa_hoc)

# TODO: Đếm tổng số học sinh tham gia các hoạt động
# tong_hoc_sinh = ?
# print(f"Tổng số học sinh: {len(tong_hoc_sinh)} em")

```

3. SO SÁNH CÁC CẤU TRÚC DỮ LIỆU

3.1 Bảng so sánh tổng quan

```

In [1]: # Tạo bảng so sánh
import pandas as pd

so_sanh = {
    "Đặc điểm": ["Có thứ tự", "Thay đổi được", "Trùng lặp", "Truy cập bằng index",
    "List": ["Có", "Có", "Có", "Có", "Chậm O(n)"],
    "Tuple": ["Có", "Không", "Có", "Có", "Chậm O(n)"],
    "Set": ["Không", "Có", "Không", "Không", "Nhanh O(1)"],
    "Dictionary": ["Có (3.7+)", "Có", "Key: Không, Value: Có", "Bằng key", "Nhanh O(1)"]
}

df = pd.DataFrame(so_sanh)
print(df.to_string(index=False))

```

	Đặc điểm	List	Tuple	Set	Dictionary
	Có thứ tự	Có	Có	Không	Có (3.7+)
	Thay đổi được	Có	Không	Có	Có
	Trùng lặp	Có	Có	Không	Key: Không, Value: Có
	Truy cập bằng index	Có	Có	Không	Bằng key
	Tốc độ tìm kiếm	Chậm O(n)	Chậm O(n)	Nhanh O(1)	Nhanh O(1)

3.2 Khi nào sử dụng cấu trúc nào?

```

In [ ]: # Ví dụ thực tế về việc chọn cấu trúc dữ liệu

# 1. LIST - Khi cần thứ tự và thay đổi
diem_hoc_ki = [8.5, 7.0, 9.0, 6.5] # Có thể thêm điểm mới

```

```

print("Điểm học kì (List):", diem_hoc_ki)

# 2. TUPLE - Khi cần thứ tự nhưng không thay đổi
thong_tin_ca_nhan = ("Nguyễn Văn A", "1990-01-01", "Nam") # Thông tin cố định
print("Thông tin cá nhân (Tuple):", thong_tin_ca_nhan)

# 3. SET - Khi cần Loại bỏ trùng Lặp và kiểm tra nhanh
mon_hoc_da_dang_ki = {"Toán", "Lý", "Hóa", "Toán"} # Tự động Loại bỏ "Toán" trùng
print("Môn đã đăng kí (Set):", mon_hoc_da_dang_ki)

# 4. DICTIONARY - Khi cần ánh xạ key-value
diem_mon_hoc = {"Toán": 8.5, "Lý": 7.0, "Hóa": 9.0}
print("Điểm môn học (Dict):", diem_mon_hoc)

```

3.3 Chuyển đổi giữa các cấu trúc

```

In [2]: # Dữ Liệu gốc
danh_sach_goc = [1, 2, 3, 2, 4, 1, 5]
print("Dữ liệu gốc (List):", danh_sach_goc)

# Chuyển đổi qua Lại
tuple_tu_list = tuple(danh_sach_goc)
print("Chuyển thành Tuple:", tuple_tu_list)

set_tu_list = set(danh_sach_goc) # Loại bỏ trùng Lặp
print("Chuyển thành Set:", set_tu_list)

list_tu_set = list(set_tu_list) # Danh sách không trùng Lặp
print("Set về List:", list_tu_set)

# Thống kê
print(f"Số phần tử gốc: {len(danh_sach_goc)}")
print(f"Số phần tử unique: {len(set_tu_list)}")
print(f"Số phần tử bị trùng: {len(danh_sach_goc) - len(set_tu_list)}")

```

Dữ liệu gốc (List): [1, 2, 3, 2, 4, 1, 5]

Chuyển thành Tuple: (1, 2, 3, 2, 4, 1, 5)

Chuyển thành Set: {1, 2, 3, 4, 5}

Set về List: [1, 2, 3, 4, 5]

Số phần tử gốc: 7

Số phần tử unique: 5

Số phần tử bị trùng: 2



TODO 3: Bài tập tổng hợp

Xây dựng hệ thống quản lý thư viện đơn giản:

```

In [ ]: # TODO 3: Hoàn thành hệ thống quản lý thư viện

# Dữ Liệu mẫu
# Tuple: Thông tin sách (tên, tác giả, năm xuất bản, thể Loại)
sach_1 = ("Python cho người mới bắt đầu", "Nguyễn Văn A", 2023, "Lập trình")
sach_2 = ("Khoa học dữ liệu", "Trần Thị B", 2022, "Khoa học")
sach_3 = ("Trí tuệ nhân tạo", "Lê Văn C", 2023, "Công nghệ")

```

```
sach_4 = ("Python nâng cao", "Nguyễn Văn A", 2023, "Lập trình")

# List: Danh sách tất cả sách
tat_ca_sach = [sach_1, sach_2, sach_3, sach_4]

# TODO: Tạo set chứa tất cả tác giả (không trùng)
# tat_ca_tac_gia = set()
# for sach in tat_ca_sach:
#     tat_ca_tac_gia.add(?) # Thêm tác giả vào set
# print("Tất cả tác giả:", tat_ca_tac_gia)

# TODO: Tạo set chứa tất cả thể loại
# tat_ca_the_loai = ?
# print("Tất cả thể loại:", tat_ca_the_loai)

# TODO: Tìm sách xuất bản năm 2023
# sach_2023 = []
# for sach in tat_ca_sach:
#     if sach[?] == 2023: # Kiểm tra năm xuất bản
#         sach_2023.append(sach[0]) # Thêm tên sách
# print("Sách xuất bản 2023:", sach_2023)

# TODO: Dictionary - Nhóm sách theo thể loại
# sach_theo_the_loai = {}
# for sach in tat_ca_sach:
#     the_loai = sach[3]
#     if the_loai not in sach_theo_the_loai:
#         sach_theo_the_loai[the_loai] = []
#     sach_theo_the_loai[the_loai].append(sach[0])
#
# print("\nSách theo thể loại:")
# for the_loai, danh_sach in sach_theo_the_loai.items():
#     print(f"{the_loai}: {danh_sach}")

# TODO: Thống kê
# print(f"\nThống kê:")
# print(f"Tổng số sách: {len(tat_ca_sach)}")
# print(f"Số tác giả: {len(tat_ca_tac_gia)}")
# print(f"Số thể loại: {len(tat_ca_the_loai)}")
```

4. BÀI TẬP THỰC HÀNH



TODO 4: Quản lý danh sách phát nhạc

```
In [ ]: # TODO 4: Tạo hệ thống quản lý playlist nhạc

# Dữ liệu: Tuple chứa (tên bài, ca sĩ, thể loại, thời lượng_giây)
bai_hat_1 = ("Cố chàng trai viết lên cây", "Phan Mạnh Quỳnh", "Ballad", 240)
bai_hat_2 = ("Nơi này có anh", "Sơn Tùng MTP", "Pop", 210)
bai_hat_3 = ("Em gái mưa", "Hương Tràm", "Ballad", 320)
bai_hat_4 = ("Bùa yêu", "Bích Phương", "Pop", 180)
bai_hat_5 = ("Cố chàng trai viết lên cây", "Phan Mạnh Quỳnh", "Ballad", 240) # Trùng

# TODO: Tạo playlist (List) chứa tất cả bài hát
```

```

playlist = [bai_hat_1, bai_hat_2, bai_hat_3, bai_hat_4, bai_hat_5]
print("Playlist gốc:", playlist)

# TODO: Loại bỏ bài hát trùng lặp bằng set
bai_hat_unique = set(playlist)
print(f"Số bài hát gốc: {len(playlist)}")
print(f"Số bài hát không trùng: {len(bai_hat_unique)}")

# TODO: Tạo set chứa tất cả ca sĩ
tat_ca_ca_si = set()
for bai_hat in playlist:
    tat_ca_ca_si.add(bai_hat[1]) # Thêm tên ca sĩ
print("Tất cả ca sĩ:", tat_ca_ca_si)

