Python Modulo

Summary: in this tutorial, you'll learn about the Python modulo operator (`%`) and how to use it effectively.

Introduction to the Python modulo operator

Python uses the percent sign (%) as the modulo operator. The modulo operator always satisfies the following equation:

```
N = D * ( N // D) + (N % D)
```

In this equation:

- N is the numerator.
- D is the denominator.
- // is the floor division operator (https://www.pythontutorial.net/advanced-python/python-floor-division/)
- And % is the modulo operator

If both N and D are positive integers (https://www.pythontutorial.net/advanced-python/python-integers/) , the modulo operator returns the remainder of N / D. However, it is not the case for the negative numbers.

Therefore, you should always stick with the above equation.

Simple Python modulo operator examples

The following example illustrates how to use the modulo operator (%) with positive integers:

```
a = 16
b = 5

m = a % b
f = a // b

# show the result

print(f'{a} % {b} = {m}')  # 1
print(f'{a} // {b} = {f}')  # 3
```

Output:

```
1
3
```

For positive numbers, the result is quite apparent. And you can check the equation quickly:

```
16 = 5 * (16 // 5) + 16 % 5
16 = 5 * 3         + 1
```

The following shows how to use the modulo operator (`%`) with negative integers:

```
a = -16
b = 5

m = a % b
```

```
f = a // b

# show the result

print(f'{a} % {b} = {m}')  # 4
print(f'{a} // {b} = {f}')  # -4
```

And the equation is satisfied:

```
-16 = 5 * (-16 % 5) + (-16) % 5
-16 = 5 * -4        - 4
```

Practical Python modulo operator examples

Let's take some practical examples of using the modulo operator (%)

1) Using the modulo operator to check if a number is even or odd

The following defines a function (https://www.pythontutorial.net/python-basics/python-functions/) that uses the modulo operator (%) to check if a number is even:

```
def is_even(num):
    return num % 2 == 0
```

And the following defines a function that uses the modulo operator to check if a number is odd:

```
def is_odd(num):
    return num % 2 != 0
```

2) Using the modulo operator to convert units

The following example uses the modulo operator (`%`) to convert seconds to days, hours, minutes, and seconds. It can be handy if you want to develop a countdown program:

```
from math import floor


def get_time(total_seconds):
    return {
        'days': floor(total_seconds / 60 / 60 / 24),
        'hours': floor(total_seconds / 60 / 60) % 24,
        'minutes': floor(total_seconds / 60) % 60,
        'seconds': total_seconds % 60,
    }


print(get_time(93750))
```

Output:

```
{'days': 1, 'hours': 2, 'minutes': 2, 'seconds': 30}
```

Summary

- Python uses the percent sign (`%`) as the modulo operator.
- The modulo operator (%) always satisfies the equation `N = D * ( N // D) + (N % D)` .