

Tuần 27 - Thứ 4: Dictionary cơ bản trong Python

Mục tiêu học tập

Sau bài học này, bạn sẽ:

- Hiểu Dictionary là gì và tại sao cần sử dụng
- Biết cách tạo và sử dụng Dictionary
- Nắm vững các thao tác cơ bản với Dictionary
- Biết khi nào nên dùng Dictionary thay vì List

Cách sử dụng giáo trình

- Đọc phần **LÝ THUYẾT** trước
- Làm theo từng bước **TODO**
- Chạy code để xem kết quả
- Thử thay đổi giá trị để hiểu rõ hơn

LÝ THUYẾT 1: Dictionary là gì?

Dictionary (Từ điển) là kiểu dữ liệu lưu trữ theo cặp **key-value** (khóa-giá trị).

Ví dụ thực tế: Thông tin về một học sinh

- Tên: "Nguyễn Văn An"
- Tuổi: 20
- Điểm: 8.5

So sánh:

- **List:** ["Nguyễn Văn An", 20, 8.5] ❌ Khó nhớ vị trí
- **Dictionary:** {"ten": "Nguyễn Văn An", "tuoi": 20, "diem": 8.5} ✅ Rõ ràng, dễ hiểu

In []: *# TODO 1.1: Chạy code này để thấy sự khác biệt giữa List và Dictionary*

```
# Cách cũ: Sử dụng List
hoc_sinh_list = ["Nguyễn Văn An", 20, 8.5]
print("=== SỬ DỤNG LIST ===")
print("Tên học sinh:", hoc_sinh_list[0]) # Phải nhớ vị trí 0 là tên
print("Tuổi:", hoc_sinh_list[1])         # Vị trí 1 là tuổi
```

```

print("Điểm:", hoc_sinh_list[2])          # Vị trí 2 là điểm

print("\n=== SỬ DỤNG DICTIONARY ===")
# Cách mới: Sử dụng Dictionary
hoc_sinh_dict = {
    "ten": "Nguyễn Văn An",
    "tuoi": 20,
    "diem": 8.5
}
print("Tên học sinh:", hoc_sinh_dict["ten"]) # Rõ ràng hơn
print("Tuổi:", hoc_sinh_dict["tuoi"])
print("Điểm:", hoc_sinh_dict["diem"])

print("\n🤖 Bạn thấy cách nào dễ hiểu hơn?")

```

```

In [ ]: # TODO 1.2: Tự tạo thông tin về bản thân bằng Dictionary
# Thay đổi các giá trị bên dưới thành thông tin của bạn

thong_tin_toi = {
    "ten": "Tên của bạn",          # TODO: Thay bằng tên thật của bạn
    "tuoi": 0,                     # TODO: Thay bằng tuổi của bạn
    "so_thich": "Chưa biết",       # TODO: Thay bằng sở thích của bạn
    "mon_hoc_yeu_thich": "Chưa có" # TODO: Thay bằng môn học bạn thích
}

# In ra thông tin
print("Thông tin về tôi:")
print(f"Tên: {thong_tin_toi['ten']}")
print(f"Tuổi: {thong_tin_toi['tuoi']}")
print(f"Sở thích: {thong_tin_toi['so_thich']}")
print(f"Môn học yêu thích: {thong_tin_toi['mon_hoc_yeu_thich']}")

```

LÝ THUYẾT 2: Các cách tạo Dictionary

Cú pháp cơ bản: `ten_dict = {"key1": "value1", "key2": "value2"}`

3 cách tạo Dictionary:

1. **Dictionary trống:** `{}`
2. **Dictionary có dữ liệu sẵn:** `{"key": "value"}`
3. **Dùng hàm dict():** `dict(key=value)`

```

In [ ]: # TODO 2.1: Chạy code để xem 3 cách tạo Dictionary

print("=== CÁCH 1: TẠO DICTIONARY TRỐNG ===")
# Tạo Dictionary trống, sau đó thêm dữ liệu
san_pham = {}
print("Ban đầu:", san_pham)

# Thêm dữ liệu vào Dictionary trống
san_pham["ten"] = "Điện thoại iPhone"
san_pham["gia"] = 20000000
san_pham["mau"] = "Xanh"

