



# Lập trình với Python

## Chương 3: Cấu trúc điều khiển

# Tài liệu tham khảo

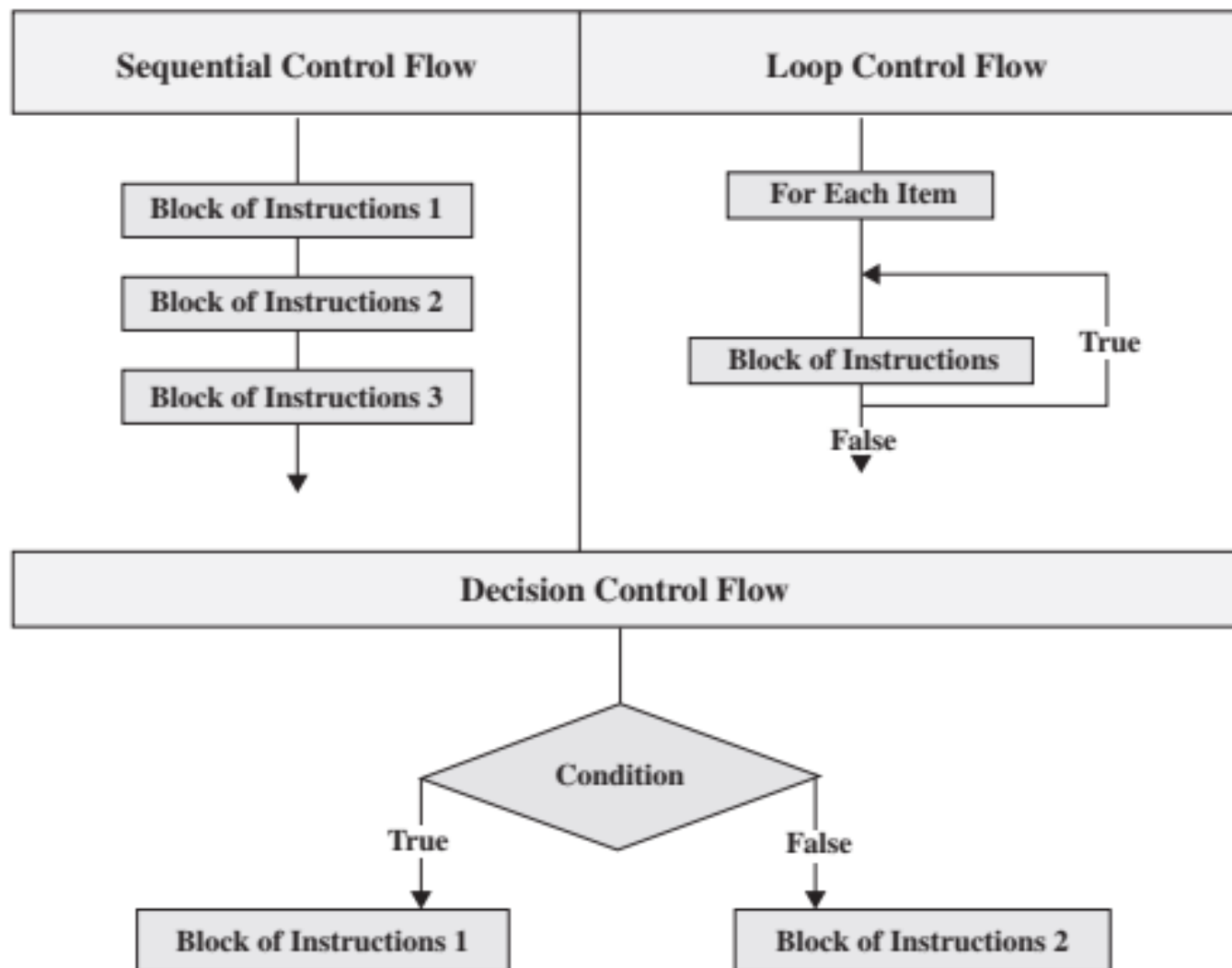
1. Hill, C. (2020). Learning scientific programming with Python. Cambridge University Press.
2. Stephenson, B. (2019). The Python Workbook 2<sup>nd</sup>. Springer.
3. Pine, D. J. (2019). Introduction to Python for science and engineering. CRC Press.
4. Kong, Q., Siau, T., & Bayen, A. M. (2021) Python Programming and Numerical Methods - A Guide for Engineers and Scientists. Elsevier Inc.

# Cấu trúc điều khiển

- Cấu trúc điều khiển tuần tự: Điều này đề cập đến việc thực thi từng dòng một, các câu lệnh được thực thi tuần tự, theo cùng một thứ tự mà chúng xuất hiện trong chương trình.
- Cấu trúc kiểm soát quyết định: Tùy thuộc vào điều kiện đúng hay sai, cấu trúc quyết định có thể bỏ qua việc thực thi toàn bộ khối các câu lệnh hoặc thậm chí thực thi một khối câu lệnh thay vì các câu lệnh khác.

# Cấu trúc điều khiển

- Cấu trúc điều khiển vòng lặp: Đây là cấu trúc điều khiển cho phép thực hiện một khối câu lệnh nhiều lần cho đến khi điều kiện kết thúc vòng lặp được đáp ứng (vòng lặp for và vòng lặp while). Các cấu trúc điều khiển vòng lặp còn được gọi là câu lệnh lặp lại.



# Cấu trúc điều khiển if

- Cú pháp cho câu lệnh if:

Keyword

*if Boolean\_Expression:*  
*statement (s)*

Colon should be  
present at the end

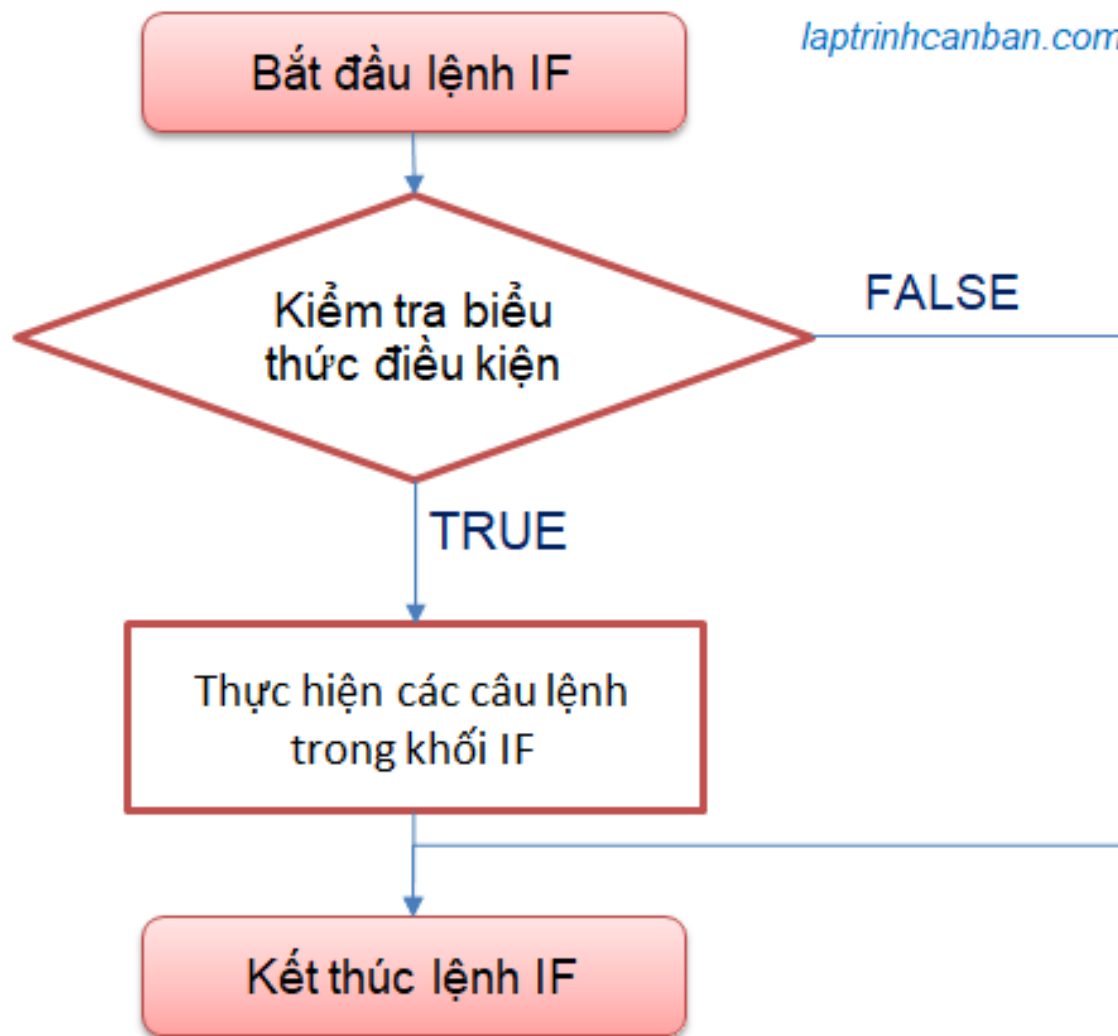
Indentation

1. >>> if 20 > 10:

2. ... print(f"20 is greater than 10")

**Output**

20 is greater than 10



# Cấu trúc điều khiển if

- Câu lệnh if bắt đầu bằng từ khóa if và kết thúc bằng dấu hai chấm. Các biểu thức trong câu lệnh if phải là biểu thức Boolean. Nếu biểu thức Boolean là True thì các câu lệnh trong khối if sẽ được thực thi; nếu là False thì không câu lệnh nào được thực thi.
- Trong Python, các câu lệnh khối if được xác định thông qua thụt đầu dòng.



# Cấu trúc điều khiển if

1. `number = int(input("Enter a number"))`
2. `if number >= 0:`
3. `print(f"The number entered by the user is a positive number")`

## OUTPUT

Enter a number 67

The number entered by the user is a positive number

# Cấu trúc điều khiển if

1. `weight = int(input("How many pounds does your suitcase weigh?"))`
2. `if weight > 50:`
3.     `print(f"There is a $25 surcharge for heavy luggage")`
4.     `print(f"Thank you")`

## OUTPUT

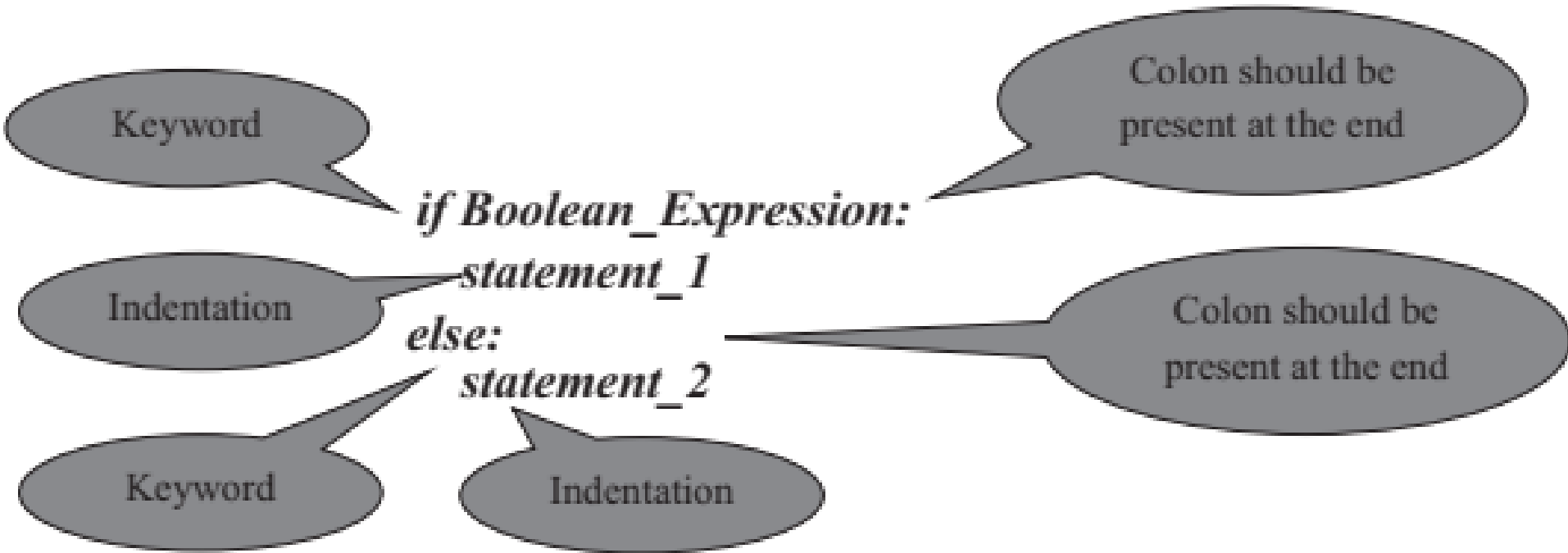
```
How many pounds does your suitcase weigh? 75
There is a $25 surcharge for heavy luggage
Thank you
```

