TRƯỜNG ĐẠI HỌC ĐÀ LẠT

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙠🕮🙢**



**BÁO CÁO WORD**

**VIẾT CHƯƠNG TRÌNH QUẢN LÝ KHÁCH HÀNG**

**BẰNG NNLT PYTHON**

**GVHD: Nguyễn Thị Lương**

**Nhóm 3**

**Sinh viên thực hiện:**

**2411849 - Nguyễn Đình Quốc Huy**

**2411876 - Nguyễn Nguyên Bảo Long**

**2411938 - Dương Minh Tuấn**

***Đà Lạt, tháng 11 năm 2024***

**I. Tìm hiểu về ngôn ngữ lập trình Python**

**1. Giới thiệu ngôn ngữ lập trình Python**

Python là một ngôn ngữ lập trình cấp cao, dễ học và dễ đọc, được phát triển bởi **Guido van Rossum** vào năm 1991. Nổi bật nhờ cú pháp đơn giản, thư viện phong phú, dễ đọc dễ hiểu phù hợp với người mới học lập trình và khả năng ứng dụng đa dạng (phát triển web, trí tuệ nhân tạo, khoa học dữ liệu, tự động hóa, v.v.). Python đã trở thành một trong những ngôn ngữ phổ biến nhất hiện nay.

**2. Tìm hiểu về ngôn ngữ lập trình Python**

**a) Nhập, xuất**

**\* Hàm nhập dữ liệu (input):** Dùng để nhận dữ liệu từ người dùng.

|  |  |
| --- | --- |
| a = input('Nhập 1 số nguyên: ')  print (type(a))  a = int(input("Nhập 1 số nguyên: "))  print(type(a)) | Nhập 1 số nguyên: 12  <class 'str'>  Nhập 1 số nguyên: 12  <class 'int'> |

**\* Hàm xuất dữ liệu (print):** Dùng để xuất thông tin ra màn hình.

|  |  |
| --- | --- |
| print ("Hello", "python")  print (f"Số nguyên vừa nhập là {a}") | Hello python  Số nguyên vừa nhập là 12 |

**b) Kiểu dữ liệu:**

**\* Kiểu dữ liệu số:** Có 3 kiểu dữ liệu số:

**- Số nguyên (interger):** Không giới hạn ký tự.

**- Số thực (float)**:

+ Lưu lớn nhất xấp xỉ 1.8\*10308, các giá trị lớn hơn được mô tả bởi chuỗi inf (Infinity).

+ Giá trị số thực nhỏ nhất có thể lưu là 5.0 \* 10-324, các giá trị nhỏ hơn số này được coi là 0.

**- Số phức (complex):** Gồm phần thực và phần ảo, có thể trích xuất phần thực, phần ảo của số phức x bằng x.real và x.imag.

+ Độ chính xác: tối đa 15-17 chữ số thập phân.

+ Giá trị lớn/nhỏ nhất như kiểu float.

\* **Kiểu đúng sai**: Kiểu bool chỉ lưu 2 giá trị True hoặc False.

**\* Kiểu chuỗi (string):** Được đặt trong dấu nháy đơn hoặc kép, trên 1 dòng, trong trường hợp xâu nhiều dòng ta đặt giữa 3 dấu nháy kép

\* **Ép kiểu dữ liệu**: Khi muốn chỉ định kiểu cho 1 biến nào đó bạn có thể ép kiểu cho nó.