```

```

print("Sau khi thêm:", san_pham)

print("\n=== CÁCH 2: TẠO VỚI DỮ LIỆU SẴN ===")
# Tạo Dictionary với dữ liệu ngay từ đầu
mon_an = {
    "ten": "Phở bò",
    "gia": 50000,
    "vi_tri": "Quầy số 5",
    "co_san": True
}
print("Món ăn:", mon_an)

print("\n=== CÁCH 3: DÙNG HÀM dict() ===")
# Cách viết ngắn gọn hơn (không cần dấu ngoặc kép cho key)
diem_thi = dict(toan=9, van=8, anh=7, ly=8.5)
print("Điểm thi:", diem_thi)

```

```

In [ ]: # TODO 2.2: Tự tạo Dictionary theo yêu cầu

# YÊU CẦU 1: Tạo Dictionary trống về xe hơi, sau đó thêm thông tin
xe_hoi = {} # TODO: Tạo Dictionary trống

# TODO: Thêm các thông tin sau vào Dictionary xe_hoi:
# - hang: "Toyota"
# - mau: "Đỏ"
# - nam_san_xuat: 2023
# - gia: 8000000000

# Viết code thêm thông tin ở đây:

print("Thông tin xe hơi:", xe_hoi)

# YÊU CẦU 2: Tạo Dictionary về nhà hàng với dữ liệu sẵn
# TODO: Tạo Dictionary 'nha_hang' với các thông tin:
# - ten: "Nhà hàng ABC"
# - dia_chi: "123 Đường XYZ"
# - diem_danh_gia: 4.5
# - so_ban: 20

nha_hang = {
    # TODO: Điền thông tin vào đây
}

print("\nThông tin nhà hàng:", nha_hang)

```

LÝ THUYẾT 3: Truy cập dữ liệu trong Dictionary

2 cách truy cập dữ liệu:

1. Dùng `dict["key"]` : Nhanh nhưng báo lỗi nếu key không tồn tại
2. Dùng `dict.get("key")` : An toàn hơn, có thể đặt giá trị mặc định

Khi nào dùng cách nào:

- Dùng `[]` khi **chắc chắn** key tồn tại
- Dùng `get()` khi **không chắc** key có hay không

```
In [4]: # TODO 3.1: Thực hành truy cập dữ liệu

# Tạo Dictionary mẫu về cửa hàng
cua_hang = {
    "ten": "Siêu thị Mini",
    "dia_chi": "456 Phố ABC",
    "so_nhan_vien": 15,
    "gio_mo_cua": "7:00",
    "gio_dong_cua": "22:00"
}

print("=== CÁCH 1: DÙNG [] ===")
# Truy cập khi CHẮC CHẮN key tồn tại
print("Tên cửa hàng:", cua_hang["ten"])
print("Địa chỉ:", cua_hang["dia_chi"])
print("Số nhân viên:", cua_hang["so_nhan_vien"])

print("\n=== CÁCH 2: DÙNG get() ===")
# Truy cập an toàn hơn
print("Giờ mở cửa:", cua_hang.get("gio_mo_cua"))
print("Giờ đóng cửa:", cua_hang.get("gio_dong_cua"))

# Truy cập key không tồn tại với giá trị mặc định
print("Số điện thoại:", cua_hang.get("so_dien_thoai", "Chưa cập nhật"))
print("Website:", cua_hang.get("website", "Không có"))
```

```
=== CÁCH 1: DÙNG [] ===
Tên cửa hàng: Siêu thị Mini
Địa chỉ: 456 Phố ABC
Số nhân viên: 15

=== CÁCH 2: DÙNG get() ===
Giờ mở cửa: 7:00
Giờ đóng cửa: 22:00
Số điện thoại: Chưa cập nhật
Website: Không có
```

```
Out[4]: {'ten': 'Siêu thị Mini',
        'dia_chi': '456 Phố ABC',
        'so_nhan_vien': 15,
        'gio_mo_cua': '7:00',
        'gio_dong_cua': '22:00'}
```

```
In [5]: # TODO 3.2: So sánh sự khác biệt khi key không tồn tại

hoc_sinh = {
    "ten": "Trần Thị Mai",
    "lop": "10A",
    "diem_toan": 8.5
}
```

```

print("=== THỬ TRUY CẬP KEY TỒN TẠI ===")
print("Tên (dùng []):", hoc_sinh["ten"])
print("Tên (dùng get()):", hoc_sinh.get("ten"))

print("\n=== THỬ TRUY CẬP KEY KHÔNG TỒN TẠI ===")

# Dùng get() - AN TOÀN
print("Điểm Văn (get()):", hoc_sinh.get("diem_van", "Chưa có điểm"))

# Dùng [] - SẼ BÁO LỖI (đã bỏ trong comment để tránh lỗi)
# print("Điểm Văn ([]):", hoc_sinh["diem_van"]) # Sẽ báo lỗi KeyError

print("\n💡 Kết luận: Dùng get() an toàn hơn khi không chắc key tồn tại!")