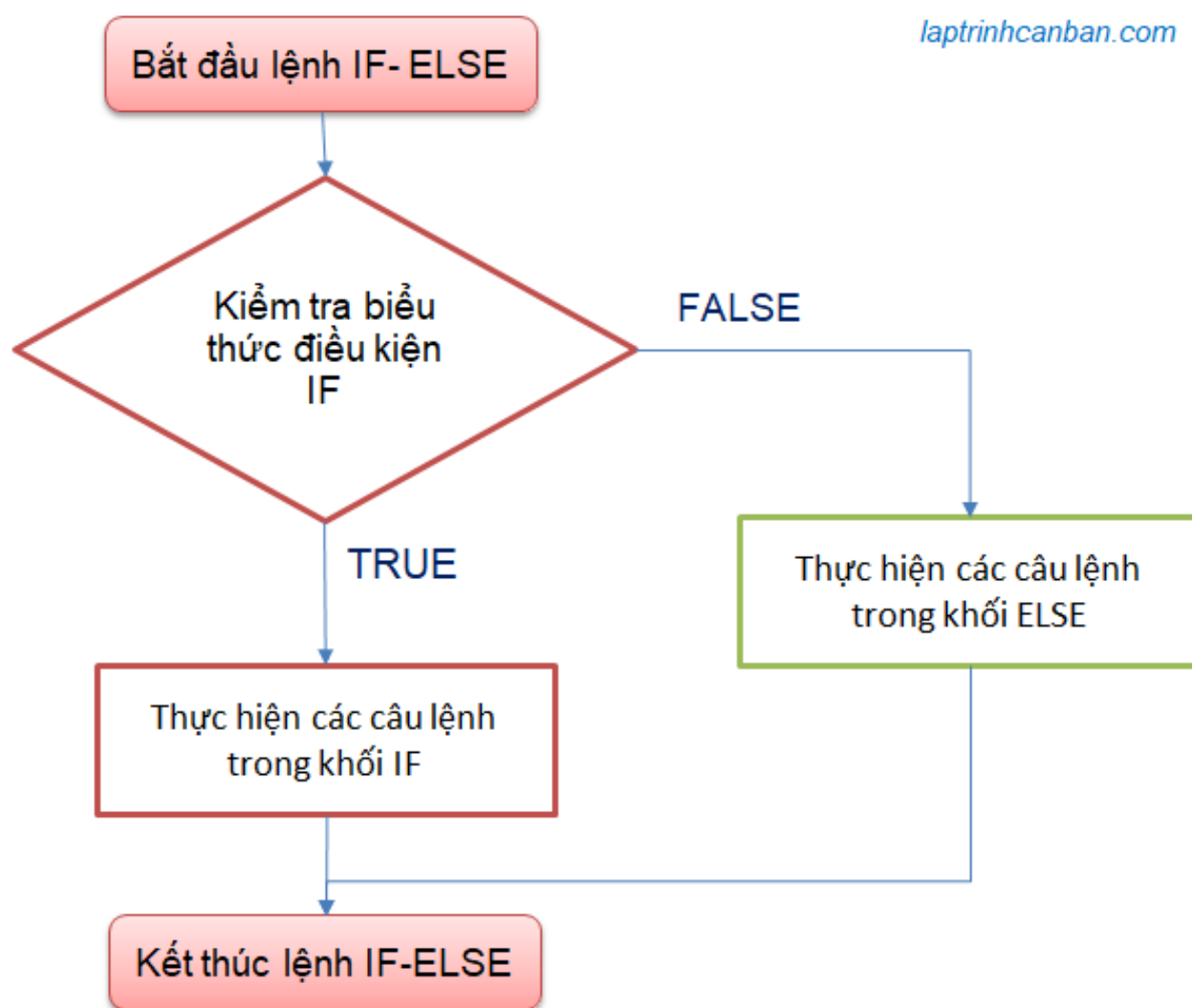
# Cấu trúc điều khiển if...else

- Một câu lệnh if cũng có thể được theo sau bởi một câu lệnh else tùy chọn.
- Các câu lệnh trong khối if được thực thi nếu Boolean\_Expression là đúng.
- Sử dụng khối else để thực thi các câu lệnh nếu Boolean\_Expression là sai.
- Câu lệnh if... else cho phép đưa ra quyết định hai chiều.

# Cấu trúc điều khiển if...else

- Cú pháp cho câu lệnh if...else:





# Cấu trúc điều khiển if...else

- Nếu Boolean\_Expression được đánh giá là True, thì câu lệnh\_1 được thực thi. Ngược lại, nó được đánh giá là False thì câu lệnh\_2 được thực thi.
- Thụt lề được sử dụng để tách các khối. Sau thực hiện câu lệnh\_1 hoặc câu lệnh\_2, điều khiển được chuyển sang câu lệnh tiếp theo sau câu lệnh if.
- Ngoài ra, các từ khóa if và else phải được căn chỉnh ở cùng một vị trí cột.

# Cấu trúc điều khiển if...else

1. `number = int(input("Enter a number"))`
2. `if number % 2 == 0:`
3.     `print(f'{number} is Even number')`
4. `else:`
5.     `print(f'{number} is Odd number')`

## OUTPUT

Enter a number: 45

Lập trình Py 45 is Odd number

# Cấu trúc điều khiển if...else

1. `number_1 = int(input("Enter the first number"))`
2. `number_2 = int(input("Enter the second number"))`
3. `if number_1 > number_2:`
4.     `print(f'{number_1} is greater than {number_2}')`
5. `else:`
6.     `print(f'{number_2} is greater than {number_1}')`