**c) Câu lệnh điều khiển:**

**\* Câu lệnh if, else, elif:**

if a < b:

print (a, " nhỏ hơn ", b)

elif a == b:

print (a, " bằng ", b)

else:

print (a, " lớn hơn ",b)

**\* Vòng lặp for:**

**-** Cú pháp:

for i in range(start, stop, step):

- Trong đó:

+ start : Giá trị bắt đầu của dãy số (Mặc định là 0)

+ stop : Giá trị cuối cùng của dãy số (Cận này không được lấy)

+ step : Bước nhảy của dãy số (Mặc định là 1)

- Ví dụ:

|  |  |
| --- | --- |
| for i in range(1, 5, 1):  print (i, end = " ") | 1 2 3 4 |

**\* Vòng lặp while:**

- Cú pháp:

While điều kiện:

# code khi điều kiện đúng

else:

# code khi điều khiện sai

- Ví dụ:

|  |  |
| --- | --- |
| n = 1  while (n <= 3):  print ("Vòng lặp tương ứng với n =", n)  n += 1 | Vòng lặp tương ứng với n = 1  Vòng lặp tương ứng với n = 2  Vòng lặp tương ứng với n = 3 |

**d) Hàm:**

Hàm là một khối mã được định nghĩa để thực hiện một nhiệm vụ cụ thể. Hàm trong Python được định nghĩa bằng từ khóa “ def ”.

Ví dụ:

def chan\_le():

n = int(input("Nhập số nguyên n:"))

if (n % 2 == 0):

print("Chẵn")

else:

print("Lẻ")

Một số hàm có sẵn: abs(), pow(), max(), min(),…

**e) Danh sách (List):**

Danh sách là một cấu trúc dữ liệu cho phép lưu trữ nhiều giá trị trong một biến.

Ví dụ:  
danh\_sach = [1, 2, 3, 4]  
print(danh\_sach)

Một số chức năng:

- append(): Thêm vào cuối danh sách

- insert(): Thêm vào vị trí

- pop(), del(): xóa khỏi danh sách

- count(): đếm số lần xuất hiện trong danh sách

- reverse(): lật ngược danh sách

- sort(): sắp xếp

**f) Lớp (Class):**

Là 1 cách để tạo ra các đối tượng với thuộc tính và phương thức chung, là một phần quan trọng của lập trình hướng đối tượng. Được định nghĩa bằng từ khóa “ ***class*** ”.

Tên lớp được viết theo kiểu mỗi chữ cái đầu của từ viết hoa.

***Đối tượng:*** khi 1 lớp được định nghĩa, có thể sử dụng lớp đó để tạo ra các đối tượng, là phiên bản cụ thể của lớp.

***Thuộc tính:*** là các biến gắn liền với một đối tượng, được khai báo và gán giá trị trong phương thức \_\_init\_\_.

***Phương thức:*** là các hàm được định nghĩa trong lớp, luôn có tham số self dùng để tham chiếu đến đối tượng hiện tại.

***Hàm*** \_\_init\_\_(): là hàm khởi tạo, tự động được gọi khi đối tượng được tạo từ lớp, được dùng để gán giá trị ban đầu cho các thuộc tính

# Tạo lớp

class Nguoi:

def \_\_init\_\_(self, ten, tuoi):

self.ten = ten # Thuộc tính

self.tuoi = tuoi

def gioi\_thieu(self): # Phương thức

print(f"Tôi tên là {self.ten}, năm nay {self.tuoi} tuổi.")

# Tạo đối tượng

nguoi\_1 = Nguoi("An", 25)

nguoi\_2 = Nguoi("Bình", 30)

# Gọi phương thức

nguoi\_1.gioi\_thieu()

nguoi\_2.gioi\_thieu()​

Kết quả:

Tôi tên là An, năm nay 25 tuổi.

Tôi tên là Bình, năm nay 30 tuổi.

**II. Mô tả chương trình:**

***Hàm:***

- Nhập danh sách khách hàng từ người dùng hoặc thêm danh sách bổ sung.

- Xuất thông tin toàn bộ danh sách khách hàng.

- Tìm kiếm khách hàng theo số điện thoại hoặc tên.

- Lọc danh sách khách hàng theo loại đàn hoặc theo giá đàn.

***Menu:*** Chọn các chức năng để xuất ra màn hình.

***Project:*** [*https://github.com/NguyenDinhQuocHuy/Project\_LTCT/blob/main/deadline.py*](https://github.com/NguyenDinhQuocHuy/Project_LTCT/blob/main/deadline.py)