

Программирование на Python

Урок №2

План урока:

1. Отличие чисел и строк
2. Истина и ложь
3. Что такое “условие” и где оно применяется
4. Сравнение чисел
5. Программа авторизации

Результат:

```
password = "my_password"
name = "Илья"
print("Введите имя пользователя")
user_name = input()
print("Введите пароль")
user_password = input()

if name == user_name:
    if password == user_password:
        print("Вход выполнен")
    else:
        print("Вход запрещен, неправильный пароль")
else:
    print("Вход запрещен, неправильное имя")

print("Конец работы программы")
```

Отличие чисел и строк:

На прошлом занятии мы познакомились с понятием переменная. Как можно было заметить, все, что мы складывали в переменные, было выделено кавычками. Так в Python обозначаются строки. Что же такое строка? Строкой в программировании называется набор любых символов, выделенных кавычками. К примеру, имя, по которому компьютер здоровается с нами, для компьютера является просто набором символов в определенном порядке. Компьютер не видит существенной разницы между словами “дом” и “бежать”. Более того, для компьютера даже “55” и “0” являются просто набором символов, а не числом. Для того, чтобы ввести в компьютер именно число, нам необходимо записать его без кавычек. Давайте посмотрим, чем строка “55” отличается от числа 55 и попробуем сложить строки “55” и “5” и числа 55 и 5.

```
stroka1 = "55"
stroka2 = "5"
summa_strok = stroka1 + stroka2

chislo1 = 55
chislo2 = 5
summa_chisel = chislo1 + chislo2

print(summa_strok)
print(summa_chisel)
```

Сумма строк это просто их соединение. А сумма чисел это арифметическая операция. Если попробовать сложить строку и число, то получится ошибка. То есть, если мы пишем число без кавычек, то получим полноценное число, которые можно складывать, делить, умножать и так далее. А если с кавычками, то получится строка, которую можно только соединить с другой строкой. Но тогда возникает вопрос, а что нам делать, если мы хотим, чтобы пользователь ввел число, а не строку, как в случае с именами. Для этого используется команда **int(input())**. Пользователю не надо самому ставить кавычки. При использовании **int(input())** компьютер воспримет введенную информацию как число, а при использовании **input()** как строку.

Истина и ложь:

Перед тем, как начать изучать условия, важно разобраться в таких терминах как “истина” и “ложь”. Так как по многим вопросам не может быть однозначного ответа истина это или ложь, то решено было использовать математику, как область, в которой истина и ложь определены всегда однозначно. Поэтому компьютер может определять истинность утверждения только математических выражений. Например, выражения $1 = 1$, $10 > 0$, $6 < 7$ являются истиной, а $0 = 5$, $1 > 2$ и $4 < 3$ ложью.

Что такое “условие” и где оно применяется:

Условия часто помогают нам в реальных ситуациях, когда нам необходимо сделать универсальный путь решения проблемы. К примеру, мама дала 100 рублей и отправила в магазин купить хлеб, а если денег хватит, то и молоко. Как мы поступим в такой ситуации? Купим хлеб, сравним оставшееся количество денег с ценой молока и определим — можем мы его купить или нет. Если условие является верным, то мы выполняем определенные действия (в нашем случае покупаем молоко), если же нет, то делаем иначе (не покупаем молоко и идем домой). Главное, это сформулировать наши требования так, чтобы компьютер их понял правильно.

Сравнение чисел:

Давайте попробуем написать программу, которая будет определять — хватит ли 100 рублей на хлеб и молоко. Для начала, нам необходимо узнать цену хлеба и молока.

```
print("Введите цену хлеба")
hleb = int(input())
print("Введите цену молока")
moloko = int(input())
```

Теперь нам надо добавить сюда наше условие. Для этого нам понадобится “компьютерное если”.

```
if (100 - hleb) > moloko:
    print("Покупаем молоко")
else:
    print("На молоко не хватит")
```

Ключевое слово **if**, которые выделено оранжевым цветом, переводится с английского как “если”. Далее идет наше условие на сдачу со ста рублей, которая должна быть больше, чем цена на молоко. Двоеточие в конце можно понимать как слово “делаем”. Далее идет следствие нашего условия. Если сдача больше, чем цена на молоко, то печатаем “Покупаем молоко”. Потом идет ключевое слово **else**. Оно переводится с английского как “иначе” и опять же снова двоеточие, которые мы понимаем как слово “делаем”. И если условие оказалось ложным, то есть, сдача меньше, чем цена молока, то печатаем “На молоко не хватит”. Так же, строчки которые идут после **if** и **else** сдвинуты на Tab. Это необходимо делать, чтобы показать компьютеру какие строки кода должны выполняться в случае (не)выполнения условия, а какие не должны.

