

Чтобы решить эту задачу можно использовать список.

Нужно создать список, где каждый элемент будет соответствовать количеству дней в каждом месяце года.

1) Создание списка с днями месяцев: Сначала мы создаем список `month_days_list`, в котором каждый элемент соответствует количеству дней в каждом месяце года. Этот список будет выглядеть так: `[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]`. Здесь каждый элемент списка представляет количество дней в соответствующем месяце, начиная с января и заканчивая декабрем.

2) Ввод пользователя и преобразование в число: Мы запрашиваем у пользователя ввести номер месяца через `input`, преобразуем его в целое число.

3) Индексация и сопоставление с номерами месяцев: В Python индексация элементов в списке начинается с 0. Однако, номера месяцев начинаются с 1 (январь - 1, февраль - 2, и т.д.). Чтобы сопоставить введенный номер месяца с правильным индексом в списке, нужно от номера месяца отнять 1. Например, если пользователь вводит номер месяца 3 (март), мы используем индекс 2 ($3 - 1$) для доступа к соответствующему элементу в списке.

4) Проверка допустимости введенного номера: До обращения к списку необходимо убедиться, что полученный индекс находится в допустимых пределах (от 0 до 11). Если индекс выходит за эти пределы, это означает, что введен недопустимый номер месяца, и пользователю выводится соответствующее сообщение.

5) Вывод количества дней: Если индекс допустим, количество дней в соответствующем месяце выводится пользователю. Это делается путем обращения к элементу списка по индексу.

Применение списка для решения этой задачи позволяет сократить код, уменьшить количество условий `if`, повысить читаемость и упростить поддержку кода. Такой подход также облегчает изменения в коде, например, если понадобится добавить поддержку високосных лет.