Python Type Conversion

Summary: in this tutorial, you'll learn about the type conversion in Python and some useful type conversion functions.

Introduction to type conversion in Python

To get an input from users, you use the `input()` function. For example:

```
value = input('Enter a value:')
print(value)
```

When you execute this code, it'll prompt you for an input on the Terminal:

```
Enter a value:
```

If you enter a value, for example, a number (https://www.pythontutorial.net/python-basics/python-numbers/) , the program will display that value back:

```
Enter a value:100
100
```

However, the `input()` function returns a string (https://www.pythontutorial.net/python-basics/python-string/) , not an integer (https://www.pythontutorial.net/advanced-python/python-integers/) .

The following example prompts you for entering two input values: net price and tax rate. After that, it calculates the net price and displays the result on the screen:

```python
price = input('Enter the price ($):')
tax = input('Enter the tax rate (%):')

net_price = price * tax / 100

print(f'The net price is ${net_price}')
```

When you execute the program and enter some numbers:

```
Enter the price ($):100
Enter the tax rate (%):10
```

... you'll get the following error:

```
Traceback (most recent call last):
  File "app.py", line 4, in <module>
    net_price = price * tax / 100
TypeError: can't multiply sequence by non-int of type 'str'
```

Since the input values are strings, you cannot apply the arithmetic operator (`+`) to them.

To solve this issue, you need to convert the strings to numbers before performing calculations.

To convert a string to a number, you use the `int()` function. More precisely, the `int()` function converts a string to an integer.

The following example uses the `int()` function to convert the input strings to numbers:

```python
price = input('Enter the price ($):')
tax = input('Enter the tax rate (%):')

net_price = int(price) * int(tax) / 100
print(f'The net price is ${net_price}')
```

If you run the program, enter some values, you'll see that it works correctly:

```
Enter the price ($):100
Enter the tax rate (%):10
The net price is $ 10.0
```

Other type conversion functions

Besides the `int(str)` functions, Python support other type conversion functions. The following shows the most important ones for now:

- `float(str)` – convert a string to a floating-point number (https://www.pythontutorial.net/advanced-python/python-float/) .

- `bool(val)` – convert a value to a boolean (https://www.pythontutorial.net/python-basics/python-boolean/) value, either `True` or `False` .

- `str(val)` – return the string representation of a value.

Getting the type of a value

To get the type of a value, you use the `type(value)` function. For example:

```python
>>> type(100)
<class 'int'>
>>> type(2.0)
<class 'float'>
```

```
>>> type('Hello')
<class 'str'>
>>> type(True)
<class 'bool'>
```

As you can see clearly from the output:

- The number `100` has the type of `int` .

- The number `2.0` has the type of `float` .

- The string `'Hello'` has the type of `str` .

- And the `True` value has the type of `bool` .

In front of each type, you see the `class` keyword. It isn't important for now. And you'll learn more about the class (https://www.pythontutorial.net/python-oop/python-class/) later.

Summary

- Use the `input()` function to get an input string from users.

- Use type conversion functions such as `int()` , `float()` , `bool()` , and `str(vaue)` to convert a value from one type to another.

- Use the `type()` function to get the type of a value.