Python Float to Int

Summary: in this tutorial, you'll learn how to convert a float to an integer.

Suppose that you have a float (https://www.pythontutorial.net/advanced-python/python-float/) such as 20.3, and you want to convert it to an integer (https://www.pythontutorial.net/advanced-python/python-integers/) .

When you convert a float to an integer, you'll have a data loss. For example, `20.3` may become `20` or `21` .

Python provides you with some functions in the `math` module for converting from a `float` to an `int` , including:

- Truncation
- Floor
- ceiling

Truncation

The `trunc(x)` function returns the integer part of the number `x` . It ignores everything after the decimal point. For example:

```
from math import trunc

print(trunc(12.2))
print(trunc(12.5))
print(trunc(12.7))
```

Output:

```
12
12
12
```

Similarly, the int() constructor accepts a float and uses truncation to cast a `float` to an `int` :

```
print(int(12.2))
print(int(12.5))
print(int(12.7))
```

Floor

The `floor(x)` function returns the largest integer less than or equal to `x` . For example:

```
from math import floor

print(floor(12.2))
print(floor(12.5))
print(floor(12.7))
```

Output:

```
12
12
```

```
12
```

The following shows how floor() function is applied to a positive number:

For positive numbers, `floor(x)` and `trunc(x)` return the same result. However, it’s not the case for negative numbers. For example:

The following picture shows how the `floor()` function is applied to a negative number:

```
from math import floor, trunc

print(floor(-12.7))
print(trunc(-12.7))
```

Output:

```
-13
-12
```

The following picture illustrates the difference between the `floor()` and `trunc()` function when you apply them to a negative number:

Ceiling

The `ceil(x)` function returns the smallest integer greater than or equal to `x` . For example:

```
from math import ceil


print(ceil(12.7))
```

Output:

```
13
```

The following illustrates how the `ceil()` function is applied to a positive number:

This example uses the ceil() function for negative numbers:

```
from math import ceil

print(ceil(-12.7))
```

Output:

```
-12
```

The following illustrates how the `ceil()` function is applied to a negative number:

Summary

- Convert a float to an int always results in a data loss.
- The `trunc()` function returns the integer part of a number.
- The `floor()` function returns the largest integer less than or equal to a number.
- The `ceil()` function returns the smallest integer greater than or equal to a number.