```

=== THỬ TRUY CẬP KEY TỒN TẠI ===

Tên (dùng []): Trần Thị Mai

Tên (dùng get()): Trần Thị Mai

=== THỬ TRUY CẬP KEY KHÔNG TỒN TẠI ===

Điểm Văn (get()): Chưa có điểm

💡 Kết luận: Dùng get() an toàn hơn khi không chắc key tồn tại!

```

In [ ]: # TODO 3.3: Thực hành truy cập thông tin

# Dictionary về một bộ phim
phim = {
    "ten": "Avengers: Endgame",
    "nam": 2019,
    "the_loai": "Hành động",
    "thoi_luong": 181, # phút
    "diem_imdb": 8.4
}

# TODO: Sử dụng Dictionary 'phim' ở trên để:
# 1. In ra tên phim
# 2. In ra năm sản xuất
# 3. In ra thời lượng (tính bằng giờ và phút)
# 4. Kiểm tra xem có thông tin về đạo diễn không (dùng get())
# 5. Kiểm tra xem có thông tin về doanh thu không (dùng get() với giá trị mặc định)

print("=== THÔNG TIN PHIM ===")
# TODO: Viết code ở đây

# Tính thời lượng phim
thoi_luong_phut = phim["thoi_luong"]
gio = thoi_luong_phut // 60
phut = thoi_luong_phut % 60
print(f"Thời lượng: {gio} giờ {phut} phút")

```

LÝ THUYẾT 4: Thêm, sửa, xóa dữ liệu

Các thao tác cơ bản:

- **Thêm:** `dict["key_moi"] = "gia_tri_moi"`
- **Sửa:** `dict["key_cu"] = "gia_tri_moi"`
- **Xóa:** `del dict["key"]`

Lưu ý: Nếu key đã tồn tại thì sẽ **sửa**, nếu chưa tồn tại thì sẽ **thêm**.

```
In [ ]: # TODO 4.1: Thực hành thêm, sửa, xóa dữ liệu

# Bắt đầu với Dictionary đơn giản
sach = {
    "ten": "Python cho người mới bắt đầu",
    "tac_gia": "Nguyễn Văn A",
    "gia": 150000
}

print("=== DICTIONARY BAN ĐẦU ===")
print(sach)

print("\n=== THÊM THÔNG TIN MỚI ===")
# Thêm số trang
sach["so_trang"] = 250
# Thêm năm xuất bản
sach["nam_xuat_ban"] = 2023
# Thêm nhà xuất bản
sach["nha_xuat_ban"] = "NXB Giáo dục"
print(sach)

print("\n=== SỬA THÔNG TIN CŨ ===")
# Cập nhật giá mới
sach["gia"] = 120000 # Giảm giá
# Sửa tên tác giả
sach["tac_gia"] = "Nguyễn Văn B"
print(sach)

print("\n=== XÓA THÔNG TIN KHÔNG CẦN ===")
# Xóa năm xuất bản
del sach["nam_xuat_ban"]
print(sach)

print(f"\n📊 Số thông tin hiện tại: {len(sach)} mục")
```

```
In [ ]: # TODO 4.2: Quản lý thông tin học sinh

# Tạo thông tin học sinh ban đầu
hoc_sinh = {
    "ma_so": "HS001",
    "ho_ten": "Lê Thị Lan",
    "lop": "10A1"
}

print("Thông tin ban đầu:", hoc_sinh)

# TODO: Thực hiện các thao tác sau:
# 1. Thêm tuổi: 16
```

```
# 2. Thêm địa chỉ: "Hà Nội"
# 3. Thêm điểm trung bình: 8.2
# 4. Sửa lớp thành "10A2" (chuyển Lớp)
# 5. Sửa điểm trung bình thành 8.5 (cập nhật điểm mới)

# Viết code ở đây:
```

```
print("\nThông tin sau khi cập nhật:", hoc_sinh)
```

```
# TODO: Xóa mã số học sinh (không cần thiết nữa)
# Viết code xóa ở đây:
```

```
print("\nThông tin cuối cùng:", hoc_sinh)
```

LÝ THUYẾT 5: Các phương thức quan trọng

Các phương thức hay dùng:

- `keys()` : Lấy tất cả các key
- `values()` : Lấy tất cả các value
- `items()` : Lấy tất cả cặp key-value
- `len()` : Đếm số cặp key-value
- `"key" in dict` : Kiểm tra key có tồn tại không

In []: *# TODO 5.1: Thực hành các phương thức cơ bản*

```
# Dictionary về điểm thi
bang_diem = {
    "toan": 9.0,
    "van": 8.5,
    "anh": 7.0,
    "ly": 8.0,
    "hoa": 7.5
}

print("=== DICTIONARY GỐC ===")
print(bang_diem)

print("\n=== LẤY TẤT CẢ CÁC KEY (MÔN HỌC) ===")
cac_mon_hoc = list(bang_diem.keys())
print("Các môn học:", cac_mon_hoc)

print("\n=== LẤY TẤT CẢ CÁC VALUE (ĐIỂM SỐ) ===")
cac_diem_so = list(bang_diem.values())
print("Các điểm số:", cac_diem_so)

print("\n=== LẤY TẤT CẢ CẶP KEY-VALUE ===")
print("Chi tiết điểm thi:")
for mon_hoc, diem in bang_diem.items():
    print(f" • {mon_hoc.upper()}: {diem} điểm")
```

```
print("\n=== THỐNG KÊ ===")
print(f"Tổng số môn thi: {len(bang_diem)} môn")
print(f"Điểm trung bình: {sum(bang_diem.values()) / len(bang_diem):.2f}")
print(f"Điểm cao nhất: {max(bang_diem.values())}")
print(f"Điểm thấp nhất: {min(bang_diem.values())}")
```

```
In [ ]: # TODO 5.2: Thực hành kiểm tra key tồn tại

# Dictionary về kho hàng
kho_hang = {
    "ao_thun": 120,
    "quan_jeans": 80,
    "giay_the_thao": 45,
    "mu_luoi_trai": 30,
    "tui_xach": 25
}

print("=== KHO HÀNG HIỆN TẠI ===")
for san_pham, so_luong in kho_hang.items():
    print(f"• {san_pham.replace('_', ' ').title()}: {so_luong} cái")

print("\n=== KIỂM TRA SẢN PHẨM ===")

# Danh sách sản phẩm cần kiểm tra
can_kiem_tra = ["ao_thun", "quan_short", "giay_the_thao", "kinh_mat", "mu_luoi_trai"]

for san_pham in can_kiem_tra:
    if san_pham in kho_hang:
        so_luong = kho_hang[san_pham]
        tinh_trang = "Còn hàng" if so_luong > 0 else "Hết hàng"
        print(f"✅ {san_pham.replace('_', ' ').title()}: {so_luong} cái - {tinh_trang}")
    else:
        print(f"❌ {san_pham.replace('_', ' ').title()}: Không có trong kho")
```

```
In [ ]: # TODO 5.3: Thực hành với dữ liệu của bạn

# TODO: Tạo Dictionary về sở thích của bạn với ít nhất 5 hoạt động
# Key: tên hoạt động, Value: mức độ yêu thích (1-10)
so_thich = {
    # TODO: Điền ít nhất 5 sở thích với điểm từ 1-10
    # Ví dụ: "doc_sach": 9,
}

# TODO: Sau khi tạo xong Dictionary, chạy các lệnh dưới để phân tích

if len(so_thich) > 0:
    print("=== PHÂN TÍCH SỞ THÍCH CỦA BẠN ===")
    print(f"Số sở thích: {len(so_thich)}")

    # Tìm sở thích yêu thích nhất
    yeu_thich_nhat = max(so_thich, key=so_thich.get)
    print(f"Yêu thích nhất: {yeu_thich_nhat.replace('_', ' ').title()} ({so_thich[yeu_thich_nhat]})")

    # Tìm sở thích ít thích nhất
    it_thich_nhat = min(so_thich, key=so_thich.get)
```



```

print(f"Ít thích nhất: {it_thich_nhat.replace('_', ' ').title()} ({so_thich[it_

# Tính điểm trung bình
diem_tb = sum(so_thich.values()) / len(so_thich)
print(f"Điểm trung bình: {diem_tb:.2f}/10")

print("\nChi tiết:")
for hoat_dong, diem in so_thich.items():
    print(f" • {hoat_dong.replace('_', ' ').title()}: {diem}/10")
else:
    print("⚠️ Bạn chưa điền sở thích! Hãy thêm ít nhất 5 sở thích vào Dictionary")

```

LÝ THUYẾT 6: Khi nào dùng Dictionary thay vì List?

