

ОПЕРАТОРЫ ВЫБОРА

На этом уроке вы:

- научитесь работать с конструкциями **if** и создавать блоки кода, которые выполняются, только если условие дает Истину.
- выясните, что, расширив конструкцию **if** с помощью **elif**, можно выполнять разные блоки кода в зависимости от поставленных условий.
- узнали, как использовать ключевое слово **else**, выполняя команды, если ни одно из предыдущих условий не дало Истину.

В конце урока в вашей папке **Lesson7** должны находиться следующие папки с работающим кодом:

conditions.py

password.py

if_age.py

your_age.py

if.py

else_if.py

***money.py**

***old_enough.py**

***PolygonOrRosette.py**

Операторы управления программой

Операторы управления программой – это средства, с помощью которых можно изменять порядок выполнения программы. Они позволяют выполнить ветвление в программе, циклическое повторение нескольких инструкций, передачу управления в нужное место программы.

Язык Python имеет три категории операторов управления программой:

- 1) оператор выбора – это **if** в нескольких вариантах;
- 2) операторы цикла – это **for** и **while**;
- 3) операторы перехода – это **break** и **continue**.

Во многих операторах управления используются так называемые условия, выполнение которых проверяется, что позволяет выбирать возможное продолжение хода выполнения программы.

В обычном языке, если условие выполняется, мы говорим "да", если не выполняется, то "нет". В языках программирования вместо "да" используется термин "истина", а вместо "нет" – термин "ложь".

Проверка условий на "истину" и "ложь" используется в языке Python настолько часто, что для этого даже введен специальный тип данных – логический, который содержит всего два значения: **True** и **False**. Если условие "истинно", то его значение равно **True**, а если "ложно" – то **False**. Значения **True** и **False** являются ключевыми, то есть predetermined словами языка Python, их надо обязательно записывать с заглавной буквы.

Операторы сравнения

Часто в программах условия создаются при сравнении величин, для чего в языке Python существуют особые *операторы сравнения*. Эти операторы показаны в таблице. Некоторые из этих операторов, а также примеры их применения показаны в таблице:

Оператор	Значение	Пример	Результат
==	Равно	5==2	false
		10==10	true
!=	Не равно	6!=8	true
		3!=3	false
>	Больше	8 > 3	true
		3 > 8	false
<	Меньше	5 < 1	false
		1 < 5	true
>=	Больше или равно	10>=10	true
		9>=10	false
<=	Меньше или равно	8<=20	true
		15<=3	false

Задание 1 (conditions.py).

Рассмотрим выполнение программы с операторами сравнения при использовании чисел, числовых и строчных переменных.

1. Создайте скрипт **conditions.py** в папке **Lesson10**.
2. Напишите код:

```
# Сравнение чисел
```

```
print(5 > 2)
print(5 < 2)
print(5 == 2)
print(5 != 2)
```

```
input() # Нажми Enter для продолжения
```

```
# Сравнение значений переменных
```

```
a = 5
b = 2
print(a > b)
print(a < b)
print(a == b)
print(a != b)
```

```
input( ) # Нажми Enter для продолжения
```

```
# Сравнение строк
```

```
name1 = "Миша"
name2 = "Саша"
print(name1 == name2)
print(name1 != name2)
```

3. Запустите скрипт на выполнение из окна редактора.
4. Сделайте выводы о работе операторов сравнения.