## OUTPUT

Enter the first number 8

Enter the second number 10

10 is greater than 8

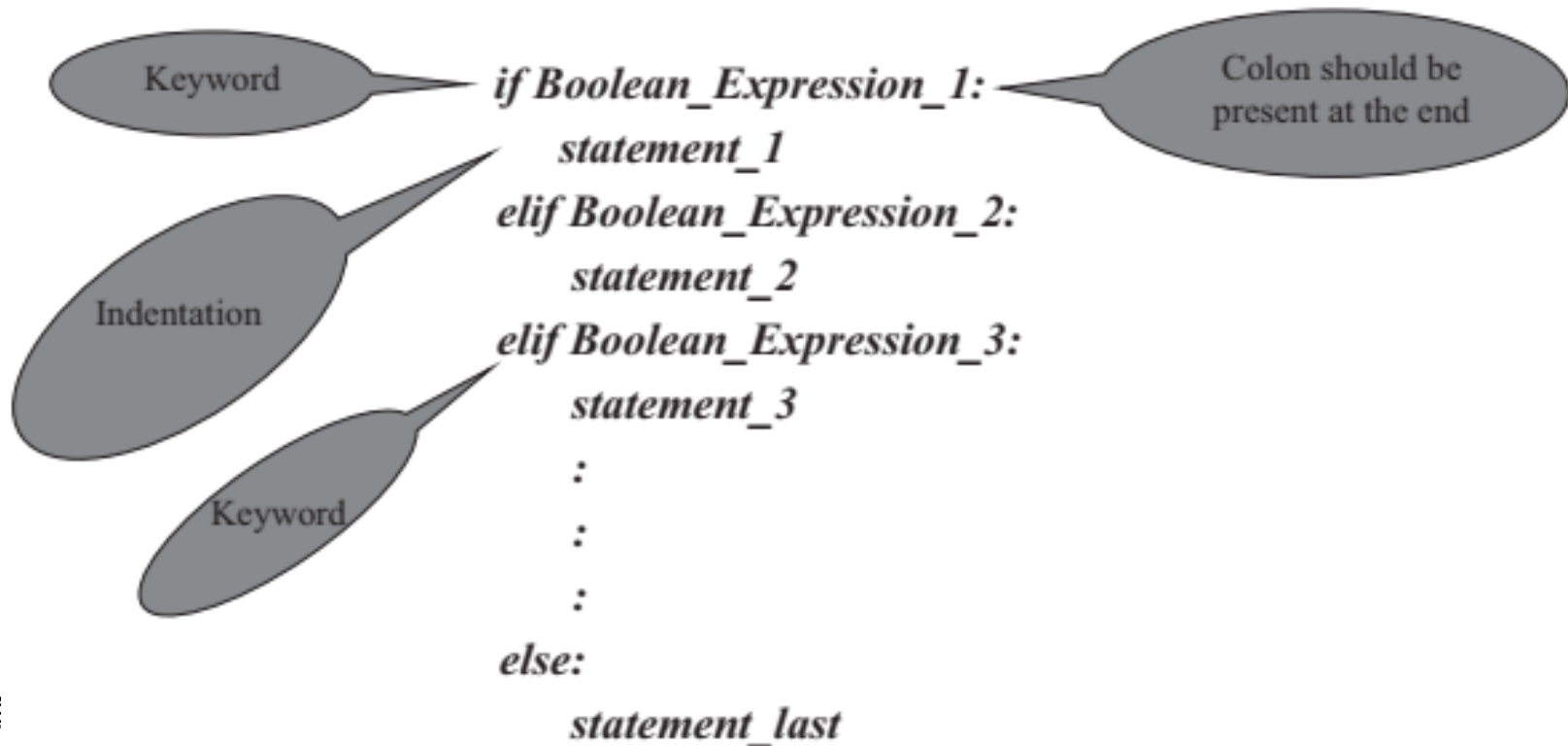


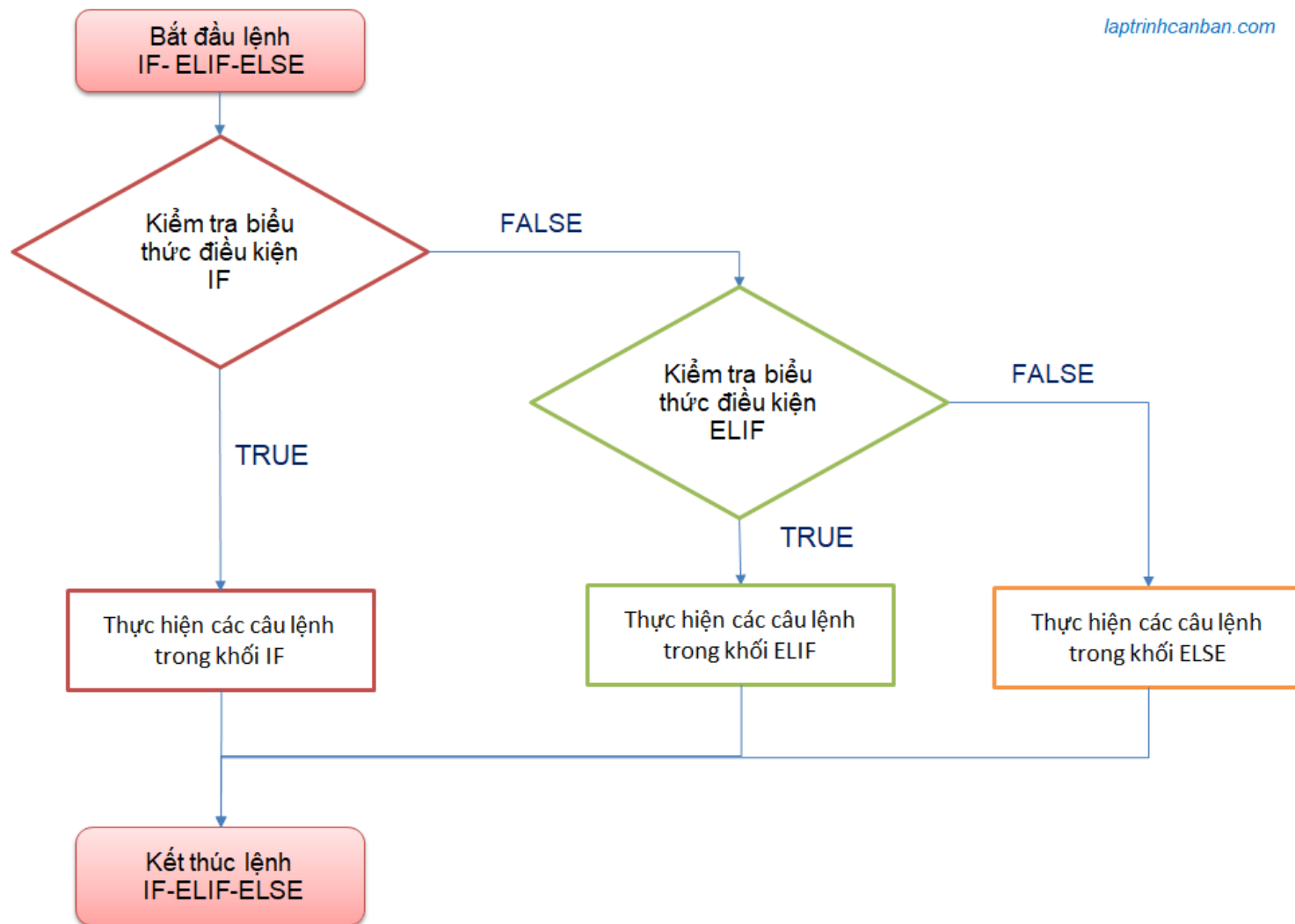
# Cấu trúc điều khiển if...elif...else

- if... elif... else còn được gọi là câu lệnh kiểm tra quyết định đa chiều. Khi chúng ta cần chọn từ một số lựa chọn thay thế khả thi, câu lệnh elif được sử dụng cùng với if.
- Từ khóa “elif” là viết tắt của “else if” và rất hữu ích để tránh thụt lề quá mức.
- Câu lệnh else phải luôn đứng sau cùng.

# Cấu trúc điều khiển if...elif...else

- Cú pháp cho câu lệnh if...elif...else:





## Cấu trúc

```
1. score = float(input("Enter your score"))
2. if score < 0 or score > 1:
3.     print("Wrong Input")
4. elif score >= 0.9:
5.     print("Your Grade is "A" ")
6. elif score >= 0.8:
7.     print("Your Grade is "B" ")
8. elif score >= 0.7:
9.     print("Your Grade is "C" ")
10. elif score >= 0.6:
11.     print("Your Grade is "D" ")
12. else:
13.     print("Your Grade is "F" ")
```

.else

### OUTPUT

```
Enter your score0.92
Your Grade is "A"
```

Cá

```
1. fruit_type = input("Enter the Fruit Type:")
2. if fruit_type == "Oranges":
3.     print('Oranges are $0.59 a pound')
4. elif fruit_type == "Apples":
5.     print('Apples are $0.32 a pound')
6. elif fruit_type == "Bananas":
7.     print('Bananas are $0.48 a pound')
8. elif fruit_type == "Cherries":
9.     print('Cherries are $3.00 a pound')
10. else:
11.     print(f'Sorry, we are out of {fruit_type}')
```

### OUTPUT

Enter the Fruit Type: Cherries

Cherries are \$3.00 a pound

# Cấu trúc điều khiển if lồng nhau

- Trong một số tình huống, bạn phải đặt câu lệnh if bên trong một câu lệnh khác.
- Một câu lệnh if chứa một câu lệnh if khác trong khối if của nó hoặc khối else được gọi là một khối câu lệnh if lồng nhau.

