Бывают ситуации, когда вашей программе необходимо выполнить некоторый фрагмент кода, только если выполнено определенное условие. Это условие можно установить в Python, и в этом разделе мы узнаем, как это сделать!

**Простая инструкция if**

Итак, в Python фрагмент кода, который должен выполняться только при некотором условии, должен быть помещен в тело **оператора if**. Шаблон тот же, что и в английском языке: сначала идет ключевое слово **if**, затем условие, а затем список выражений для выполнения. Условие всегда является **логическим выражением**, то есть его значение равно либо True, либо False. Вот один из примеров того, как должен выглядеть код с условным выражением:

biscuits = 17

if biscuits >= 5:

print("It's time for tea!")

Обратите внимание, что условие заканчивается двоеточием, а новая строка начинается с **отступа**. Обычно для обозначения каждого уровня отступа используются 4 пробела. Фрагмент кода, в котором все строки имеют одинаковый уровень отступа, называется **блоком кода**. В Python для разделения различных блоков кода используется *только*отступ, следовательно, только отступ показывает, какие строки кода должны выполняться при выполнении if инструкции, а какие должны выполняться *независимо*от if инструкции. Посмотрите на следующий пример:

if biscuits >= 5:

print("It's time for tea!")

print("What tea do you prefer?")

print("What about some chocolate?")

В этом примере строка "It's time for tea!", так же как "What tea do you prefer?", будет напечатана только при наличии 5 или более файлов cookie. Строка "What about some chocolate?" будет напечатана независимо от количества файлов cookie.

ifОператор выполняется, только если выполняется его условие (логическое значение равно True), в противном случае он пропускается.

Логические значения в основном проясняют, нужно ли выполнять фрагмент кода или нет. Поскольку сравнения приводят к bool, всегда полезно использовать их в качестве условия.

Однако есть одна ошибка. Не следует путать оператор сравнения для равенства == с оператором присваивания =. Только первое обеспечивает надлежащее условие. Постарайтесь избежать этой распространенной ошибки в своем коде.

**Вложенный оператор if**

Иногда условие оказывается слишком сложным для простого if утверждения. В этом случае вы можете использовать так называемые **вложенные**if инструкции. Чем больше if операторов вложено, тем сложнее становится ваш код, что обычно не очень хорошо. Однако это не означает, что вам нужно любой ценой избегать вложенных if инструкций. Давайте взглянем на приведенный ниже код:

rainbow = "red, orange, yellow, green, blue, indigo, violet"

warm\_colors = "red, yellow, orange"

my\_color = "orange"

if my\_color in rainbow:

print("Wow, your color is in the rainbow!")

if my\_color in warm\_colors:

print("Oh, by the way, it's a warm color.")

Приведенный выше пример иллюстрирует вложенный if оператор. Если переменная my\_color представляет собой строку, содержащую название цвета из радуги, мы вводим тело первого ifоператора. Сначала мы печатаем сообщение, а затем проверяем, относится ли наш цвет к теплым цветам. Оператор членства in просто показывает, является ли my\_color подстрокой соответствующей строки rainbow или warm\_colors. Так же, как и арифметические сравнения, он возвращает логическое значение.

Вот что мы увидим в нашем случае:

Wow, your color is in the rainbow!

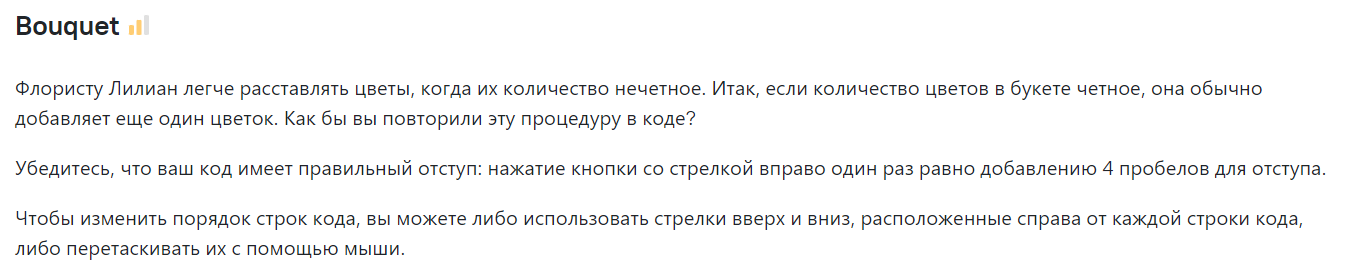
Oh, by the way, it's a warm color.

