Самое важное

Ввод вещественного числа

Ранее мы говорили о вводе целых чисел — int(input(...)), что, по сути, является двумя последовательными операциями:

1) ввод какой-либо строки;
2) преобразование строки к числовому типу int.

Однако если вы введете не целое число, int() выдаст ошибку, так как эта функция не сможет преобразовать точку-разделитель в строке в число.

На помощь придёт функция float(): она знает, как обращаться с точками, и её можно будет использовать вместо int.

Синтаксис
float(input(<сообщение пользователю>))

Важно знать! float может работать и с целыми числами, но это всё равно будет тип float. Пример: x = float(input(...)) — если мы введём в консоль 1, то input передаст нам строку ‘1’, а float преобразует эту строку в число 1.0. Это всё ещё будет единица, но тип у неё будет float.

Округление при помощи round()
Функция round() позволяет округлять вещественные числа до нужного нам количества знаков после запятой.

Синтаксис
round(<число>, <количество знаков после запятой>)

Пример
x = 1.23456
x_rounded = round(x, 2) — так мы округлим наше число до 1.23.

Приведение типов
Операции int() и float() называются операциями явного приведения типов. Это значит, что с их помощью можно явно изменить один тип данных на другой. Мы уже делали это со строкой, которую получаем из input(), но это не единственный способ использования этих операций.

Пример: int(1.25) → 1 — в результате такой операции мы отбросим всё, что было в числе после точки, и оставим только целую часть числа (тип итогового числа будет int).

При этом обратите внимание, что в этот раз у Python не возникает проблем с обработкой точки, как это было со строкой. Это происходит, потому что преобразуется

вещественное число, а не строка. Вещественное число несёт в себе информацию о том, для чего нужна эта точка (для разделения целой и дробной частей). Строка же, в отличие от float-чисел, такой информации не несёт, так как точку в строке можно использовать для совершенно разных целей.

Пример: float(1) → 1.0 — в результате этой операции мы никак не изменим итоговое значение числа, но тип числа будет изменен на float.

Модуль math
Модуль — это файл с кодом, таким же, какой пишется в задачах. Модуль нужен для того, чтобы не писать вручную все операции с нуля. Например, модуль math позволит нам пользоваться готовыми математическими функциями вроде sqrt() — вычисления корня числа.

Подключение модуля выполняется при помощи операции import math. Далее через обращение к модулю и название функции можно пользоваться всеми встроенными возможностями этого модуля: math.sqrt(x), math.sin(x), math.cos(x).

Более подробный перечень функций модуля math можно найти в документации, русский аналог — https://docs-python.ru/standart-library/modul-math-python/.

Не допускай следующих ошибок!

Не забывайте, что каждая функция должна заканчиваться скобками (). Если вы используете функции вместе, как это было с float(input()), то не забывайте ставить правильное количество скобок в конце строки.

x = float(input() — такая запись приведёт к синтаксической ошибке, причём указывать она будет на следующую строку кода, так как именно там Python будет искать недостающую скобку.

Также не стоит забывать, что ввод через input никак не ограничивается, то есть пользователь вполне себе может ввести что угодно, помимо чисел. В случае ввода букв или других символов кроме чисел int/float будут выдавать ошибку, а значит, операции вроде int(input()) — это потенциально опасные места кода.