# Cấu trúc điều khiển if lồng nhau

- Cú pháp cho câu lệnh if lồng nhau:



Nested if

```
if Boolean_Expression_1:  
    if Boolean_Expression_2:  
        statement_1  
    else:  
        statement_2  
else:  
    statement_3
```

```
1. year = int(input('Enter a year'))
2. if year % 4 == 0:
3.     if year % 100 == 0:
4.         if year % 400 == 0:
5.             print(f'{year} is a Leap Year')
6.         else:
7.             print(f'{year} is not a Leap Year')
8.     else:
9.         print(f'{year} is a Leap Year')
10. else:
11.     print(f'{year} is not a Leap Year')
```

### OUTPUT

```
Enter a year 2014
2014 is not a Leap Year
```



# Cấu trúc vòng while

- Cú pháp cho câu lệnh while:

Keyword

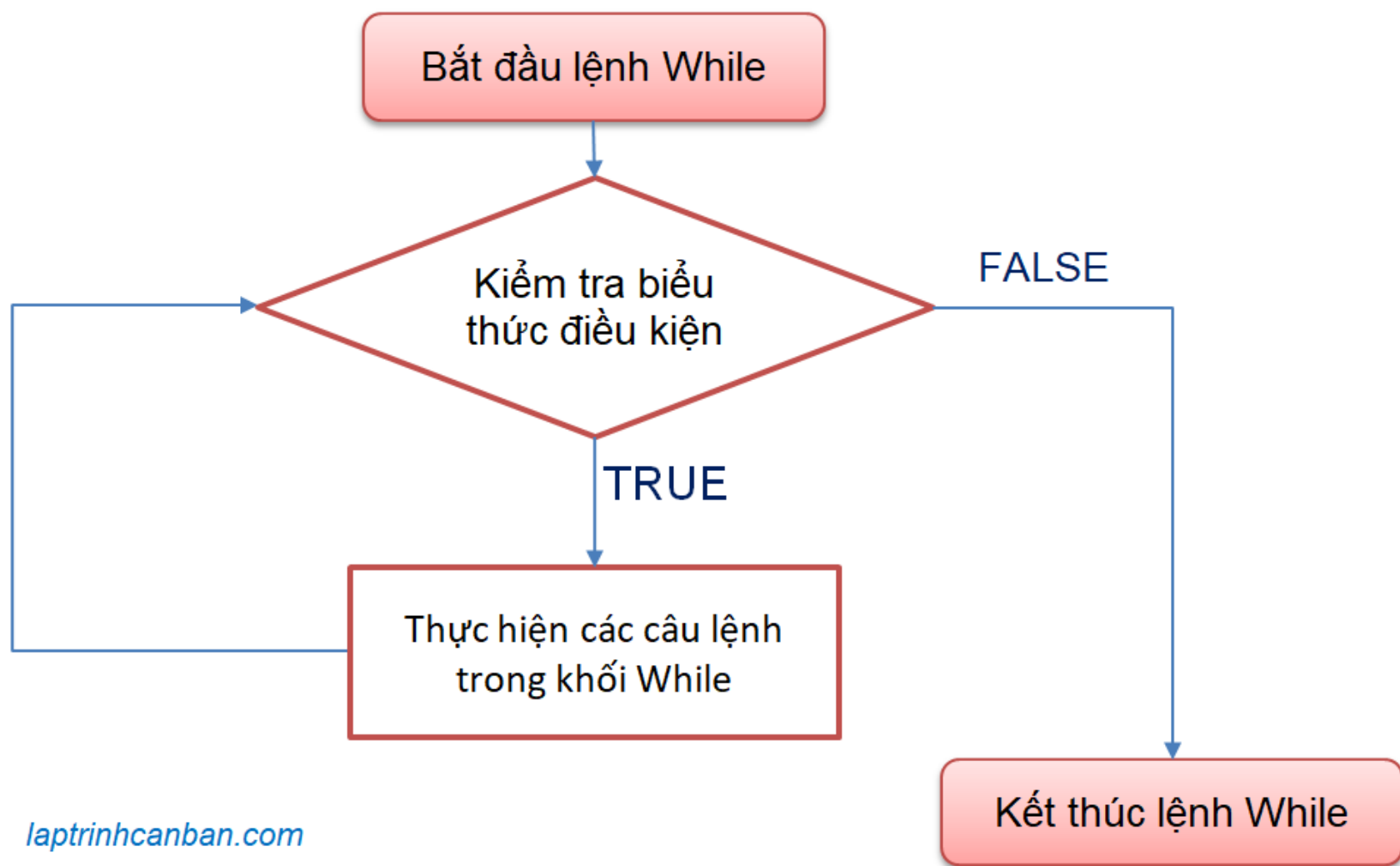
*while Boolean\_Expression:*

Colon should be present at the end

Indentation

*statement(s)*

- Vòng lặp while bắt đầu bằng từ khóa while và kết thúc bằng dấu hai chấm.



# Cấu trúc vòng while

- Điều đầu tiên xảy ra là biểu thức Boolean được đánh giá trước các câu lệnh trong khối lặp while được thực thi.
- Nếu biểu thức Boolean đánh giá là False, thì các câu lệnh trong khối lặp while không bao giờ được thực thi.
- Nếu biểu thức Boolean đánh giá là True, thì các câu lệnh trong khối vòng lặp while được thực thi.
- Sau mỗi lần lặp lại của vòng lặp, biểu thức Boolean được kiểm tra một lần nữa và nếu nó là True, vòng lặp sẽ được lặp lại một lần nữa.

# Cấu trúc vòng while

- Mỗi lần lặp lại khối lặp được gọi là một lần lặp của vòng lặp. Quá trình này tiếp tục cho đến khi biểu thức Boolean đánh giá là False và tại thời điểm này, câu lệnh while sẽ thoát ra.
- Sau đó, việc thực thi tiếp tục với câu lệnh đầu tiên sau vòng lặp while.