Dictionary phù hợp khi:

- Dữ liệu có **ý nghĩa rõ ràng** (tên, tuổi, điểm...)
- Cần **truy cập nhanh** bằng tên thay vì vị trí
- Dữ liệu **không cần thứ tự** cố định
- Cần **linh hoạt** thêm/xóa thuộc tính

List phù hợp khi:

- Dữ liệu có **thứ tự** quan trọng
- Cần **lặp qua** từng phần tử theo thứ tự
- Dữ liệu **cùng loại** (danh sách tên, số...)
- Cần **index** để truy cập

In []: *# TODO 6.1: So sánh Dictionary vs List trong thực tế*

```

print("=== VÍ DỤ 1: THÔNG TIN HỌC SINH ===")
print("Dictionary phù hợp vì có nhiều thuộc tính khác nhau:\n")

# Dùng Dictionary - RÕ RÀNG
hoc_sinh_dict = {
    "ma_so": "HS001",
    "ho_ten": "Nguyễn Văn Nam",
    "lop": "10A1",
    "diem_tb": 8.5,
    "xep_loai": "Giỏi",
    "so_dien_thoai": "0123456789"
}

print("Dictionary:")
print(f"Tên: {hoc_sinh_dict['ho_ten']}")
print(f"Lớp: {hoc_sinh_dict['lop']}")
print(f"Điểm TB: {hoc_sinh_dict['diem_tb']}")

# Dùng List - KHÓ HIỂU
hoc_sinh_list = ["HS001", "Nguyễn Văn Nam", "10A1", 8.5, "Giỏi", "0123456789"]

```

```

print("\nList (khó hiểu):")
print(f"Tên: {hoc_sinh_list[1]}") # Phải nhớ vị trí 1 là tên
print(f"Lớp: {hoc_sinh_list[2]}") # Vị trí 2 là Lớp
print(f"Điểm TB: {hoc_sinh_list[3]}") # Vị trí 3 là điểm

print("\n🤔 Nhận xét: Dictionary rõ ràng hơn nhiều!")

print("\n" + "-"*50)
print("=== VÍ DỤ 2: DANH SÁCH TÊN LỚP ===")
print("List phù hợp vì cần thứ tự và cùng loại dữ liệu:\n")

# Dùng List - PHÙ HỢP
danh_sach_lop = ["10A1", "10A2", "10A3", "10B1", "10B2", "10C1"]

print("List (phù hợp):")
for i, ten_lop in enumerate(danh_sach_lop, 1):
    print(f"{i}. {ten_lop}")

print(f"\nLớp đầu tiên: {danh_sach_lop[0]}")
print(f"Lớp cuối cùng: {danh_sach_lop[-1]}")
print(f"Tổng số lớp: {len(danh_sach_lop)}")

```

In []: # TODO 6.2: Thực hành quyết định dùng Dictionary hay List

```

print("=== BÀI TẬP: CHỌN KIỂU DỮ LIỆU PHÙ HỢP ===")
print("Với mỗi trường hợp, hãy quyết định nên dùng Dictionary hay List\n")

# TRƯỜNG HỢP 1: Lưu thông tin về một cuốn sách
print("Trường hợp 1: Thông tin sách")
print("Cần lưu: tên sách, tác giả, giá, số trang, thể loại")
print("👉 Nên dùng: Dictionary (vì có nhiều thuộc tính khác nhau)\n")

sach = {
    "ten": "Lập trình Python",
    "tac_gia": "Nguyễn Văn A",
    "gia": 200000,
    "so_trang": 300,
    "the_loai": "Giáo trình"
}
print("Dictionary:", sach)

# TRƯỜNG HỢP 2: Lưu danh sách điểm kiểm tra
print("\n" + "-"*40)
print("Trường hợp 2: Điểm kiểm tra trong tháng")
print("Cần lưu: 8.5, 7.0, 9.0, 8.0, 7.5")
print("👉 Nên dùng: List (vì cùng loại dữ liệu, có thứ tự thời gian)\n")

diem_kiem_tra = [8.5, 7.0, 9.0, 8.0, 7.5]
print("List:", diem_kiem_tra)
print(f"Điểm trung bình: {sum(diem_kiem_tra)/len(diem_kiem_tra):.2f}")
print(f"Lần kiểm tra đầu tiên: {diem_kiem_tra[0]}")

# TODO: Bạn hãy quyết định cho các trường hợp sau
print("\n" + "-"*40)
print("TODO: Bạn hãy quyết định nên dùng Dictionary hay List:")

