



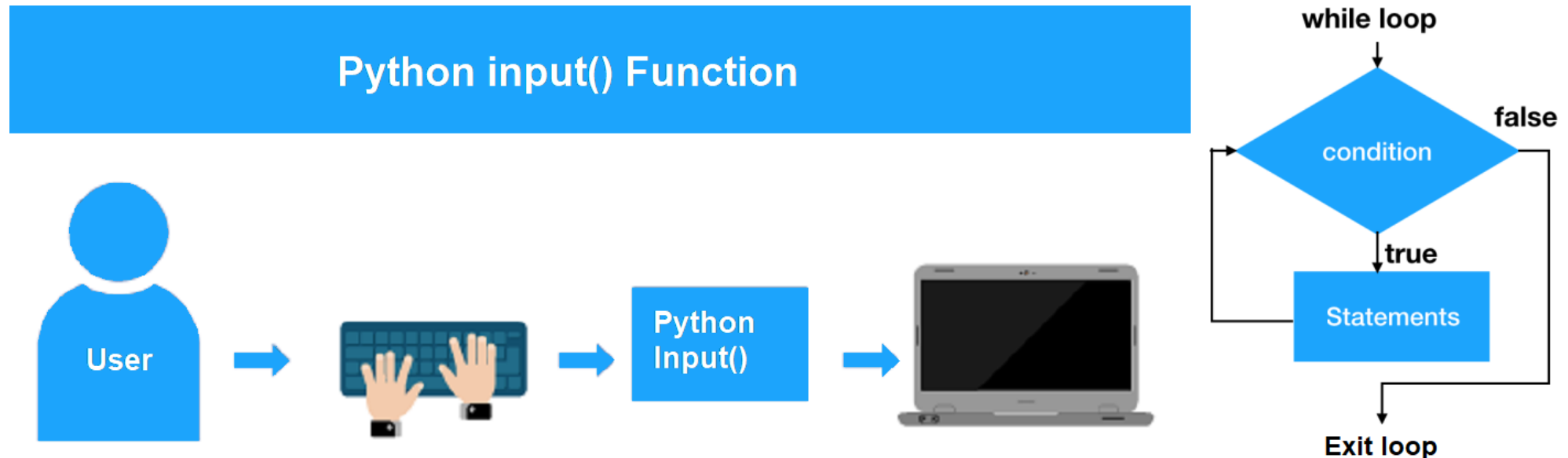
# **CHƯƠNG 6: ĐẦU VÀO NGƯỜI DÙNG VÀ VÒNG LẶP WHILE**

P T Ha  
CNTT1. PTIT

# MỤC TIÊU

---

- ❑ Hiểu được câu lệnh vào ra màn hình, bàn phím, tệp và vòng lặp
- ❑ Lập trình được cho các bài toán sử dụng các câu lệnh vào ra
- ❑ Lập trình vòng lặp đơn giản và phức tạp.



# NỘI DUNG

---

## ❑ Cách hàm input hoạt động:

- Câu lệnh nhập
- Câu lệnh xuất
- Xử lý lỗi nhập liệu

## ❑ Vòng lặp While

- Sử dụng cờ
- Sử dụng break và continue
- Vòng lặp vô hạn

## ❑ Vòng lặp While với danh sách và từ điển

- Vòng lặp While với danh sách
- Nhập dữ liệu vào từ điển bằng vòng lặp While

# CÂU LỆNH NHẬP DỮ LIỆU

- ❑ Câu lệnh nhận dữ liệu từ bàn phím là `input()`, có cú pháp như sau:

*Variable = input("Message")*

- ❑ **Chú ý:**

- Giá trị nhận vào từ bàn phím là giá trị kiểu chuỗi.
- Phần thông báo hiển thị thông tin hướng dẫn cho người dùng
- Sublime Text không hỗ trợ phần nhập dữ liệu nên sử dụng Terminal để chạy các file này

```
1 # Đầu vào dữ liệu kiểu số nguyên
2 age = int(input("Please enter your age: "))
3
4 print(age)
```

```
Please enter your age: 20
20
```

# CÂU LỆNH NHẬP DỮ LIỆU

---

- ❑ Sử dụng terminal để chạy các file với hàm nhập dữ liệu:

```
chapter6.py x
# Đầu vào dữ liệu kiểu số nguyên
age = int(input("Please enter your age: "))
print(age)
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.657]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\HoaiT\Documents\PTIT\python_work>python chapter6.py
Please enter your age: 20
20
```