1. `i = 0`
2. `while i < 10:`
3.     `print(f'Current value of i is {i}')`
4.     `i = i + 1`

### OUTPUT

Current value of i is 0

Current value of i is 1

Current value of i is 2

Current value of i is 3

Current value of i is 4

Current value of i is 5

Current value of i is 6

Current value of i is 7

Current value of i is 8

Current value of i is 9

```
1. number = int(input("Enter a number up to which you want to find the average"))
2. i = 0
3. sum = 0
4. count = 0
5. while i < number:
6.     i = i + 1
7.     sum = sum + i
8.     count = count + 1
9. average = sum/count
10. print(f"The average of {number} natural numbers is {average}")
```

## OUTPUT

```
Enter a number up to which you want to find the average 5
The average of 5 natural numbers is 3.0
```

```
1. m = int(input("Enter first positive number"))
2. n = int(input("Enter second positive number"))
3. if m == 0 and n == 0:
4.     print("Invalid Input")
5. if m == 0:
6.     print(f"GCD is {n}")
7. if n == 0:
8.     print(f"GCD is {m}")
9. while m != n:
10.     if m > n:
11.         m = m - n
12.     if n > m:
13.         n = n - m
14. print(f"GCD of two numbers is {m}")
```

#### OUTPUT

```
Enter first positive number8
Enter second positive number12
GCD of two numbers is 4
```

# Cấu trúc vòng while

1. `number = int(input('Enter a number'))`
2. `result = 0`
3. `remainder = 0`
4. `while number != 0:`
5.     `remainder = number % 10`
6.     `result = result + remainder`
7.     `number = int(number / 10)`
8. `print(f"The sum of all digits is {result}")`

## OUTPUT

Enter a number1234

The sum of all digits is 10



# Cấu trúc vòng for

- Cú pháp cho câu lệnh for:

Keyword

*for iteration\_variable in sequence:*

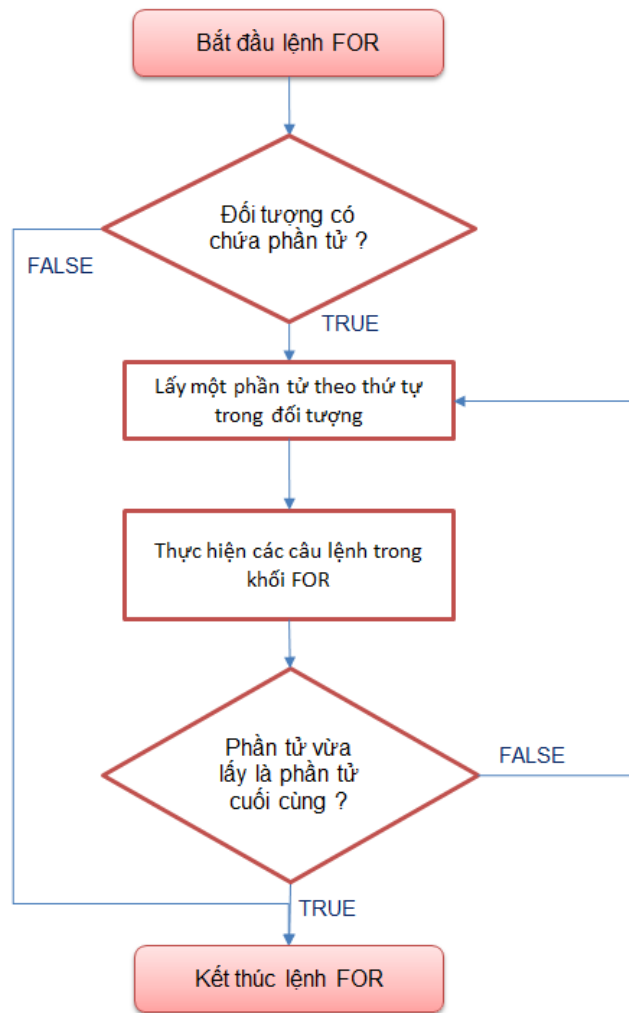
Colon should be present at the end

Indentation

*statement(s)*

Keyword

# Cấu



# Cấu trúc vòng for

- Vòng lặp for bắt đầu bằng từ khóa for và kết thúc bằng dấu hai chấm.
- Ở đây, `iteration_variable` có thể là bất kỳ tên biến hợp lệ. Sau đó, khối câu lệnh được thực thi và việc thực thi câu lệnh tiếp tục cho đến khi tất cả các mục trong trình tự được hoàn thành.
- Hàm `range()` được tích hợp sẵn hữu ích trong vòng lặp for.

# Cấu trúc vòng for

- Hàm range() tạo ra một chuỗi các số có thể được lặp lại thông qua việc sử dụng vòng lặp for.
- Cú pháp cho hàm range():

*range([start ,] stop [, step])*

# Cấu trúc vòng for

1. `print("Only "stop" argument value specified in range function")`
2. `for i in range(3):`
3.     `print(f"{i}")`
4. `print("Both "start" and "stop" argument values specified in range function")`
5. `for i in range(2, 5):`
6.     `print(f"{i}")`
7. `print("All three arguments "start", "stop" and "step" specified in range function")`
8. `for i in range(1, 6, 3):`
9.     `print(f"{i}")`

# Cấu trúc vòng for

## OUTPUT

Only "stop" argument value specified in range function

0

1

2

Both "start" and "stop" argument values specified in range function

2

3

4

All three arguments "start", "stop" and "step" specified in range function

1

4

# Cấu trúc vòng for

1. for each\_character in "Blue":
2.     print(f"Iterate through character {each\_character} in the string 'Blue'")

## OUTPUT

Iterate through character B in the string 'Blue'  
Iterate through character l in the string 'Blue'  
Iterate through character u in the string 'Blue'  
Iterate through character e in the string 'Blue'

# Cấu trúc vòng for

```
1. number = int(input("Enter a number"))
2. even = 0
3. odd = 0
4. for i in range(number):
5.     if i % 2 == 0:
6.         even = even + i
7.     else:
8.         odd = odd + i
9. print(f"Sum of Even numbers are {even} and Odd numbers are {odd}")
```

## OUTPUT

Enter a number 10

Sum of Even numbers are 20 and Odd numbers are 25



```
1. number = int(input('Enter a number'))
2. factorial = 1
3. if number < 0:
4.     print("Factorial doesn't exist for negative numbers")
5. elif number == 0:
6.     print('The factorial of 0 is 1')
7. else:
8.     for i in range(1, number + 1):
9.         factorial = factorial * i
10. print(f"The factorial of number {number} is {factorial}")
```