Программа авторизации:

Для лучшего усвоения условий и сдвигов в Python, попробуем написать программу, которой мы пользуемся чуть ли не ежедневно. А именно программу ввода пароля при запуске чего-либо. Итак, первое что нам нужно сделать это разобраться в самой программе

авторизации. Что она делает и как. Во-первых, нам нужно создать переменную, в которой мы будем хранить правильный пароль. Назовем ее **password**. Эта переменная должна быть строкой, так как в пароле мы можем использовать не только цифры, но и буквы. Во-вторых, попросим пользователя ввести пароль и сложим его в переменную **user_password**

```
password = "my_password"
print("Введите пароль")
user_password = input()
```

Вариант некомпьютерной активности: каждый из ребят использует для пароля один из терминов программирования, изученных до этого. Затем ученики делятся на пары и описывают друг другу термины, не называя сами термины. Затем они должны понять что за термин им загадали и ввести правильный пароль в программе человека, который был в их паре. Теперь нам надо узнать совпадает ли введенный пользователем пароль и пароль, который хранится у нас. Если пароль введен верно, напишем "Вход выполнен". а иначе "Вход запрещен".

```
password = "my_password"
print("Введите пароль")
user_password = input()

if password == user_password:
    print("Вход выполнен")
else:
    print("Вход запрещен")
```

Тут можно заметить одну особенность, проверка на равенство в условии записывается не привычным нам один знаком равенства, а двумя. Это сделано для того, чтобы не путать сравнение и складывание значения в переменную. Попробуем внести в нашу программу немного новых функций. Пусть в конце работы программа выводит на экран сообщение "Конец работы программы".

```
password = "my_password"
print("Введите пароль")
user_password = input()

if password == user_password:
    print("Вход выполнен")
else:
    print("Вход запрещен")
print("Конец работы программы")
```

Так как строка “Конец работы программы” должна выводиться в любом случае, то она не сдвинута на Tab и потому выполнится в любом случае, а не только при срабатывании “иначе”. Можно сдвинуть последнюю строчку на Tab и посмотреть что изменится. Теперь введем в нашу программу авторизацию не только по паролю, но и по имени пользователя. Для этого необходимо создать новую переменную **name** и положить туда имя нашего пользователя. И перед вводом пароля попросить ввести имя пользователя и положить его в переменную **user_name**.

```
password = "my_password"
name = "Илья"
print("Введите имя пользователя")
user_name = input()
print("Введите пароль")
user_password = input()

if password == user_password:
    print("Вход выполнен")
else:
    print("Вход запрещен")
print("Конец работы программы")
```

Теперь нам необходимо срабатывание сразу двух условий — правильного имени И пароля. Самое главное не запутаться в отступах.


```

password = "my_password"
name = "Илья"
print("Введите имя пользователя")
user_name = input()
print("Введите пароль")
user_password = input()

if name == user_name:
    if password == user_password:
        print("Вход выполнен")
    else:
        print("Вход запрещен")
else:
    print("Вход запрещен")

print("Конец работы программы")

```

Теперь поясним пользователю в каком конкретно месте он ошибся при вводе имени пользователя и пароля.

```

password = "my_password"
name = "Илья"
print("Введите имя пользователя")
user_name = input()
print("Введите пароль")
user_password = input()

if name == user_name:
    if password == user_password:
        print("Вход выполнен")
    else:
        print("Вход запрещен, неправильный пароль")
else:
    print("Вход запрещен, неправильное имя")

print("Конец работы программы")

```

Задания:

1. Переписать программу так, чтобы пароль состоял из чисел(без кавычек).
2. Переписать программу так, чтобы в случае ввода неправильного пароля компьютер кроме фразы "Вход запрещен, неправильный пароль" писал фразу "Попробуйте еще раз".

3. Переписать программу так, чтобы компьютер в случае неверного имени кроме фразы “Вход запрещен, неправильное имя” писал фразу “Необходимо попробовать еще”.
4. Переписать программу так, что перед(!) фразой компьютера “Конец работы программы” было выведено сообщение “Заканчиваю программу...”
5. *Переписать программу так, чтобы после фраз “Попробуйте еще раз” и “Необходимо попробовать еще” пользователь снова вводил имя и пароль, без перезапуска программы.

Тайминг:

Тема	Время с начала занятия, мин
Отличие чисел и строк	10
Истина и ложь	15
Что такое “условие” и где оно применяется	35
Сравнение чисел	40
Программа авторизации	80
Задания	85
Контрольные вопросы	90

Контрольные вопросы:

1. Чем строка отличается от числа?
2. Можно ли сложить строку и число?
3. Что такое истина и ложь для компьютера?
4. Что такое условие?
5. Когда выполняется блок иначе?
6. Правильно ли было введено имя пользователя в последней программе, если компьютер выдал сообщение “Вход запрещен, неправильный пароль”?
7. Правильно ли был введен пароль в последней программе, если компьютер выдал сообщение “Вход запрещен, неправильное имя пользователя”? Почему мы не можем точно сказать?