# TODO: Tạo set chứa các thể loại nhạc
the_loai_nhac = set()
for bai_hat in playlist:
    the_loai_nhac.add(bai_hat[2]) # Thêm thể loại
print("Thể loại nhạc:", the_loai_nhac)

# TODO: Tính tổng thời lượng playlist (giờ)
tong_thoi_luong = 0
for bai_hat in bai_hat_unique:
    tong_thoi_luong += bai_hat[3] # Cộng thời lượng

# Chuyển đổi sang phút:giây
phut = tong_thoi_luong // 60
giay = tong_thoi_luong % 60
print(f"Tổng thời lượng: {phut} phút {giay} giây")

# TODO: Tìm bài hát dài nhất
bai_dai_nhat = max(bai_hat_unique, key=lambda x: x[3])
print(f"Bài hát dài nhất: {bai_dai_nhat[0]} - {bai_dai_nhat[3]} giây")

```



TODO 5: Bài tập Bonus - Thống kê dữ liệu nâng cao

Áp dụng kiến thức Tuple và Set vào phân tích dữ liệu:

```

In [ ]: # TODO 5: Phân tích dữ liệu học sinh

# Dữ liệu: Tuple (tên, Lớp, điểm_toán, điểm_Lý, điểm_hóa)
hoc_sinh_1 = ("An", "10A", 8.5, 7.0, 9.0)
hoc_sinh_2 = ("Bình", "10B", 7.5, 8.0, 6.5)
hoc_sinh_3 = ("Chi", "10A", 9.0, 9.5, 8.0)
hoc_sinh_4 = ("Dung", "10C", 6.0, 7.0, 7.5)
hoc_sinh_5 = ("Em", "10B", 8.0, 6.5, 9.5)

danh_sach_hs = [hoc_sinh_1, hoc_sinh_2, hoc_sinh_3, hoc_sinh_4, hoc_sinh_5]

# Tìm các lớp có trong danh sách
cac_lop = set()
for hs in danh_sach_hs:
    cac_lop.add(hs[1])
print("Các lớp:", cac_lop)

```



```
# Tính điểm trung bình mỗi học sinh
print("\nĐiểm trung bình:")
for hs in danh_sach_hs:
    ten, lop, toan, ly, hoa = hs
    dtb = (toan + ly + hoa) / 3
    print(f"{ten} ({lop}): {dtb:.2f}")

# Tìm học sinh có điểm cao nhất từng môn
hs_toan_cao = max(danh_sach_hs, key=lambda x: x[2])
hs_ly_cao = max(danh_sach_hs, key=lambda x: x[3])
hs_hoa_cao = max(danh_sach_hs, key=lambda x: x[4])

print(f"\nTốt nhất môn Toán: {hs_toan_cao[0]} - {hs_toan_cao[2]} điểm")
print(f"Tốt nhất môn Lý: {hs_ly_cao[0]} - {hs_ly_cao[3]} điểm")
print(f"Tốt nhất môn Hóa: {hs_hoa_cao[0]} - {hs_hoa_cao[4]} điểm")
```

<https://create.kahoot.it/share/w27-t5/b24bf199-2623-49c9-88e0-946712604be6>

5. TÓM TẮT BÀI HỌC

Những điều đã học:

1. **Tuple**: Cấu trúc dữ liệu không thay đổi, có thứ tự, cho phép trùng lặp
2. **Set**: Cấu trúc dữ liệu không có thứ tự, không trùng lặp, hỗ trợ phép toán tập hợp
3. **So sánh cấu trúc**: List vs Tuple vs Set vs Dictionary
4. **Ứng dụng thực tế**: Quản lý dữ liệu, loại bỏ trùng lặp, phân tích thống kê

Bài tập về nhà:

1. Tạo chương trình quản lý điểm số lớp học bằng Tuple và Set
2. Xây dựng hệ thống quản lý kho hàng đơn giản
3. Phân tích dữ liệu khảo sát sở thích âm nhạc

Buổi tiếp theo:

Mixed data structures, Student database, Data organization - W27-T5 chiều