# Урок 02.02. Математические операции

## Деление без остатка (нацело) и с остатком

Рассмотрим такую задачу: в компании есть 41 тонна яблок, которые нужно разместить по ящикам. В один ящик входит 3 тонны яблок. Вопрос: сколько можно заполнить целых ящиков?

Давайте начнем писать программу:

apples = 41

boxes = 3

full\_boxes = apples / boxes

print(‘Всего нужно ящиков:’, full\_boxes)

После запуска программы мы увидим дробное число, а нам нужны всё же целые ящики. Так вот для этого предусмотрено целочисленное деление, тогда программа будет выглядеть так:

apples = 41

boxes = 3

full\_boxes = apples // boxes

print(‘Всего нужно ящиков:’, full\_boxes)

Теперь мы видим целое число. И это число получается после отбрасывания дробной части при делении. Происходит не округление, а именно убирание/стирание/отбрасывание дробной части.

А теперь давайте узнаем сколько тонн не поместилось в ящики. Для напишем следующую программу:

apples = 41

boxes = 3

full\_boxes = apples // boxes

print(‘Всего нужно ящиков:’, full\_boxes)

rest\_apples = apples % boxes

print(‘Осталось тонн яблок:’, rest\_apples)

С этой задачей мы закончили. Теперь заказчик знает сколько ему нужно ящиков и сколько тонн яблок у него осталось.

## Закономерность чисел и остаток от деления

Далее рассмотри ряды чисел:

3, 8, 13, 18, … 13 % 5 =3, 18 % 5 = 3

6, 11, 16, 21, … 11 % 5 = 1, 16 % 5 = 1

Как мы видим, числа у этих рядов увеличиваются на 5 одинаково. Но эти ряды отличаются. Так если взять остаток от деления этих числе на 5, то мы увидим, что разница между этими рядами именно в остатке от деления или начале отсчета.

Давай другой пример рссмотрим:

6, 16, 26, 36, … 26 % 10 = 6

9, 19, 29, 39, … 39 % 10 = 9

И тут мы видим одну очень прекрасную закономерность: если взять от числа остаток от деления на 10, то мы получим последнюю цифру числа. Давайте проверим это на практике:

print(3 % 10)

print(52 % 10)

print(179 % 10)

print(1045 % 10)

После запуска программы мы видим на экране последнюю цифру числа.

А если заменить «%» на «//»:

print(3 // 10)

print(52 // 10)

print(179 // 10)

print(1045 // 10)

Запустим программу. И мы видим другой результат, оно и понятно. Но что же мы видим. А на самом деле нам представлены числе без последней цифры.

## Сокращенные арифметические операторы

Представьте такую задачу: мы с вами идем по магазину и набираем продукты, много продуктов. Приходим на кассу и начинаем считать сумму. Берем один продукт и вводим сумму, потом второй и прибавляем к предыдущему результату и так далее. В таком случае получим такую программу:

products = 0

products = products + 100

print(‘Итоговая сумма:’, products)

products = products + 200

print(‘Итоговая сумма:’, products)

products = products + 550

print(‘Итоговая сумма:’, products)

products = products + 270

print(‘Итоговая сумма:’, products)

Таким образом, мы набрали корзину продуктов и посчитали общую сумму всех продуктов. На самом деле нашу программу можно записать в более коротком и понятном стиле:

products = 0

products += 100

print(‘Итоговая сумма:’, products)

products += 200

print(‘Итоговая сумма:’, products)

products += 550

print(‘Итоговая сумма:’, products)

products += 270

print(‘Итоговая сумма:’, products)

Аналогичным образом обстоят дела и с «-», «\*», «/», «//», «%».