# CÂU LỆNH NHẬP DỮ LIỆU

---

- ❑ Để sử dụng hàm nhập dữ liệu trên Sublime Text, cần cài package “SublimeREPL”.
- ❑ Để cài đặt package: Preferences => Package Control => Install Package => Nhập tên package cần cài đặt.
- ❑ Để chạy chương trình, chọn: Tools => SublimeREPL => Python => Python-RUN current file

```
# Đầu vào dữ liệu kiểu số nguyên  
age = int(input("Please enter your age: "))  
print(age)
```



```
Please enter your age: 20  
20  
***Repl Closed***|
```

# KIỂU SỐ NGUYÊN VÀ SỐ THỰC

- ❑ Python cho phép không cần khai báo kiểu dữ liệu cho biến. Phụ thuộc vào giá trị của phép gán, Python sẽ chọn kiểu dữ liệu cho biến.

```
1 a = 6
2 b = 9.6
3 print("Kiểu dữ liệu của a và b là: ", type(a), type(b))
```

Kiểu dữ liệu của a và b là: <class 'int'> <class 'float'>

- ❑ Giá trị nhận vào từ bàn phím muốn trả về giá trị kiểu nào phải ép kiểu cho giá trị đó.

```
1 # Đầu vào dữ liệu kiểu số nguyên
2 age = int(input("Please enter your age: "))
3 int_age = int(age)
4 print(f"Kiểu dữ liệu của 'age': {type(age)}, Kiểu dữ liệu của 'int_age': {type(int_age)}")
```

Please enter your age: 20

Kiểu dữ liệu của 'age': <class 'int'>, Kiểu dữ liệu của 'int\_age': <class 'int'>

# XỬ LÝ LỖI NHẬP DỮ LIỆU

---

- ❑ Khi đưa ra thông báo để nhập vào một số , người sử dụng hoàn toàn có thể nhập vào một chuỗi . Do đó , câu lệnh chuyển đổi kiểu có thể hoạt động sai và chương trình phải dừng .
- ❑ Sử dụng câu lệnh try-except cho trường hợp người dùng nhập giá trị số không phù hợp

```
1 try:
2     a = int(input("Nhập vào số nguyên a: "))
3 except:
4     a = 0
5
6 print(a)
```

Nhập vào số nguyên a: PTIT2021  
0



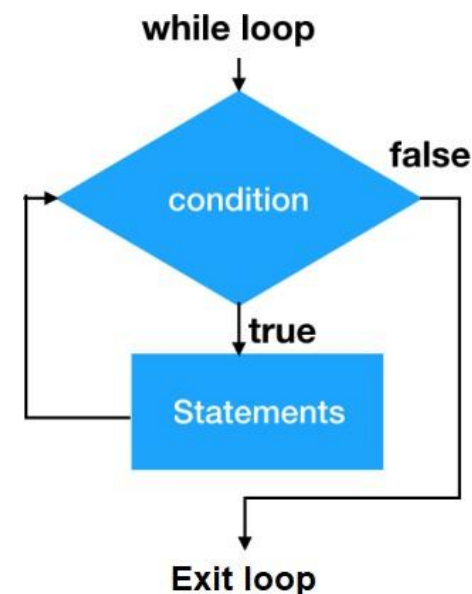
# BÀI TẬP

- ❑ Bài tập 1: Viết chương trình hỏi người dùng số lượng người muốn đặt bàn. Nếu số lượng người lớn hơn 8, hiển thị thông báo vui lòng chờ khi có bàn. Ngược lại, hiển thị bàn của họ đã sẵn sàng.
- ❑ Bài tập 2: Nhập vào một số bất kỳ. Hiển thị thông báo số đó có phải bội số của 10 hay không.

# VÒNG LẶP WHILE

- ❑ Vòng lặp cho phép lặp lại đoạn code cho tới khi một điều kiện nhất định được thỏa mãn . Python chỉ hỗ trợ vòng lặp for và vòng lặp while .

*while conditional:  
statements*



- ❑ Trong đó :
  - **conditonal**: có thể là một biến hoặc một biểu thức có giá trị là True hoặc False .
  - **statements**: là tập lệnh thực thi được xác định thông qua thụt lề , nếu chỉ có một lệnh thì không cần xuống dòng .

# SỬ DỤNG CỜ

---

- ❑ Đối với một vòng lặp có nhiều điều kiện kiểm tra , có thể sử dụng một biến xác định toàn bộ các trạng thái . Biến này , được gọi là cờ ( Flag )
- ❑ Vòng lặp thực thi khi cờ có giá trị True và dừng chạy khi bất kỳ điều kiện nào đặt giá trị của cờ thành False . Do đó , câu lệnh while chỉ cần kiểm tra một điều kiện giá trị Flag là True hay False .

