



PYTHON FUNDAMENTALS

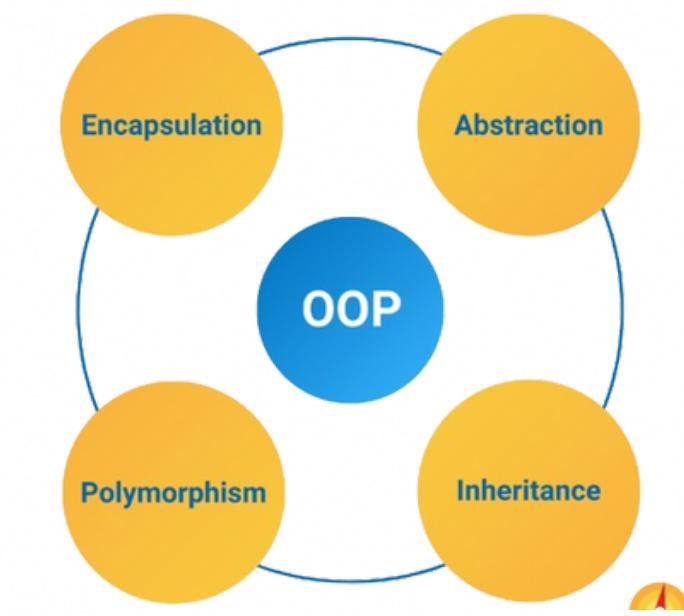


VÀI PHÚT LÀM
QUEN VỚI SYNTAX
PYTHON

NỘI DUNG ÔN TẬP



SYNTAX CƠ BẢN



HƯỚNG ĐỐI TƯỢNG (OOP)

ÔN TẬP CÚ PHÁP QUA DỰ ÁN QUẢN LÝ SẢN PHẨM TIKI

Đặc tả yêu cầu:

1. Tạo list gồm 3 sản phẩm
2. Thực hiện chức năng duyệt, thêm, xoá, sửa trên list (CRUD)
3. Thực hiện tính năng tìm sản phẩm theo tên

The screenshot shows the homepage of the Tiki website. At the top, there's a search bar with the placeholder "Giao nhanh 2H & đúng khung giờ" and a "Tim kiếm" button. To the right are links for "Trang chủ", "Astra", "Tài khoản", and a shopping cart icon. Below the header, there's a navigation menu with categories like "khô đẹp", "nhà cửa", "sách", etc. Two prominent promotional banners are visible: one for "TRUNG THU ĐOÀN VIÊN KHOẺ MẠNH VỮNG VÀNG" with coupons for 10K, 50K, and 20K, and another for "LOTTE FINANCE" with a "GIẢM 200K CHO ĐƠN 500K" offer. The main content area features a "Sản phẩm bán chạy" section with a grid of smartphone products from brands like Apple, Xiaomi, Redmi, Oppo, and Samsung. Below this is a "Bạn có thể thích" section showing various household items like SunMate diapers and towels.

HƯỚNG DẪN THỰC HIỆN

1. Tạo list gồm 3 sản phẩm

Cú pháp khai báo **list**
tương tự **mảng**

list = []

Dictionary trong python
gần giống với object

d = {key:value}

Lưu ý: Trong python **không có** từ khoá khai báo biến cũng
như kiểu dữ liệu (python sẽ tự hiểu khi ta sử dụng biến)

```
demo.py > ...
1  # Khởi tạo và gán danh sách sản phẩm
2  products = [
3      {
4          "id": 1,
5          "name": "Laptop",
6          "price": 1000,
7          "description": "A good laptop",
8          "image": f"https://picsum.photos/id/1/200/300"
9      },
10     {
11         "id": 2,
12         "name": "Phone",
13         "price": 500,
14         "description": "A smart phone",
15         "image": f"https://picsum.photos/id/2/200/300"
16     },
17     {
18         "id": 3,
19         "name": "Tablet",
20         "price": 700,
21         "description": "An Android tablet",
22         "image": f"https://picsum.photos/id/3/200/300"
23     }
24 ]
25 }
```

HƯỚNG DẪN THỰC HIỆN

2. Duyệt, thêm, xoá, sửa

```
29 # In danh sách sản phẩm  
30 for product in products:  
31     print(product)
```

Dùng vòng lặp **for** để duyệt list
lưu ý **python** sử dụng dấu **:** thay cho **{**
so với ngôn ngữ khác (**js,c++,java ...**)

for [item] in [list]:
 //Xử lý

Thêm 1 phần tử vào list bằng cú pháp

list.append(item)

list.append(item)

```
26 #Thêm sản phẩm  
27 products.append({  
28     "id": 4,  
29     "name": "Camera",  
30     "price": 300,  
31     "description": "A digital camera",  
32     "image": "https://picsum.photos/id/4/200/300"  
33 })
```

