**1: How to check a variable’s type?**

***Để xác định kiểu của biến chúng ta phải xác định kiểu dữ liệu mà biến đó lưu trữ***

Trước hết ta xét ví dụ sau, đưới đây là cách khai báo một kiểu dữ liệu

*var = 1998 # var is integer variable*

*var = “Huyen” # var is string variable*

*var = 3.14 # var is float variable*

Kiểu dữ liệu của biến được quyết định dựa trên giá trị được lưu trữ trong biến.

Ta có thể xác định được string vì chúng được đặt trong dấu ngoặc kép ( hoặc ngoặc đơn) hoặc cũng có khi là 3 lần dấu ngoặc đơn ( 3 lần dấu ngoặc kép)

Tương tự như thế ta có thể xác định được kiểu integer vì nó không được đặt trong dấu nào cả , và nó chứa các số nguyên

Ta cũng có thể xác định được kiểu float, chúng cũng giống như integer nhưng nó chứa các số thực( các số có chứa dấu thập phân)

Ngoài ra để chắc chắn hơn, bạn có thể dùng **type**() ,nó là 1 hàm được xây dựng sẵn trong python. Nó trả về datatype của biến.

Ví dụ:

*var = 1001*

*type(var) # <type ‘int’>*

*var = 10,01*

*type(var) #<type ‘float’>*

*var = “Hoang Huyen”*

*type(var) #<type ‘string’>*

**2: In what cases, you will get SyntaxError from the compiler telling you that some of your variables have invalid names? Can you give 3 different examples of invalid names?**

Các trường hợp đặt tên cho biến chúng ta sẽ gặp lỗi SyntaxError của trình biên dịch là :

1. Tên biến có chứa dấu cách

***Ex: >>> Nico Robin = “ She is beautiful ”***

***SystaxError : invalid systax***

1. Đặt tên biến bắt đầu bằng chữ số

***Ex: >>> 7749day = “ The name of the book”***

***SystaxError : invalid systax***

1. Chứa các ký tự đặc biệt ( trừ dấu ‘’\_”)

***Ex: >>>money$ = “ Luffy current bounty of 66.000.000 Beri”***

***SystaxError : invalid systax***

***Ex: >>> money$ = 66000000***

***SystaxError : invalid systax***

1. Không được đặt tên trùng với từ khóa trong python ( def, in, class, break, else, and, as, assert, continue, del, elif, except, exec, finally, for, from, global, if, import, is, not, or, return, True, **False, None, Try, While,….)**

***Ex: def = “ DEFEND”***

***SystaxError : invalid systax***