# SỬ DỤNG CỜ

---

- ❑ Ví dụ : Để dừng chương trình nhập ' Q ' hoặc ' quit “

```
prompt = "\n Tell me something, and I will repeat it back to you:"  
prompt += "\n Enter 'quit' or 'Q' to end the program"  
  
active = True  
while active:  
    message = input(prompt)  
    if message == 'quit' or message=="Q":  
        active = False  
    else:  
        print(message)
```



```
Tell me something, and I will repeat it back to you:  
Enter 'quit' or 'Q' to end the program. hello class  
hello class
```

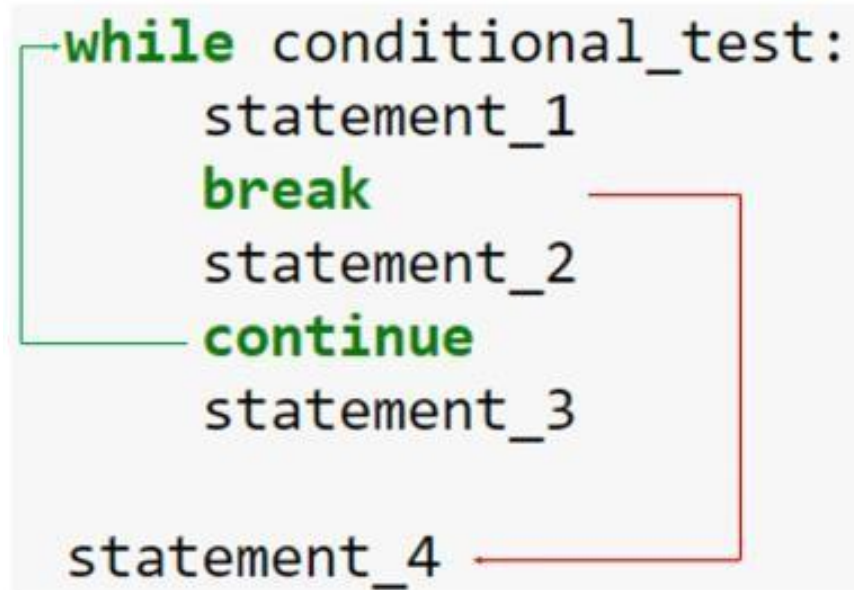
```
Tell me something, and I will repeat it back to you:  
Enter 'quit' or 'Q' to end the program. quit
```

# SỬ DỤNG BREAK VÀ CONTINUE

---

- ❑ Để kết thúc vòng lặp, ta có thể sử dụng câu lệnh break. Câu lệnh break nằm trong vòng lặp của khối lệnh nào thì vòng lặp của khối lệnh đó sẽ kết thúc.
- ❑ Câu lệnh continue bỏ qua những lệnh ở dưới continue và tiếp tục vòng lặp.

```
while conditional_test:  
    statement_1  
    break  
    statement_2  
    continue  
    statement_3  
statement_4
```



# VÒNG LẶP VÔ HẠN

- ❑ Xem xét một ví dụ về vòng lặp vô hạn sau:

```
1 total = 1
2 while total < 6:
3     print(total)
4     total += 1
```

1  
2  
3  
4  
5

```
1 total = 1
2 while total < 6:
3     print(total)
4     # total += 1
```

1  
1  
1  
1  
1

-----  
KeyboardInterrupt

- ❑ Nếu chương trình trong một vòng lặp vô hạn, hãy nhấn Ctrl-C để đóng cửa sổ chương trình để thoát khỏi vòng lặp vô hạn.
- ❑ Để tránh viết các vòng lặp vô hạn, hãy kiểm tra vòng lặp while đảm bảo vòng lặp dừng lại tại giá trị mong muốn.

# VÒNG LẶP WHILE VỚI DANH SÁCH

- ❑ Vòng lặp while cho phép lưu trữ, thay đổi và sắp xếp các phần tử trong danh sách và từ điển.
- ❑ Bài tập ví dụ: Có một danh sách người dùng đăng ký mới nhưng chưa được xác minh. Sau khi xác minh người dùng này, họ sẽ được chuyển sang một danh sách khác gồm những người dùng đã được xác thực.

```
1 unconfirmed_users = ['alice', 'brian', 'paul']
2 confirmed_users = []
3
4 while unconfirmed_users:
5     current_user = unconfirmed_users.pop()
6
7     print(f"Verifying user: {current_user.title()}")
8     confirmed_users.append(current_user)
9
10 print("\nThe following users have been confirmed:")
11 for confirmed_user in confirmed_users:
12     print(confirmed_user.title())
```

