**Python Booleans**

Booleans represent one of two values: True or False.

Logic đại diện cho một trong hai giá trị: True hoặc False.

#Booleans biểu diễn một trong hai giá trị: True hoặc False

**Boolean Values**

Các giá trị Logic

In programming you often need to know if an expression is True or False.

Trong lập trình, bạn thường xuyên cần phải biết một biểu thức là True hoặc False

You can evaluate any expression in Python, and get one of two answers, True or False.

Bạn có thể đánh giá bất kỳ biểu thức nào trong Python, và nhận về một trong hai câu trả lời, True hoặc False.

When you compare two values, the expression is evaluated and Python returns the Boolean answer:

Khi bạn so sánh hai giá trị, biểu thức được đánh giá và Python trả về câu trả lời Logic:

Example

Ví dụ:

print(10 > 9)  
print(10 == 9)  
print(10 < 9)Try it Yourself »

When you run a condition in an if statement, Python returns True or False:

Khi bạn chạy một điều kiện trong câu lệnh if, Python trả về True hoặc False:

Example

Ví dụ:

Print a message based on whether the condition is True or False:

In thông báo dựa trên điều kiện là True hoặc False:

#In ra một tin nhắn/thông điệp dựa trên/dựa vào điều kiện là True hay False

a = 200  
b = 33  
  
if b > a:  
  print("b is greater than a")  
else:  
  print("b is not greater than a")

**Evaluate Values and Variables**

**Đánh giá các giá trị và các biến**

The bool() function allows you to evaluate any value, and give you True or False in return,

Hàm bool() cho phép bạn đánh giá bất kỳ giá trị nào, và trả về cho bạn True hoặc False,

Example

Ví dụ:

Evaluate a string and a number:

Đánh giá một chuỗi và một số:

print(bool("Hello"))  
print(bool(15))

Example

Ví dụ:

Evaluate two variables:

Đánh giá hai biến:

x = "Hello"  
y = 15  
  
print(bool(x))  
print(bool(y))  
Try it Yourself »

Most Values are True

Phần lớn các giá trị là đúng

Almost any value is evaluated to True if it has some sort of content.

Hầu như bất kỳ giá trị nào được đánh giá là True nếu nó có một số loại nội dung.

#Hầu như các giá trị được đánh giá là True nếu nó có chứa nội dung [không phải rỗng].

Any string is True, except empty strings.

Mọi chuỗi đều là True, ngoại trừ các chuỗi rỗng.

Any number is True, except 0.

Mọi số đều là True, ngoại trừ 0.

Any list, tuple, set, and dictionary are True, except empty ones.

Mọi danh sách, bộ tuple, bộ, và từ điển đều True, ngoại trừ những danh sách rỗng.

#Bất kỳ list, tuple, set và dictionary đều là True, trừ những cái [list, tuple, set và dictionary] rỗng.

Example

Ví dụ:

The following will return True:

Sau đây sẽ trả về True:

bool("abc")  
bool(123)  
bool(["apple", "cherry", "banana"])

Some Values are False

Một số giá trị là sai

In fact, there are not many values that evaluate to False, except empty values, such as (), [], {}, "", the number 0, and the value None. And of course the value False evaluates to False.

Trên thực tế, không có nhiều giá trị đánh giá thành False, ngoại trừ các giá trị rỗng, như là (), [], {}, “”, số 0, và None giá trị. Và tất nhiên giá trị False đánh giá là False.

Example

Ví dụ:

The following will return False:

Sau đây sẽ trả về False:

bool(False)  
bool(None)  
bool(0)  
bool("")  
bool(())  
bool([])  
bool({})

One more value, or object in this case, evaluates to False, and that is if you have an object that is made from a class with a \_\_len\_\_ function that returns 0 or False:

Một lần nữa giá trị, hoặc đối tượng trong trường hợp, đánh giá là False, và đó là nếu bạn có một đối tượng được tạo từ một lớp với hàm \_len\_ được trả về 0 hoặc False:

#Thêm một giá trị hay đối tượng đánh giá trả về False trong trường hợp này, và đó là khi bạn có một đối tượng được tạo ra từ một class với hàm \_\_len\_\_ trả về 0 hoặc False

#phần này ai học OOP rồi dễ hiểu, chưa học chắc khó hình dung

Example

Ví dụ:

class myclass():  
  def \_\_len\_\_(self):  
    return 0  
  
myobj = myclass()  
print(bool(myobj))

Try it Yourself »

Functions can Return a Boolean

Các hàm có thể trả về một giá trị Logic

You can create functions that returns a Boolean Value:

Bạn có thể tạo các hàm trả về giá trị Logic:

Example

Ví dụ:

Print the answer of a function:

In câu trả lời của một hàm:

def myFunction() :  
  return True  
  
print(myFunction())Try it Yourself »

You can execute code based on the Boolean answer of a function:

Bạn có thể thực thi mã dựa trên câu trả lời Logic của một hàm:

Example

Ví dụ:

Print "YES!" if the function returns True, otherwise print "NO!":

In “YES!” nếu hàm trả về True, nếu không thì int “NO!”:

def myFunction() :  
  return True  
  
if myFunction():  
  print("YES!")  
else:  
  print("NO!")

Python also has many built-in functions that return a boolean value, like the isinstance() function, which can be used to determine if an object is of a certain data type:

Python cũng có nhiều hàm tích hợp trả về giá trị logic, như hàm isinstance(), có thể được sử dụng để xác định xem một đối tượng có thuộc một kiểu dữ liệu nhất định hay không:

Example

Ví dụ:

Check if an object is an integer or not:

Kiểm tra xem một đối tượng là số nguyên hay không:

x = 200  
print(isinstance(x, int))Try it Yo111urself »----------