HƯỚNG DẪN THỰC HIỆN

2. Duyệt, thêm, xoá, sửa

Tương tự các cú pháp **lambda filter**
các ngôn ngữ khác **python** kết hợp
for if (list comprehension) để filter

products= [products_filter]
Xoá bằng cách filter

```
34 # Xoá sản phẩm theo ID
35 print("----- After removing product with ID 2 -----")
36 products = [product for product in products if product["id"] != 2]
```

```
18 # Cập nhật sản phẩm có id là 1
19 for product in products:
20     if product["id"] == 1:
21         product["name"] = "Macbook"
22         product["price"] = 1500
```

Để **update** trong **python** chỉ cần dùng
vòng lặp để duyệt list và update
tương tự các ngôn ngữ khác

HƯỚNG DẪN THỰC HIỆN

3. Tìm kiếm sản phẩm theo tên hoặc theo 1 tiêu chí nào đó

Python là ngôn ngữ đa dạng về cách viết khá sát nghĩa với ngôn ngữ tự nhiên vì vậy cú pháp python có nhiều cách diễn đạt

```
34 # Tìm sản phẩm theo tên  
35 for product in products:  
36     if "Phone" in product["name"]:  
37         print(product)
```

Dùng **if in** trong
python để diễn đạt

Cách này được dùng nhiều bởi nó so sánh
được địa chỉ (giá trị reference value)

```
27 #Tìm sản phẩm theo tên  
28 for product in products:  
29     if product["name"] == "Phone":  
30         print(product)
```

Tương tự với cách tìm id

BÀI TẬP

Dựa vào bài tập vừa rồi hãy xây dựng một list sinh viên với các thuộc tính sau: mã sinh viên, tên sinh viên, điểm toán, điểm văn, điểm hóa.

Đặc tả yêu cầu:

1. Tạo list gồm 5 sinh viên
2. In thông tin các sinh viên có điểm trung bình lớn hơn 5
3. In ra các sinh viên có điểm hóa dưới 5



OOP - HƯỚNG ĐỐI TƯỢNG - QUA BÀI TOÁN QUẢN LÝ NHÂN VIÊN

Đặc tả yêu cầu:

1. Xây dựng lớp đối tượng nhân viên bao gồm các thông tin cơ bản (maNhanVien, tenNhanVien, luongCoBan, chucVu, heSoLuong). Xây dựng phương thức tính lương nhân viên dựa trên hệ số lương
2. Mở rộng bài toán với role QuanLy và GiamDoc có thêm bonus và danh sách phòng ban quản lý, giám đốc thì có danh sách chi nhánh quản lý



1. XÂY DỰNG CLASS NHÂN VIÊN VỚI CÁC THÔNG TIN CƠ BẢN

Như những ngôn ngữ lập trình khác python dùng từ khoá **class** để khai báo **lớp đối tượng**

Với **python** hàm và phương thức khai báo là giống nhau bắt đầu bằng từ **def**, khác nhau ở chỗ hàm thì không có **self** và khai báo bên ngoài **class**. Còn phương thức thì ngược lại.

Tạo đối tượng **python** không cần từ khoá **new** như gọi hàm bình thường

self là từ khoá luôn có ở các phương thức của **python**. Tương tự con trỏ **this** của các ngôn ngữ lập trình khác. Ví dụ bên dưới là hàm khởi tạo có tham số

```

1 class NhanVien:
2     def __init__(self, maNhanVien, tenNhanVien, LuongCoBan, chucVu, heSoLuong):
3         self.maNhanVien = maNhanVien
4         self.tenNhanVien = tenNhanVien
5         self.luongCoBan = LuongCoBan
6         self.chucVu = chucVu
7         self.heSoLuong = heSoLuong
8
9     def tinhLuong(self):
10        """Tính lương dựa trên lương cơ bản và hệ số lương"""
11        return self.luongCoBan * self.heSoLuong
12
13    def hienThiThongTin(self):
14        """Hiển thị thông tin cơ bản của nhân viên"""
15        return f"""
16            ID: {self.maNhanVien},
17            Tên: {self.tenNhanVien},
18            Chức vụ: {self.chucVu},
19            Lương: {self.tinhLuong()}"""
20
21
22
23
24 # Sử dụng lớp NhanVien
25 nv1 = NhanVien("NV001", "Nguyen Van A", 1000, "Developer", 3.5)
26 print(nv1.hienThiThongTin()) # ID: NV001, Tên: Nguyen Van A, Chức vụ: Developer,