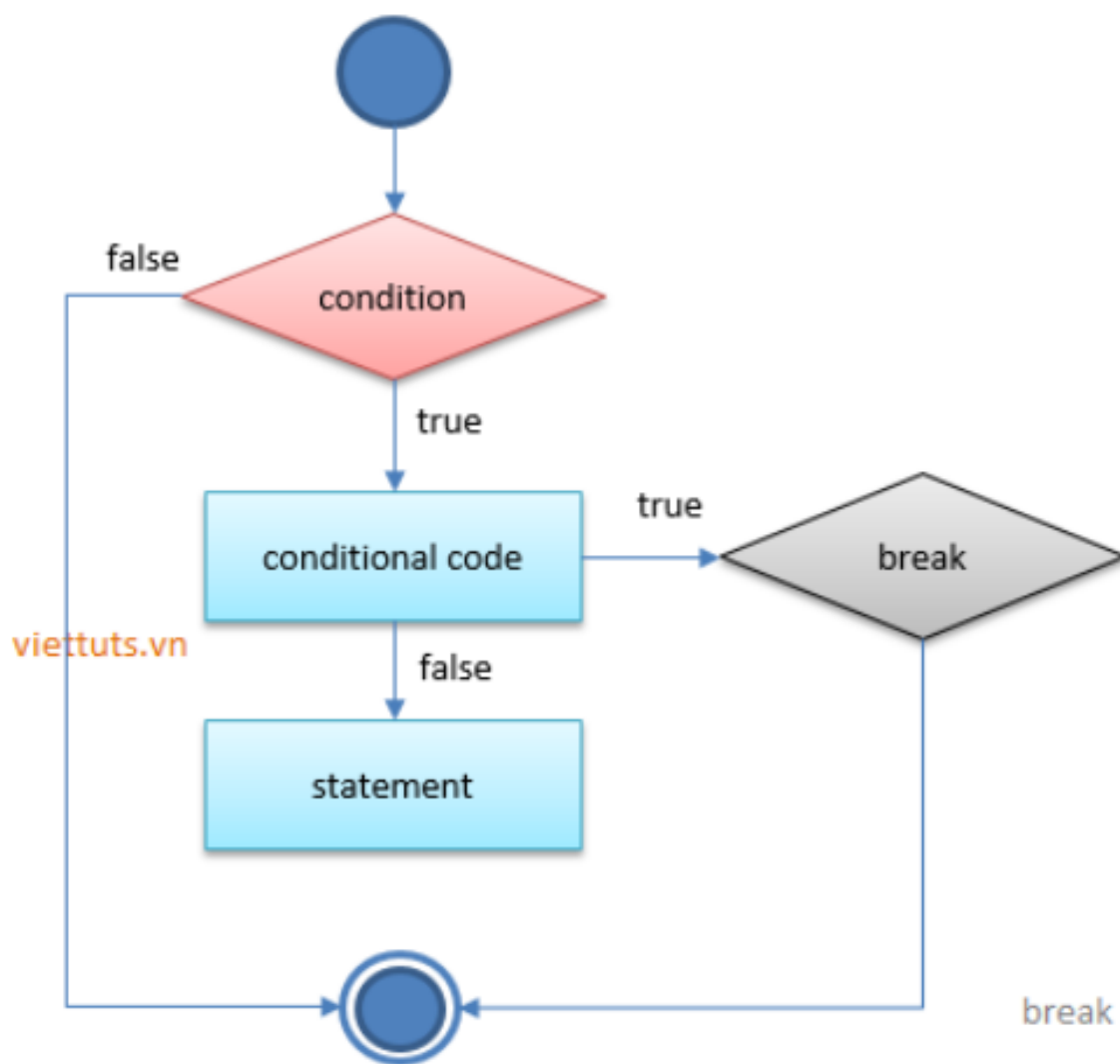
## OUTPUT

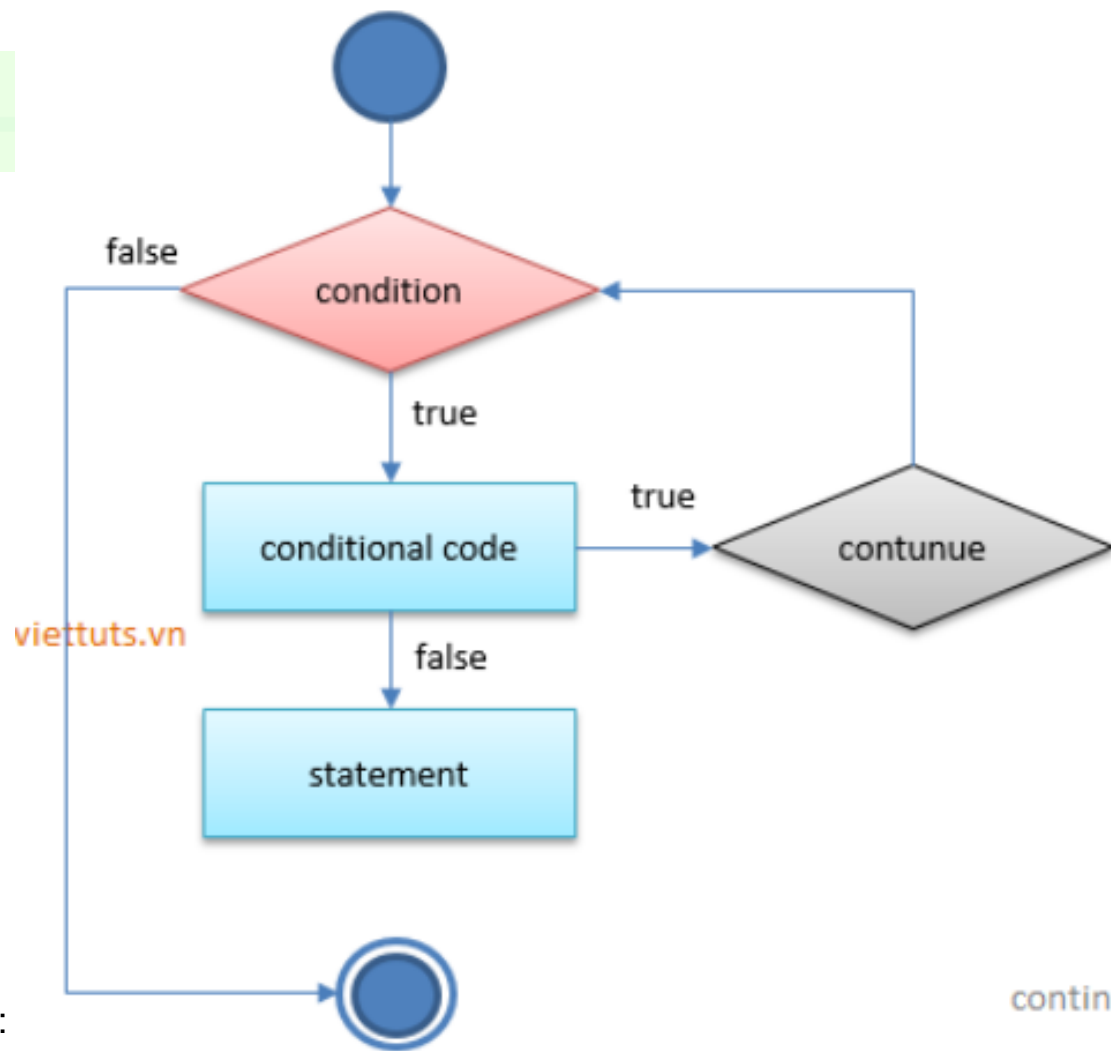
Enter a number 5

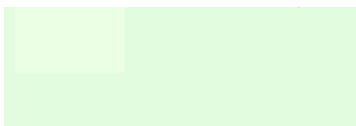
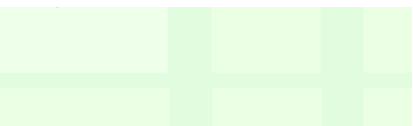
Lập trình The factorial of number 5 is 120

# Lệnh continue và break

- Câu lệnh break và continue cung cấp khả năng kiểm soát tốt hơn đối với việc thực thi mã trong một vòng lặp.
- Bất cứ khi nào gặp câu lệnh break, điều khiển thực thi ngay lập tức chuyển đến lệnh đầu tiên sau vòng lặp.
- Để chuyển quyền điều khiển cho lần lặp tiếp theo mà không thoát khỏi vòng lặp, hãy sử dụng câu lệnh continue.
- Cả hai tiếp tục và không phá vỡ câu lệnh có thể được sử dụng trong vòng lặp while và for.