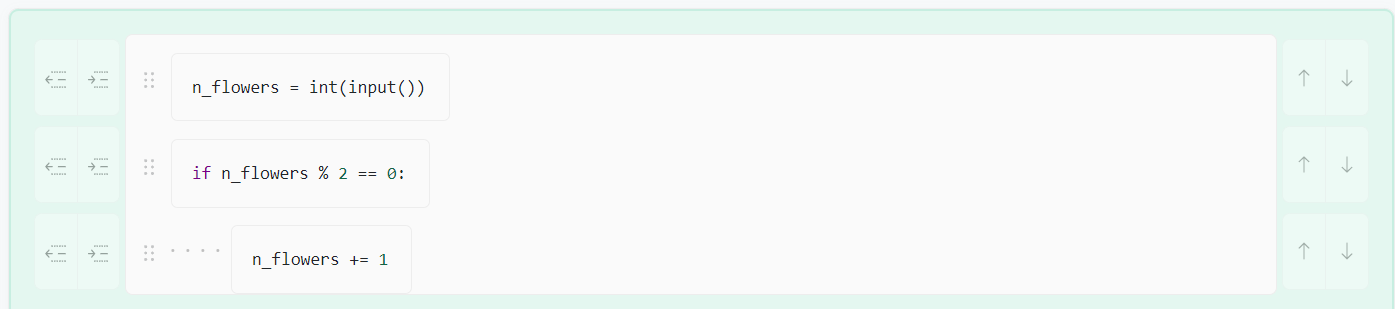
Когда дело доходит до вложенных if операторов, правильный отступ имеет решающее значение, поэтому не забывайте делать отступы в каждом операторе, который начинается с if ключевого слова.

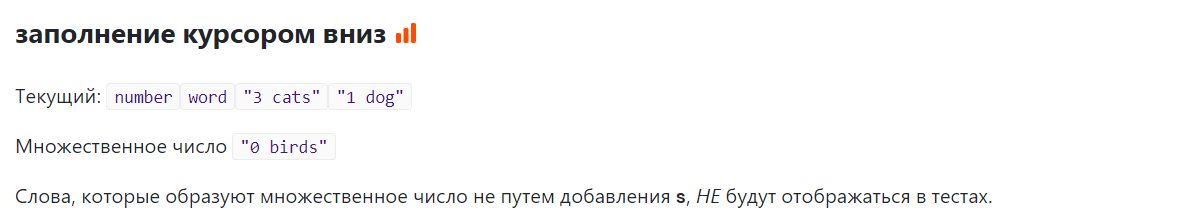
**Краткие сведения**

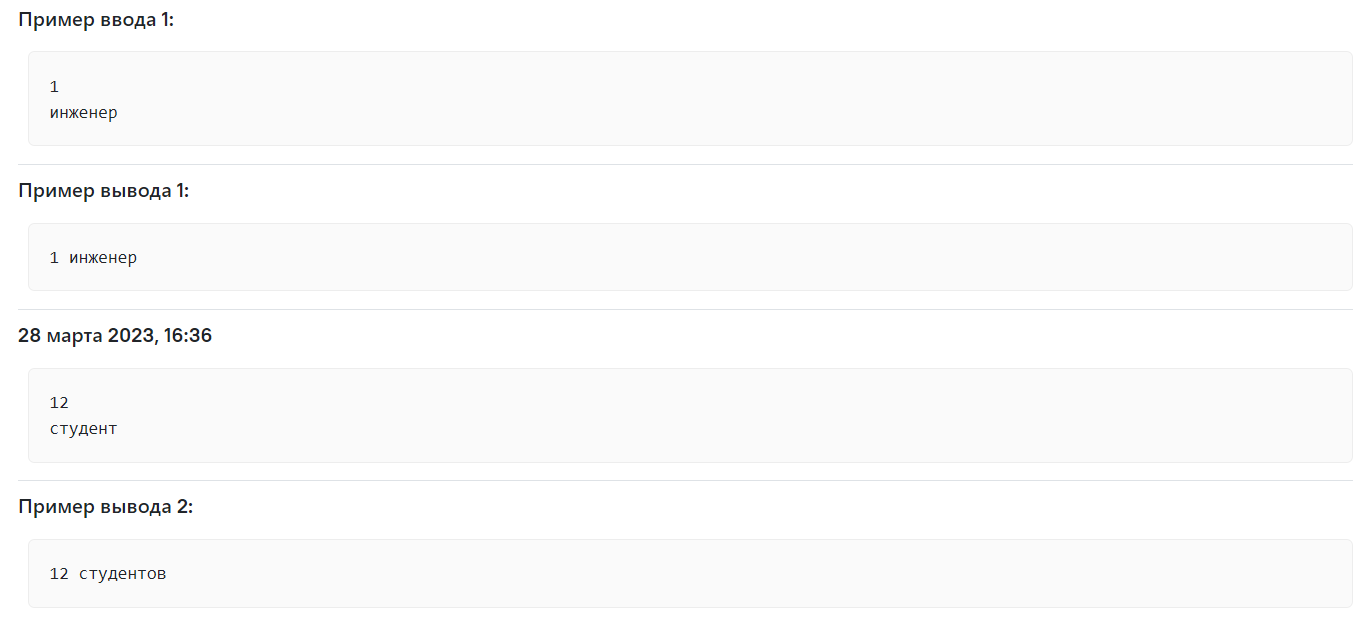
Подводя итог, в этом разделе мы узнали, как писать и вкладывать условия в Python с помощью операторов if.

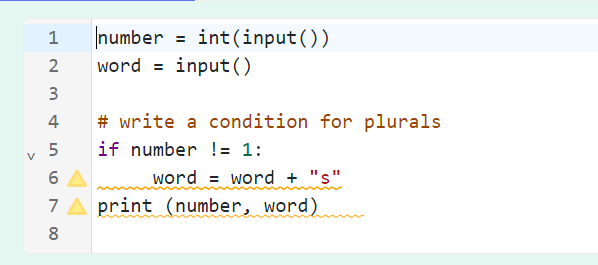
**Задачи**

****

****

****

****

****