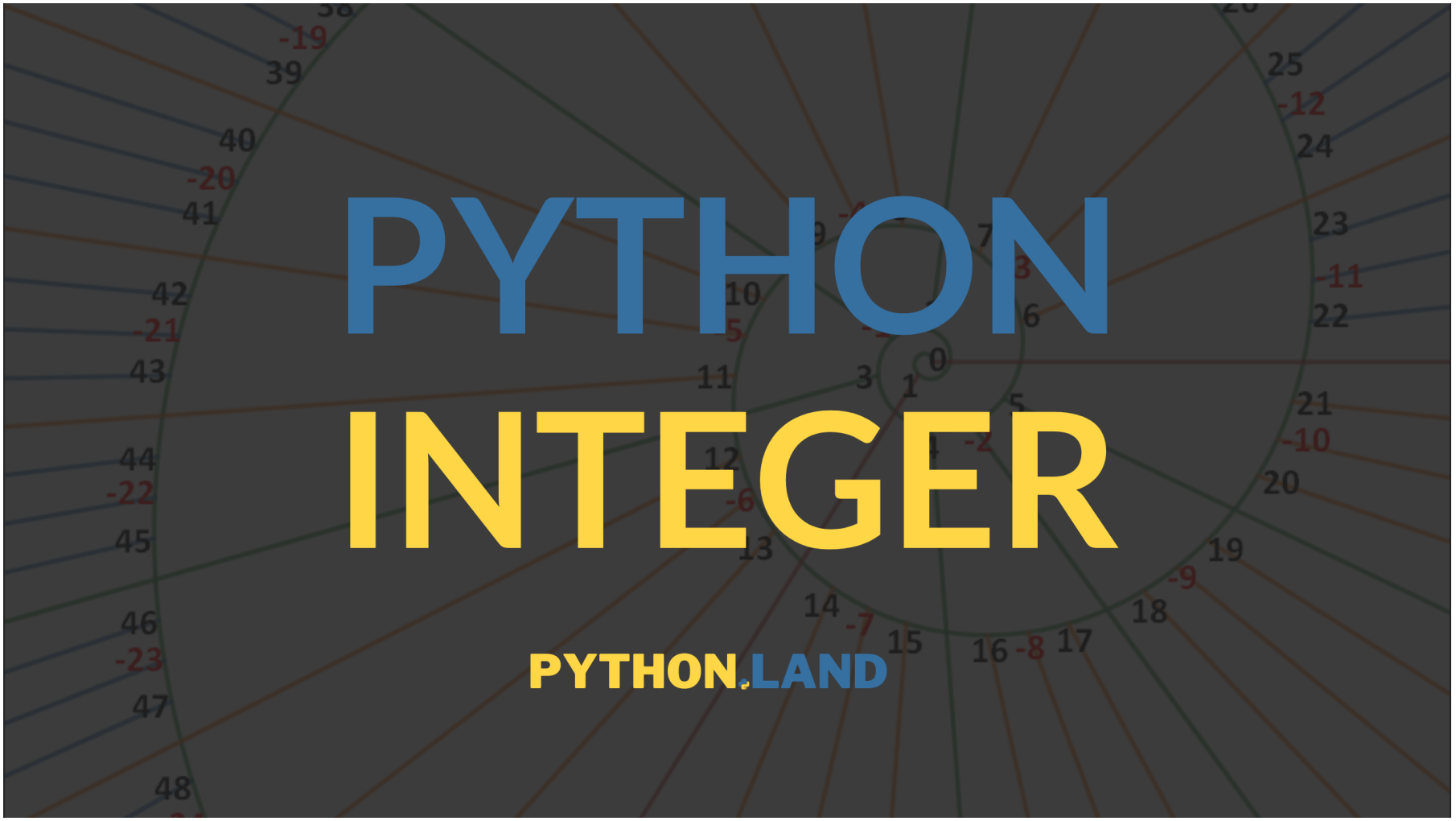
Previous: Python Data Types

Next: Python Float ›

Home » Python Data Types » Python Integer: Non-Fractional Numbers (With Example Code)



Python Integer: Non-Fractional Numbers (With Example Code)

December 11, 2022

The Python integer is a non-fractional number, like 1, 2, 45, -1, -2, and -100. It's one of the three types of numbers Python supports natively, the others being floating-point numbers and complex numbers.

Table of Contents [hide]

Beginners Python Course (2023)

Modules, Packages, And Virtual Environments (2023)

Max size of a Python integer

Unlike many other programming languages, integers in Python 3 can have large values. They are unbounded, meaning there is no limit to their size. For example:

```
>>> num = 9876234509870987234500 0
>>> num + 1
98762345098709872345001
```

Of course, there is a limit since your computer does not have unlimited memory. However, for all practical purposes, you don't have to worry about it.

Integer types

Unlike Python 2 and many other languages, Python 3 has only one integer type. This is part of Python's aspiration to be a clean, easy-to-learn language. It's one less thing we have to worry about. For more details, see PEP-0237.

Converting from and to an integer

String to integer

To convert a string to an integer in Python, use the `int()` function:

```
>>> int('100')
100
```

Integer to string

To convert an integer to a string in Python, use the `str()` function:

```
>>> str(200)
'200'
```

Float to integer

To convert a float to an integer, use the `int()` function:

```
>>> int(2.3)
2
```

Python random integer

Many use cases require a random integer. For this, you need to import the module `random`. Be warned that this offers *pseudo-randomness*, which is not suitable for cryptography.

Let's get a random number:

```
>>> import random
>>> random.randint(1,10)
```

The above instruction returns a pseudo-random number from 1 to 10 inclusive, which means *including* 1 and 10. For full details of the random module, visit the Python documentation.

Is it a Python integer?

We can use the `type()` function to check if a value is an integer. It will return `int` for integers. Here's a simple example of how to use this in an `if`-statement:

```
>>> type(2)
int
>>> if isinstance(2, int):
...     print('An integer')
...
An integer
```

Don't use `if type(2) == int`.
Using `isinstance()` is almost always the better, cleaner way and covers more use cases, like subclasses.

‹ Previous: Python Data Types

Next: Python Float