Verifying user: Paul

Verifying user: Brian

Verifying user: Alice

The following users have been confirmed:

Paul

Brian

Alice

# LOẠI BỎ CÁC PHẦN TỬ TRONG MỘT DANH SÁCH CỤ THỂ

- ❑ Phương thức `remove()` dùng để loại bỏ một phần tử cụ thể trong một danh sách. Để loại bỏ tất cả các phần tử có cùng một giá trị sử dụng `while` kết hợp với phương thức `remove()`.
- ❑ Ví dụ: Loại bỏ hết các giá trị “cat” trong danh sách.

```
1 pets = ['dog', 'cat', 'goldfish', 'cat', 'rabbit', 'cat']
2 print(pets)
3
4 while "cat" in pets:
5     pets.remove("cat")
6
7 print(pets)
```

```
['dog', 'cat', 'goldfish', 'cat', 'rabbit', 'cat']
['dog', 'goldfish', 'rabbit']
```



# NHẬP DỮ LIỆU VÀO TỪ ĐIỂN

- ❑ Ví dụ: Viết một chương trình thăm dò ý kiến về môn học yêu thích, trong đó mỗi người sẽ được lấy tên và ý kiến phản hồi của họ. Lưu trữ dữ liệu thu thập được trong một từ điển.

```
responses = {}  
#Tạo flag để biết cuộc khảo sát đang hoạt động  
polling_active = True  
  
while polling_active:  
    name = input("Vui lòng nhập tên: ")  
    response = input("Môn học yêu thích của bạn: ")  
  
    #Lưu tên và câu trả lời vào từ điển  
    responses[name] = response  
  
    repeat = input("Bạn có muốn gửi câu trả lời không? (yes/no) ")  
  
    if repeat == 'no':  
        polling_active = False  
  
print("\n--Kết quả khảo sát:--")  
for name, response in responses.items():  
    print(f"{name} thích môn học {response}")
```

```
Vui lòng nhập tên: Nam  
Môn học yêu thích của bạn: Xử lý ảnh  
Bạn có muốn gửi câu trả lời không? (yes/no) yes  
Vui lòng nhập tên: Khánh  
Môn học yêu thích của bạn: Toán rời rạc  
Bạn có muốn gửi câu trả lời không? (yes/no) yes  
Vui lòng nhập tên: Hùng  
Môn học yêu thích của bạn: Giải tích  
Bạn có muốn gửi câu trả lời không? (yes/no) no  
  
--Kết quả khảo sát:--  
Nam thích môn học Xử lý ảnh  
Khánh thích môn học Toán rời rạc  
Hùng thích môn học Giải tích
```

# BÀI TẬP

- ❑ Bài tập 3: Sử dụng vòng lặp while lặp qua một danh sách, nếu có số 100, hãy in ra số đó với chỉ mục của nó.
- ❑ Bài tập 4: Tạo một danh sách bất kỳ bằng cách sử dụng hàm input(). Sử dụng vòng lặp while, viết chương trình thêm tất cả các phần tử của danh sách vừa nhập vào danh sách mới, ngoại trừ phần tử là một chuỗi rỗng.

Ví dụ: `lst1 = ["Joe", "Sarah", "Mike", "Jess", "", "Matt", "", "Greg"]`

Output: `new_list = ["Joe", "Sarah", "Mike", "Jess", "Matt", "Greg"]`

# BÀI TẬP

- ❑ Bài tập 5: Nhập vào một số nguyên N (nhỏ hơn 100). Viết chương trình lấy ra chuỗi Fibonacci từ 0 đến N. Lưu kết quả vào một mảng

*Ví dụ N=30*

*Output: [0, 1, 1, 2, 3, 5, 8, 13, 21]*

- ❑ Bài tập 6: Nhập vào một chuỗi các số dưới dạng nhị phân có 4 chữ số và được phân tách bởi dấu phẩy. Sử dụng vòng lặp while, chuyển các số nhị phân về dạng thập phân và in ra các số chia hết cho 5.

*Ví dụ: Input: 0100,0011,1010,1001,1100,1001*

*Output: 1010*

# KẾT CHƯƠNG

- ❑ Sử dụng lệnh `input()` để nhập dữ liệu từ bàn phím
- ❑ Vòng lặp `while` sử dụng cờ, `break`, `continue`
- ❑ Vòng lặp `while` với danh sách và từ điển