```

```

print("\n1. Lưu thông tin nhân viên (tên, tuổi, lương, phòng ban)")
print("    👉 Nên dùng: _____ (điền vào chỗ trống)")

print("\n2. Lưu nhiệt độ hàng ngày trong tuần (25°C, 27°C, 26°C...)")
print("    👉 Nên dùng: _____ (điền vào chỗ trống)")

print("\n3. Lưu thông tin xe hơi (hãng, màu, năm sản xuất, giá)")
print("    👉 Nên dùng: _____ (điền vào chỗ trống)")

print("\n💡 Gợi ý: Nếu dữ liệu có nhiều thuộc tính khác nhau → Dictionary")
print("        Nếu dữ liệu cùng loại, có thứ tự → List")

```

LÝ THUYẾT 7: Dictionary lồng nhau và kết hợp với List

Trong thực tế, chúng ta thường kết hợp Dictionary và List:

- **Dictionary chứa List:** Một đối tượng có nhiều thuộc tính dạng danh sách
- **List chứa Dictionary:** Danh sách nhiều đối tượng
- **Dictionary lồng nhau:** Dictionary chứa Dictionary khác

In [6]: *# TODO 7.1: Dictionary chứa List*

```

print("=== VÍ DỤ: THÔNG TIN LỚP HỌC ===")

lop_10a1 = {
    "ten_lop": "10A1",
    "si_so": 35,
    "giao_vien_chu_nhiem": "Cô Hương",
    "phong_hoc": "A201",
    "cac_mon_hoc": ["Toán", "Văn", "Anh", "Lý", "Hóa", "Sinh"], # List trong Dicti
    "hoc_sinh_gioi": ["Nguyễn Văn A", "Trần Thị B", "Lê Văn C"] # List khác
}

print(f"Lớp: {lop_10a1['ten_lop']}")
print(f"Sĩ số: {lop_10a1['si_so']} học sinh")
print(f"GVCN: {lop_10a1['giao_vien_chu_nhiem']}")
print(f"Phòng học: {lop_10a1['phong_hoc']}")

print("\nCác môn học:")
for i, mon in enumerate(lop_10a1['cac_mon_hoc'], 1):
    print(f" {i}. {mon}")

print("\nHọc sinh giỏi:")
for hs in lop_10a1['hoc_sinh_gioi']:
    print(f" • {hs}")

print(f"\nTổng số môn học: {len(lop_10a1['cac_mon_hoc'])}")
print(f"Số học sinh giỏi: {len(lop_10a1['hoc_sinh_gioi'])}")

```

=== VÍ DỤ: THÔNG TIN LỚP HỌC ===

Lớp: 10A1

Sĩ số: 35 học sinh

GVCN: Cô Hương

Phòng học: A201

Các môn học:

1. Toán
2. Văn
3. Anh
4. Lý
5. Hóa
6. Sinh

Học sinh giỏi:

- Nguyễn Văn A
- Trần Thị B
- Lê Văn C

Tổng số môn học: 6

Số học sinh giỏi: 3

```
In [ ]: # TODO 7.2: List chứa Dictionary

print("=== VÍ DỤ: DANH SÁCH HỌC SINH ===")

danhsachhocsinh = [
    {"ten": "Nguyễn Văn An", "tuoi": 16, "diem_tb": 8.5, "lop": "10A1"},
    {"ten": "Trần Thị Bình", "tuoi": 15, "diem_tb": 9.0, "lop": "10A1"},
    {"ten": "Lê Văn Cường", "tuoi": 16, "diem_tb": 7.5, "lop": "10A2"},
    {"ten": "Phạm Thị Dung", "tuoi": 15, "diem_tb": 8.8, "lop": "10A2"}
]

print(f"Tổng số học sinh: {len(danhsachhocsinh)}")

print("\nThông tin từng học sinh:")
for i, hs in enumerate(danhsachhocsinh, 1):
    print(f"{i}. {hs['ten']} - {hs['tuoi']} tuổi - ĐTB: {hs['diem_tb']} - Lớp: {hs['lop']}")

# Tính điểm trung bình của cả nhóm
tong_diem = sum(hs['diem_tb'] for hs in danhsachhocsinh)
diem_tb_chung = tong_diem / len(danhsachhocsinh)
print(f"\nĐiểm trung bình chung: {diem_tb_chung:.2f}")

# Tìm học sinh có điểm cao nhất
hs_gioi_nhat = max(danhsachhocsinh, key=lambda hs: hs['diem_tb'])
print(f"Học sinh giỏi nhất: {hs_gioi_nhat['ten']} ({hs_gioi_nhat['diem_tb']} điểm)")
```

```
In [ ]: # TODO 7.3: Thực hành tạo cấu trúc dữ liệu phức tạp
```

```
# TODO: Tạo thông tin về nhà hàng với menu
nha_hang = {
    "ten": "Nhà hàng Hương Việt",
    "dia_chi": "123 Phố Cổ, Hà Nội",
    "so_ban": 20,
    "gio_mo_cua": "08:00",
```

```

"gio_dong_cua": "22:00",

# TODO: Thêm menu với các món ăn (Dictionary chứa Dictionary)
"menu": {
    "mon_chinh": {
        # TODO: Thêm ít nhất 3 món chính với giá
    },
    "do_uong": {
        # TODO: Thêm ít nhất 3 đồ uống với giá
    },
    "trang_mieng": {
        # TODO: Thêm ít nhất 2 món tráng miệng với giá
    }
},

# TODO: Thêm danh sách nhân viên (List chứa Dictionary)
"nhan_vien": [
    # TODO: Thêm thông tin ít nhất 3 nhân viên
    # Mỗi nhân viên có: tên, tuổi, chức vụ, lương
]
}

# Sau khi điền đầy đủ thông tin, chạy code phân tích bên dưới

print("=== THÔNG TIN NHÀ HÀNG ===")
print(f"Tên: {nha_hang['ten']}")
print(f"Địa chỉ: {nha_hang['dia_chi']}")
print(f"Giờ hoạt động: {nha_hang['gio_mo_cua']} - {nha_hang['gio_dong_cua']}")

if nha_hang['menu']['mon_chinh']: # Kiểm tra đã điền menu chưa
    print("\n=== MENU ===")
    for loai_mon, cac_mon in nha_hang['menu'].items():
        print(f"\n{loai_mon.replace('_', ' ').title()}:")
        for ten_mon, gia in cac_mon.items():
            print(f"    • {ten_mon}: {gia:,} VND")

if nha_hang['nhan_vien']: # Kiểm tra đã điền nhân viên chưa
    print("\n=== NHÂN VIÊN ===")
    for nv in nha_hang['nhan_vien']:
        print(f"    • {nv['ten']} - {nv['chuc_vu']} - {nv['luong']:,} VND")

    # Tính tổng Lương
    tong_luong = sum(nv['luong'] for nv in nha_hang['nhan_vien'])
    print(f"\nTổng chi phí lương: {tong_luong:,} VND")
else:
    print("\n⚠️ Bạn chưa điền thông tin menu và nhân viên!")

```

BÀI TẬP TỔNG HỢP

Bài tập cuối: Xây dựng hệ thống quản lý học sinh đơn giản

In []: # TODO: BÀI TẬP CUỐI - Hệ thống quản lý học sinh

Bước 1: Tạo danh sách học sinh trống

```

he_thong_quan_ly = {
    "ten_truong": "THPT ABC",
    "hoc_sinh": [], # List chứa Dictionary
    "thong_ke": {
        "tong_so_hs": 0,
        "diem_tb_chung": 0
    }
}

# Bước 2: Hàm thêm học sinh mới
def them_hoc_sinh(ten, tuoi, lop, diem_toan, diem_van, diem_anh):
    """Thêm học sinh mới vào hệ thống"""
    diem_tb = (diem_toan + diem_van + diem_anh) / 3

    hoc_sinh_moi = {
        "ten": ten,
        "tuoi": tuoi,
        "lop": lop,
        "diem": {
            "toan": diem_toan,
            "van": diem_van,
            "anh": diem_anh,
            "trung_binh": round(diem_tb, 2)
        }
    }

    he_thong_quan_ly["hoc_sinh"].append(hoc_sinh_moi)
    print(f"✅ Đã thêm học sinh: {ten}")

# Bước 3: TODO - Thêm ít nhất 5 học sinh
print("=== THÊM HỌC SINH VÀO HỆ THỐNG ===")

# TODO: Gọi hàm them_hoc_sinh() để thêm 5 học sinh
# Ví dụ: them_hoc_sinh("Nguyễn Văn A", 16, "10A1", 8.5, 7.5, 8.0)

# Bước 4: Hàm hiển thị thông tin
def hien_thi_danh_sach():
    """Hiển thị danh sách tất cả học sinh"""
    print(f"\n=== DANH SÁCH HỌC SINH - {he_thong_quan_ly['ten_truong']} ===")

    if not he_thong_quan_ly["hoc_sinh"]:
        print("Chưa có học sinh nào trong hệ thống.")
        return

    for i, hs in enumerate(he_thong_quan_ly["hoc_sinh"], 1):
        print(f"\n{i}. {hs['ten']} ({hs['tuoi']} tuổi) - Lớp {hs['lop']}")
        print(f"    Điểm: Toán {hs['diem']['toan']} | Văn {hs['diem']['van']} | Anh")
        print(f"    Điểm TB: {hs['diem']['trung_binh']}")

# Bước 5: Hàm thống kê
def thong_ke_he_thong():
    """Thống kê toàn bộ hệ thống"""
    ds_hs = he_thong_quan_ly["hoc_sinh"]