1. `n = 0`
2. `while True:`
3.     `print(f"The latest value of n is {n}")`
4.     `n = n + 1`
5.     `if n > 5:`
6.         `print(f"The value of n is greater than 5")`
7.         `break`

### OUTPUT

The latest value of n is 0

The latest value of n is 1

The latest value of n is 2

The latest value of n is 3

The latest value of n is 4

The latest value of n is 5

The value of n is greater than 5

```
1. number = int(input('Enter a number > 1: '))
2. prime = True
3. for i in range(2, number):
4.     if number % i == 0:
5.         prime = False
6.         break
7. if prime:
8.     print(f'{number} is a prime number')
9. else:
10.    print(f'{number} is not a prime number')
```

## OUTPUT

Enter a number > 1: 7

7 is a prime number

# Lệnh continue và break

1. `n = 10`
2. `while n > 0:`
3.     `print(f"The current value of number is {n}")`
4.     `if n == 5:`
5.         `print(f"Breaking at {n}")`
6.     `n = 10`
7.     `continue`
8.     `n = n - 1`

# Lệnh continue và break

## OUTPUT

The current value of number is 10

The current value of number is 9

The current value of number is 8

The current value of number is 7

The current value of number is 6

The current value of number is 5

Breaking at 5

The current value of number is 10

The current value of number is 9

The current value of number is 8

The current value of number is 7

The current value of number is 6

The current value of number is 5



# Lệnh try và except

- Xử lý ngoại lệ là một trong những tính năng quan trọng nhất của ngôn ngữ lập trình Python cho phép xử lý các lỗi do ngoại lệ gây ra.
- Ngay cả khi một câu lệnh hoặc biểu thức là cú pháp chính xác, nó có thể gây ra lỗi khi cố gắng thực thi nó. Lỗi được phát hiện trong quá trình thực thi được gọi là ngoại lệ.

# Lệnh try và except

1. `while True`
2. `print("Hello World")`

## OUTPUT

File "<ipython-input-3-c231969faf4f>", line 1

`while True`

^

SyntaxError: invalid syntax

1. `>>> 10 * (1/0)`

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

ZeroDivisionError: division by zero

2. `>>> 4 + spam*3`

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'spam' is not defined

3. `>>> '2' + 2`

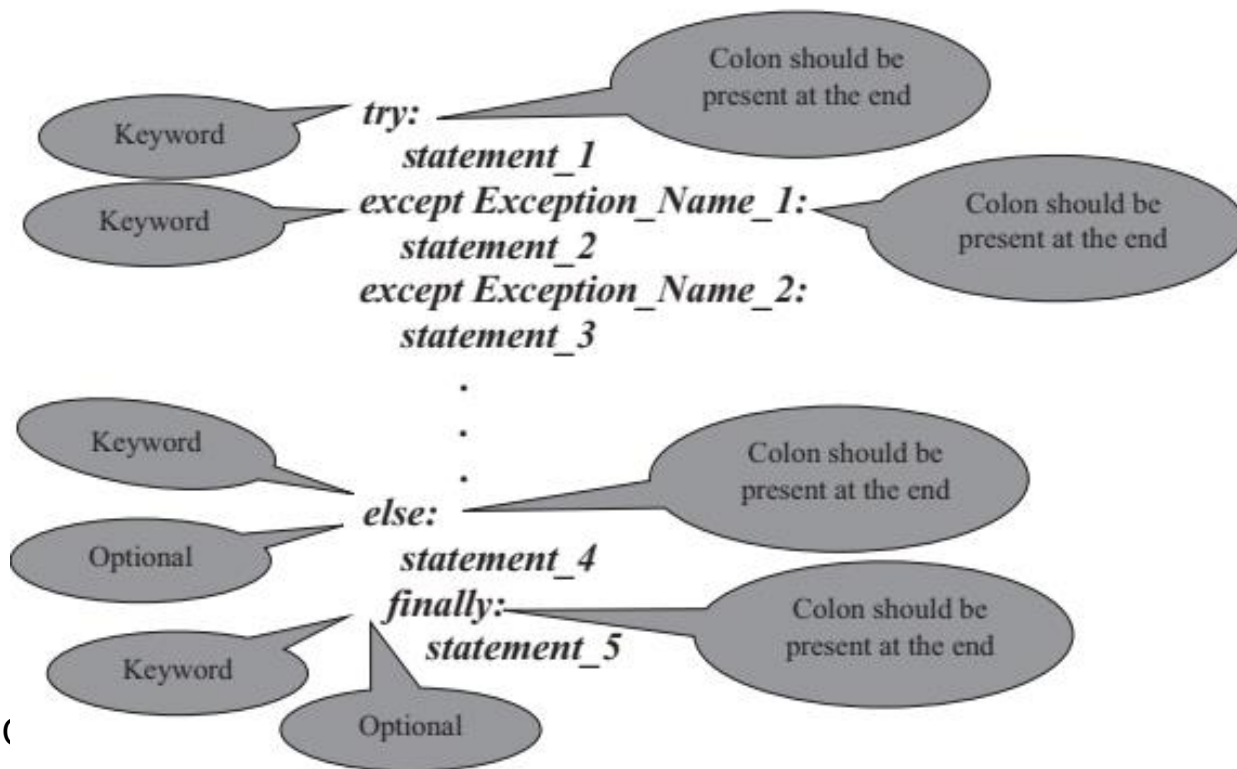
Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: Can't convert 'int' object to str implicitly 51

# Lệnh try và except

- Cú pháp cho câu lệnh try...except...finally



```
1. while True:
2.     try:
3.         number = int(input("Please enter a number: "))
4.         print(f"The number you have entered is {number}")
5.         break
6.     except ValueError:
7.         print("Oops! That was no valid number. Try again...")
```

## OUTPUT

Please enter a number: g

Oops! That was no valid number. Try again...

Please enter a number: 4

The number you have entered is 4

# Lệnh try và except

1. `x = int(input("Enter value for x: "))`
2. `y = int(input("Enter value for y: "))`
3. `try:`
4.     `result = x / y`
5. `except ZeroDivisionError:`
6.     `print("Division by zero!")`
7. `else:`
8.     `print(f"Result is {result}")`
9. `finally:`
10.     `print("Executing finally clause")`

## OUTPUT

### Case 1

Enter value for x: 8

Enter value for y: 0

Division by zero!

Executing finally clause

### Case 2

Enter value for x: p

Enter value for y: q

Executing finally clause

ValueError Traceback (most recent call last)

<ipython-input-16-271d1f4e02e8> in <module>()

ValueError: invalid literal for int() with base 10: 'p'