```

2. MỞ RỘNG NGHIỆP VỤ VỚI CLASS GIÁM ĐỐC VÀ CLASS QUẢN LÝ VÀ 1 SỐ TÍNH NĂNG

Trong **python** để tái sử dụng lại các giá trị ở **class cha** khi kế thừa thì ta dùng từ khoá **super().__init__** khá giống js

Tạo đối tượng **python** từ class
dẫn xuất

```
15 class QuanLy(NhanVien):
16     def __init__(self, maNhanVien, tenNhanVien, LuongCoBan, heSoLuong, tienThuong, danhSachPhongBan):
17         super().__init__(maNhanVien, tenNhanVien, LuongCoBan, "Quản Lý", heSoLuong)
18         self.tienThuong = tienThuong
19         self.danhSachPhongBan = danhSachPhongBan
20
21     def tinhLuong(self):
22         return super().tinhLuong() + self.tienThuong
23
24     def hienThiThongTin(self):
25         info = super().hienThiThongTin()
26         return f"{info}, Phòng ban quản lý: {', '.join(self.danhSachPhongBan)}"
```

```
# Sử dụng các lớp trên
ql1 = QuanLy("QL001", "Nguyen Van B", 2000, 4, 500, ["Kỹ thuật", "Kế toán"])
```

Khi muốn kế thừa các giá trị thuộc tính và phương thức ở lớp cha (**public** hoặc **protected**). Trong **python** ta chỉ cần (**classCha**)

2. MỞ RỘNG NGHIỆP VỤ VỚI CLASS GIÁM ĐỐC VÀ CLASS QUẢN LÝ VÀ 1 SỐ TÍNH NĂNG

ghi đè (**override**) khá đơn giản chỉ cần đặt trùng tên phương thức với class cha và định nghĩa lại hàm.

Lưu ý: các thuộc tính hay phương thức nào của cha cần gọi lại dùng **super()** thay vì **self**

if else trong **python** khá đơn giản ngoài ra ta còn có thể sử dụng các điều kiện **and** hay **or** thay cho **&&** ||

Tạo đối tượng **python** từ class
dẫn xuất

Tương tự với class giám đốc các thuộc tính bổ sung được khai báo như bình thường

```

32  v class GiamDoc(NhanVien):
33  v     def __init__(self, maNhanVien, tenNhanVien, LuongCoBan, heSoLuong, tienThuong, danhSachChiNhanh):
34  v         super().__init__(maNhanVien, tenNhanVien, LuongCoBan, "Giám Đốc", heSoLuong)
35  v         self.tienThuong = tienThuong
36  v         self.danhSachChiNhanh = danhSachChiNhanh
37
38  v     def tinhLuong(self):
39  v         # Giả sử, nếu Giám Đốc quản lý trên 3 chi nhánh thì có thêm tiền thưởng
40  v         if len(self.danhSachChiNhanh) > 3:
41  v             return super().tinhLuong() + self.tienThuong + 1000 # thêm 1000 đơn vị tiền thưởng
42  v         else:
43  v             return super().tinhLuong() + self.tienThuong
44
45
46  v     def hienThiThongTin(self):
47  v         info = super().hienThiThongTin()
48  v         return f'{info}, Chi nhánh quản lý: {", ".join(self.danhSachChiNhanh)}'
49

```

```

54     gd1 = GiamDoc("GD001", "Le Thi C", 5000, 5, 2000, ["Chi nhánh A", "Chi nhánh B", "Chi nhánh C", "Chi nhánh D"])
55

```

“ Bài tập tổng kết oop - Quản lý Học viên của Trung tâm Đào tạo

Trung tâm đào tạo lập trình Techx cần lưu trữ thông tin quản lý với các thông tin như sau:

1. Học viên:

- Đối tượng: HocVien(maHV, tenHV, ngaySinh, khoaHoc)
- Phương thức:
 - dangKyKhoaHoc(khoaHoc): Đăng ký một khóa học mới.
 - hienThiKhoaHoc(): Hiển thị danh sách khóa học đã đăng ký.

2. Khóa học:

- Đối tượng: KhoaHoc(maKhoaHoc, tenKhoaHoc, hinThuc, hocPhi)
- Phương thức:
 - thongTinKhoaHoc(): Hiển thị thông tin chi tiết về khóa học.

Yêu cầu thực hiện:

- Tạo các lớp đối tượng cho Học viên, Khóa học.
- Định nghĩa các phương thức theo yêu cầu.
- Tạo một số đối tượng (ít nhất 2 học viên, 2 khóa học) và thực hiện một số thao tác trên chúng (đăng ký khóa học, hiển thị khóa học). Học viên có thể mở rộng tính năng ứng dụng

