# Урок 03.01. Условный оператор

Давайте рассмотри такую ситуацию: мы с вами хотим стать программистом и для этого нам нужно посещать занятия по программированию и программировать дома. Для этого у нас есть простой алгоритм:

Сходить на занятие

Записать задания

Выполнить задания дома

Однако, мы можем не попасть на занятия. Например, заболели или проспали, а может просто забыли. Всякое может быть. И тогда весь наш план не может быть выполнен. И тут у нас возникают условия, которые влияют на нашу задачу. Данный пример показывает, что мы в своей жизни постоянно сталкиваемся с условиями, и в зависимости от них происходят те или иные события.

Рассмотрим второй пример: мы пошли в магазин и нам нужно купить мороженое. Родители нам дали 50 рублей. И вот мы пришли в магазин и выбираем мороженое. Конечно, мы начинаем сравнивать цены с тем, сколько у нас денег. А вот дальше внимательно давайте рассуждать: если нам хватает денег, то мы покупаем мороженое. Для программирования это будет звучать немного по-другому, а именно: **если** количество денег больше стоимости мороженого, **то** вывести «мороженое можно купить». Обратите внимание на конструкцию **если-то,** она нам пригодится. А программа будет выглядеть следующим образом:

price = int(input(‘Введите цену мороженого: ’))

if price <= 50:

 print(‘Мороженое можно купить.’)

Проверим нашу программу. Введем сначала 100, в следующий запуск 40. Наша программа работает. Тогда конструкция условного оператора будет выглядеть так:

if *условие*:

 *обычный код*

И тут мы видим одну особенность: для выполнения необходимого кода необходимо после условия написать этот код с отступом. Таким образом, мы покажем программе, что нужно выполнять, а что нет. Обычно это несколько пробелов или для удобства используют табуляцию.

Теперь улучшим нашу программу. Сделаем её немного вежливой: добавим в конце «Хорошего дня!». Такая программа будет выглядеть так:

price = int(input(‘Введите цену мороженого: ’))

if price <= 50:

 print(‘Мороженое можно купить.’)

print(‘Хорошего дня!’)

Обратите внимание, что второй print без отступа. Это говорит программе, что эта строчка выполняется, не зависимо от условия.

Теперь рассмотрим такой пример. Отец с сыном играет в игру угадай число. Отец загадывает, а сын старается отгадать. Тогда такой код будет выглядеть следующим образом:

son = int(input(‘Введите число: ’))

father = 5

if son = father:

 print(‘Угадал!’)

Запускаем программу… И видимо, что программа выдает ошибку. Ошибка синтаксиса. Ошибка заключается в том, что знак равно, как мы уже знаем, это оператор присваивания. А в условии мы хотим произвести сравнение. Как же быть? На самом деле в python’е для этого существует отдельно оператор сравнения «==». Давайте тогда применим наши новые знания:

son = int(input(‘Введите число: ’))

father = 5

if son == father:

 print(‘Угадал!’)

Запускаем программу еще раз. И теперь она запускается и даже работает.

А как выглядит тогда не равно? А вот так:

son = int(input(‘Введите число: ’))

father = 5

if son != father:

 print(‘Не угадал!’)

Для сравнения могут использоваться следующие операторы сравнения: «>», «<», «>=», «<=», «==», «!=».

## Полная форма условного оператора

На прошлом занятии мы разбирали различные задачи и могли уже написать программу с выбором действий. Мы сделать программу, если условие выполняется, то выполняется одно действие, а потом – если условие не выполняется, то выполняется другое действие. Давайте вспомним:

son = int(input(‘Введите число: ’))

father = 5

if son == father:

 print(‘Угадал!’)

if son != father:

 print(‘Не угадал!’)

На самом деле в программировании очень часто возникает подобная задача. Для этого в python’е есть более полная конструкция оператора if. Сначала формализуем на нашем языке это условие: если загаданное число сына равно загаданному числу отца, то вывести на экран «Угадал!», иначе (на английском языке – else) вывести на экран «Не угадал!». Такая программа будет выглядеть следующим образом:

son = int(input(‘Введите число: ’))

father = 5

if son == father:

 print(‘Угадал!’)

else:

 print(‘Не угадал!’)

Вот так просто будет выглядеть наша программа.

Для закрепления давайте решим следующую задачу:

Мы являемся владельцем фруктовой лавки. Нам поступают предложение о продаже яблок и апельсинов, за продажу которых нам заплатят соответствующие деньги. Однако, чтобы заработать мы анализируем сумму дохода, и сделка будет выгодной тогда, когда сумма доходов с яблок и апельсинов превышает 1000 рублей. Нам надо написать программу, в которой пользователь вводит сумму дохода от яблок, а потом от апельсинов. Дальше программа выводит на экран «Предложение выгодное!» если доход больше 1000, иначе выводит «Стоит отказаться от такого предложения». Такая программа будет выглядеть так:

apple = int(input(‘Введите доход от яблок: ’))

orange = int(input(‘Введите доход от апельсинов: ’))

if apple + orange > 1000:

 print(‘Предложение выгодное!’)

else:

 print(‘Стоит отказаться от такого предложения’)

Запустим и протестируем программу. Вот и вся задача. Я думаю, не сложно.

Попробуем улучшить нашу программу с угадыванием числа. Для этого попробуем написать программу, которая не только сообщает угадал или не угадал, но и подсказывает меньше загаданное число или больше.

son = int(input(‘Введите число: ’))

father = 5

if son == father:

 print(‘Угадал!’)

else:

 if son > father:

 print(‘Меньше’)

 else:

 print(‘Больше’)

На самом деле этот код можно ещё сократить:

son = int(input(‘Введите число: ’))

father = 5

if son == father:

 print(‘Угадал!’)

else if son > father:

 print(‘Меньше’)

else:

 print(‘Больше’)

И это пока что всё.