```

```

if not ds_hs:
    print("\n⚠ Chưa có dữ liệu để thống kê.")
    return

print("\n=== THỐNG KÊ HỆ THỐNG ===")
print(f"Tổng số học sinh: {len(ds_hs)}")

# Tính điểm trung bình chung
tong_diem_tb = sum(hs['diem']['trung_binh'] for hs in ds_hs)
diem_tb_chung = tong_diem_tb / len(ds_hs)
print(f"Điểm trung bình chung: {diem_tb_chung:.2f}")

# Tìm học sinh giỏi nhất
hs_gioi_nhat = max(ds_hs, key=lambda hs: hs['diem']['trung_binh'])
print(f"Học sinh giỏi nhất: {hs_gioi_nhat['ten']} ({hs_gioi_nhat['diem']['trung_binh']:.2f})")

# Đếm theo Lớp
thong_ke_lop = {}
for hs in ds_hs:
    lop = hs['lop']
    if lop in thong_ke_lop:
        thong_ke_lop[lop] += 1
    else:
        thong_ke_lop[lop] = 1

print("\nThống kê theo lớp:")
for lop, so_luong in thong_ke_lop.items():
    print(f" • Lớp {lop}: {so_luong} học sinh")

# Bước 6: TODO - Chạy các chức năng
print("\n" + "="*50)
print("CHẠY CÁC CHỨC NĂNG:")

# TODO: Gọi hàm hiển thị danh sách
# hien_thi_danh_sach()

# TODO: Gọi hàm thống kê
# thong_ke_he_thong()

print("\n💡 Sau khi thêm đủ 5 học sinh, hãy bỏ comment và chạy 2 hàm trên!")

```

<https://create.kahoot.it/share/w27-t4/d6386054-631e-48f7-9afc-e5510ba05d56>

TÓM TẮT KIẾN THỨC

✓ Những gì bạn đã học:

1. Dictionary cơ bản

- Cú pháp: `{"key": "value"}`
- Lưu trữ theo cặp key-value
- Truy cập: `dict["key"]` hoặc `dict.get("key")`

2. Các thao tác cơ bản

- Thêm: `dict["key_moi"] = "value"`
- Sửa: `dict["key_cu"] = "value_moi"`
- Xóa: `del dict["key"]`

3. Các phương thức quan trọng

- `keys()` : Lấy tất cả key
- `values()` : Lấy tất cả value
- `items()` : Lấy cặp key-value
- `get()` : Truy cập an toàn
- `len()` : Đếm số phần tử

4. Khi nào dùng Dictionary




- ☒ Dữ liệu có ý nghĩa rõ ràng
- ☒ Cần truy cập bằng tên
- ☒ Nhiều thuộc tính khác nhau
- ☒ Không cần thứ tự cố định

5. Kết hợp với List

- Dictionary chứa List
- List chứa Dictionary
- Dictionary lồng nhau


Chuẩn bị cho buổi sau:

Buổi chiều chúng ta sẽ thực hành:

-  Xây dựng **Từ điển Anh-Việt**
-  Tạo ứng dụng **Danh bạ điện thoại**
-  Quản lý dữ liệu phức tạp

Thử thách bản thân:

Hãy hoàn thành tất cả các TODO trong bài học này để nắm vững Dictionary!

 **Chúc bạn học tập hiệu quả! Hẹn gặp lại ở buổi thực hành!**