



Урок 7. Python: частное, остаток и цифры числа

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 7 из 34 · 45 минут · Модуль «Числа»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

Число 407 содержит ноль, но при разборе на цифры он не должен исчезнуть. Как получить сотни, десятки и единицы без работы со строками?

Результат урока: Разложить число от 100 до 999 на цифры с помощью // и %, восстановить исходное число и проверить внутренний ноль.

01. Проблемная ситуация

Предложены формулы для десятков числа 407: $407 // 10$ и $407 // 10 \% 10$. Вычислите обе и объясните, почему ответ 40 ещё не является одной цифрой. Отдельно предскажите результат для 470.

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Деление с остатком и разряд

Для неотрицательного n и положительного b : $n = (n // b) * b + n \% b$, причём $0 \leq n \% b < b$. // даёт целое частное, % — остаток. Делитель не может быть нулём. В этой задаче вход неотрицательный; у отрицательных чисел // округляет вниз, поэтому правило «просто убрать дробь» неверно.

Для трёхзначного n : $hundreds = n // 100$; $tens = n // 10 \% 10$; $units = n \% 10$. Деление на 10 убирает единицы, остаток от 10 оставляет последнюю цифру полученного числа.

Проверка обратным действием: $100 * hundreds + 10 * tens + units$ должно равняться n . Сумма цифр — другое значение; она не восстанавливает число. Внутренний ноль — важный тест, а не особый исключительный случай.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

Для 407: $407 // 100 = 4$; $407 // 10 = 40$, $40 \% 10 = 0$; $407 \% 10 = 7$. Восстановление: $400 + 0 + 7 = 407$. Код выводит три цифры через пробел, затем восстановленное число.

```
n = int(input())
hundreds = n // 100
tens = n // 10 % 10
units = n % 10
print(hundreds, tens, units)
print(100 * hundreds + 10 * tens + units)
```

Python: частное, остаток и цифры числа

Сотни	$407 // 100 = 4$
Десятки	$407 // 10 \% 10 = 0$
Единицы	$407 \% 10 = 7$
Восстановление	$4 \times 100 + 0 \times 10 + 7 = 407$

Число $407 = 4 \times 100 + 0 \times 10 + 7$; остаток от 10 выделяет одну цифру.



03. Практика 1. Проверяем материал

Разбираем трёхзначный код

Договор ввода: n — целое от 100 до 999. Для каждого теста сначала выполните разложение на бумаге.

Ввод n	Цифры; восстановление
407	4 0 7; 407
100	1 0 0; 100
999	9 9 9; 999

Запустите пример на трёх входах. Дополните третьей строкой вывод суммы цифр. Сохраните `digits.py`; запишите, почему 100 нельзя проверять только по ненулевым цифрам. Критерии: три правильные цифры, восстановление, суммы 11/1/27, границы 100 и 999 проверены.

04. Практика 2. Перенос способа

Переносим разложение на время

Новая величина: длительность n секунд, n — любое неотрицательное целое. Напишите `time_parts.py`, выводящий полные минуты и оставшиеся секунды. Используйте основание 60, не 10. Тесты: 125 → 2 5; 59 → 0 59; 60 → 1 0; 0 → 0 0. Проверьте восстановление $60 \cdot \text{minutes} + \text{seconds}$ на бумаге.

05. Обсуждение и границы вывода

Коллега предлагает всегда использовать обычное `/`, затем округлять. Почему для минут это рискованно? Сравните $59/60$ с целым частным и объясните альтернативу через `//`.

06. Индивидуальная проверка

Разберите 806 на цифры, найдите сумму и восстановите число. Затем разложите 359 секунд на минуты и секунды. Критерии: цифры, восстановление/сумма, время.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Полный вариант ниже; сумма 407 равна 11, числа 100 — 1, числа 999 — 27. Два нуля числа 100 должны остаться в строке цифр. По 1 баллу за четыре критерия.

```
n = int(input())
h = n // 100
t = n // 10 % 10
u = n % 10
print(h, t, u)
print(100 * h + 10 * t + u)
print(h + t + u)
```

Практика 2: разбор и критерии

Полные минуты $n//60$, остаток $n\%60$. Для 125: $2*60+5=125$. Критерии: основание 60, остаток 0...59, все четыре теста.

```
n = int(input())
minutes = n // 60
seconds = n % 60
print(minutes, seconds)
```

Обсуждение: возможный вывод

Нужно количество полных минут, а не ближайшее число минут: округление $59/60$ даст 1, хотя полной минуты нет. // и % прямо выражают частное и остаток. Обычное / полезно для средней величины, но не заменяет разложение.

Индивидуальная проверка: ответ и критерии

8 0 6; сумма 14; $100*8+10*0+6=806$. $359//60=5$, $359\%60=59$; $5*60+59=359$. 3 балла.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, локальную папку с правом записи, Python/IDLE, распечатку ученической части PDF, бумагу и ручку. Внешние библиотеки не нужны: только стандартный Python 3.11 или новее. Все программы, входные данные, таблицы и ключи находятся в статье и PDF. Детские аккаунты, сеть и регистрация не нужны.

Уровень — начинающий, 7 класс. Предпосылки: int и арифметические выражения урока 6. Работа в паре: автор прогнозирует и вводит код, проверяющий ведёт журнал; затем роли меняются. На бумаге трассируйте готовый код и входные данные; для графики используйте координатную сетку и схему. Учитель может показать заранее полученный вывод/PNG. Бумажный вариант проверяет понимание, но не выдаётся за реальный запуск учеником. Чтение, ввод и проверка включены в план. Матрица программы — материал учителя.

Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ



07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Деление нацело. Остаток от деления	Выделяет цифры и части длительности	Обратное равенство выполнено
Обработка целых чисел	Проверяет внутренний ноль и границы	Тесты 100, 999, 0 секунд

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

Python 3.11: // и %

<https://docs.python.org/3.11/reference/expressions.html#binary-arithmetic-operations>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 17.10.2026.