



Урок 8. Python: вещественные числа и ТОЧНОСТЬ

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 8 из 34 · 45 минут · Модуль «Числа»

Уровень: начинающий программист; Python 3.

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

Программа выводит 0.30000000000000004 вместо 0.3. Это ошибка сложения или особенность хранения числа? Различим точность вычисления и вид строки на экране.

Результат урока: Объяснить приближённое хранение float, сравнить числа с допуском и вывести результат с заданным количеством знаков.

01. Проблемная ситуация

Запишите прогноз для $0.1 + 0.2 == 0.3$. Затем сравните две реакции: «сломался компьютер» и «нужно проверить представление числа». Сформулируйте проверку, которая допускает ошибку меньше 0.000001, но не принимает 0.31 за 0.3.

02. Понятия и разобранный пример

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

Значение, представление и формат

float обычно хранит двоичное приближение вещественного числа. Многие конечные десятичные дроби, в том числе 0.1, не представимы точно в таком формате. Небольшое расхождение не означает, что допустима любая ошибка.

$\text{abs}(x - \text{expected}) \leq \text{tolerance}$ — простая проверка абсолютной ошибки. Допуск выбирают по задаче и единицам. В этой учебной задаче числа порядка единицы и допуск 0.000001; для любых масштабов это не универсальное правило.

Формат `f"{x:.2f}"` создаёт строку с двумя знаками после точки. Он не изменяет исходный `x` и не повышает точность измерений. При вводе float используйте точку, например 2.5. В школьных данных ниже значения учебные, не реальные измерения учеников.

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

$x = 0.1 + 0.2$ даёт 0.30000000000000004; $x == 0.3$ — False. Абсолютная разность около $5.55 \cdot 10^{-17}$, меньше 0.000001, поэтому проверка с допуском True. Формат с двумя знаками выводит 0.30. Запись 0.30 — вид результата, а не новое более точное измерение.

```
x = 0.1 + 0.2
print(x)
print(x == 0.3)
print(abs(x - 0.3) <= 0.000001)
print(f"{x:.2f}")
```

Python: вещественные числа и точность

Вычисление	$0.1 + 0.2 \rightarrow 0.30000000000000004$
Сравнение ==	$x == 0.3 \rightarrow \text{False}$
Допуск 0.000001	$\text{abs}(x - 0.3) \leq \text{допуск} \rightarrow \text{True}$
Только оформление	<code>f"{x:.2f}"</code> → строка 0.30

Три разных действия: вычислить float → проверить ошибку по допуску → оформить строку с двумя знаками.



03. Практика 1. Проверяем материал

Проверяем десятичную дробь

Запускайте пример целиком, затем проведите два независимых варианта, заменяя только первую строку на $x=0.31$ и $x=0.3$.

Проверяемое значение x	$\text{abs}(x-0.3) \leq 0.000001$
$0.1+0.2$	True
0.31	False
0.3	True

Составьте журнал: исходное значение, точное сравнение, проверка допуска, форматированная строка. Подчеркните, какие строки отвечают за вычисление, сравнение и оформление. Не заменяйте допуск округлением перед сравнением. Критерии: три запуска, верный False для 0.31, объяснение 0.30, различие числа и строки.

04. Практика 2. Перенос способа

Среднее и число знаков

Новые учебные данные: три длины в сантиметрах вводятся отдельными строками. Напишите `mean.py`, выводящий среднее с двумя знаками. Тесты: 2.1,2.2,2.3 $\rightarrow$ 2.20; 0,0,0 $\rightarrow$ 0.00; 1,2,2 $\rightarrow$ 1.67. На бумаге найдите точное среднее последнего набора как дробь. Объясните, почему строка 1.67 не означает точного равенства $5/3$.

05. Обсуждение и границы вывода

Можно ли печатать десять знаков после точки и утверждать, что измерение стало точнее? Сравните вычислительную погрешность `float` с неточностью исходной линейки. Предложите разумный вывод для данных до десятых сантиметра.

06. Индивидуальная проверка

Предскажите строку для `f"{1.5:.2f}"`. При $x=0.3001$ пройдет ли проверка $\text{abs}(x-0.3) \leq 0.000001$? Изменяет ли форматирование x ? Критерии: строка, сравнение, объяснение.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Для 0.31 сравнение с 0.3 False, допуск False, строка 0.31. Для 0.3 сравнение True, допуск True, строка 0.30. Исходный пример даёт False/True и строку 0.30. По 1 баллу за четыре критерия; ошибка не равна ровно 0.01 в двоичном представлении, но явно больше допуска.

Практика 2: разбор и критерии

Среднее $(a+b+c)/3$. Для последнего набора $5/3=1.666...$; вывод 1.67 — округлённая строка. Критерии: три float-ввода, деление всей суммы, два знака, объяснение округления.

```
a = float(input())
b = float(input())
c = float(input())
mean = (a + b + c) / 3
print(f"mean: .2f")
```

Обсуждение: возможный вывод

Дополнительные знаки меняют оформление, а не качество исходных данных. Ошибка хранения и ошибка измерения — разные причины неточности. В задании два знака требуются для тренировки формата; реальную точность измерений по этому выводу устанавливать нельзя.

Индивидуальная проверка: ответ и критерии

Строка 1.50. Проверка False: разность около 0.0001, больше допуска. Форматирование не меняет x, создаёт строку. 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: арифметика и консольный ввод. 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
Особенности представления вещественных чисел	Сравнивает точное равенство и допуск	Объясняет малую ошибку
Операции с вещественными числами	Вычисляет среднее и форматирует	Тесты 2.20, 0.00, 1.67

Источники и права на изображения

Исходная программа курса

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

Python 3.11: представление вещественных чисел

<https://docs.python.org/3.11/tutorial/float.html>

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.