Dynamic Typing in Python

Summary: in this tutorial, you'll learn about dynamic typing in Python and how it works.

Introduction to dynamic typing in Python

In some programming languages such as Java or C#, when declaring a variable (https://www.pythontutorial.net/python-basics/python-variables/) , you need to specify a data type for it.

For example, the following defines a variable in Java:

```
String message = 'Hello';
```

Behind the scene, Java creates a new `String` object whose value is `'Hello'` . It also creates a variable called `message` with type `String` and references the `message` variable to the `String` object.

In statically typed languages, the data types are associated with variables.

Later, if you assign an integer to the `message` variable, it's not going to work. The reason is that the `message` variable is already associated with the String type, not the integer type.

Unlike statically-typed languages, Python is a dynamically typed language. When declaring a variable in Python, you don't specify a type for it:

```
message = 'Hello'
```

In Python, the `message` variable is just a reference to an object which is a string. There is no type associated with the `message` variable.

If you assign the `message` to a number, it's perfectly fine:

```
message = 100
```

In this case, Python creates a new integer object and the `message` references to the new integer object:

To determine the type of object that a variable currently references, you use the `type()` function.

The following example defines a variable named `message` and assigned it a string `'Hello'` :

```
message = 'Hello'
print(type(message))
```

Output:

```
<class 'str'>
```

When you assign a number to the `message` variable, type of the object that the `message` variable references by also changes:

```
message = 100
print(type(message))
```

Output:

```
<class 'int'>
```

Summary

- Python is a dynamically typed language.
- In Python, variables don’t associate with any particular types.
- Use the `type()` function to get the type of the objects which variables reference.