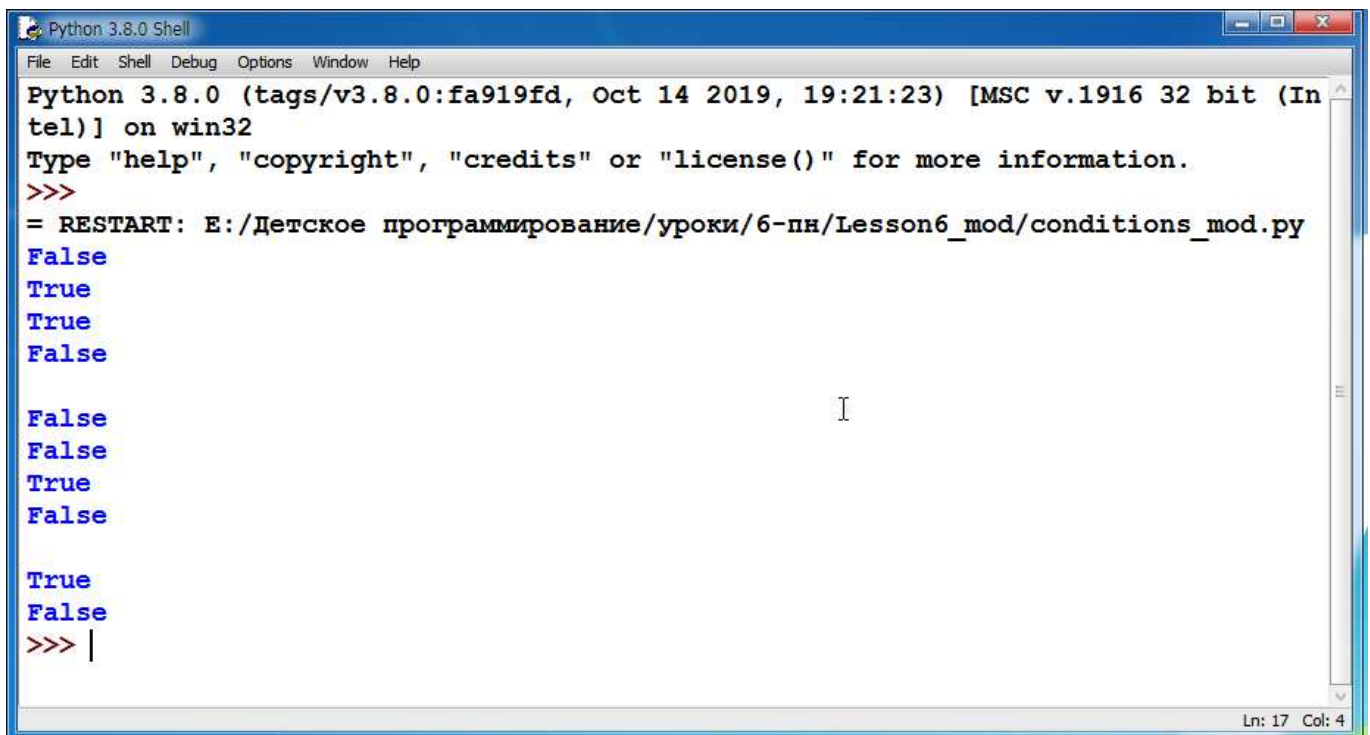
Задание 2

Измените числа и значения переменных (*операторы не изменять!*) кода программы **conditions.py**, таким образом, чтобы в оболочке результат получился таким:

```
False
True
True
False
```

```
False
False
True
False
```

```
True
False
>>>
```



The screenshot shows a Python 3.8.0 Shell window with the following content:

```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:/Детское программирование/уроки/6-пн/Lesson6_mod/conditions_mod.py
False
True
True
False

False
False
True
False

True
False
>>> |
```

The status bar at the bottom right indicates "Ln: 17 Col: 4".

Операторы выбора

Язык Python поддерживает 3 типа операторов выбора. Эти операторы позволяют проверять выполнение определенных условий и выбирать возможное продолжение вычислительного процесса. Имеются следующие конструкции этих операторов:

- 1) оператор **if** единственного выбора;
- 2) оператор **if ... else** двойного выбора;
- 3) оператор **if ... elif ... else** множественного выбора.

Оператор **if** единственного выбора

Оператор **if** единственного выбора для одной исполняемой инструкции имеет вид:

```
if условие :  
    инструкция  
.....
```

Для блока выполняемых инструкций:

```
if условие :  
    инструкция1  
    инструкция2  
.....
```

Блок – это набор сгруппированных инструкций. Все инструкции в блоке должны **обязательно** иметь одинаковый отступ. В Python принято соглашение, что отступ для инструкций в блоке должен иметь 4 пробела.

Программа вычисляет условие. Если оно истинно (**True** на языке Python), то выполняется инструкция (или блок инструкций), указанных в программе. Если условие ложно, то инструкция (или блок) не выполняются.

Рассмотрим, в качестве примера, программу "Пароль" (**password.py**), где конструкция с **if** используется для процедуры входа в компьютерную систему. Пользователь может войти лишь в том случае, если введет верный пароль.

Задание 3 password.py

1. Создай скрипт **password.py** в папке **Lesson10**.
2. Напиши код:

```
# Программа "Пароль"
print('Добро пожаловать к нам в ООО "Системы
безопасности".')
password = input("Введите пароль: ")
if password == "secret" :
    print("Доступ открыт")
input("\nНажмите Enter, чтобы выйти.")
```

3. Запусти скрипт на выполнение из окна редактора.
4. Попробуй ввести верный и неверный пароль. Сравните результаты.
5. Измените пароль – напишите его русскими буквами:

```
password == "секрет"
```

6. Проверь работу программы.
7. Сделай выводы о работе программы.

Задание 4 if_age.py

Здесь блок состоит из трех команд **print**, которые должны выполняться лишь в том случае, если условие **age > 20** окажется истинным.

Каждая строка кода в блоке начинается с отступа в четыре пробела (относительно конструкции **if**, стоящей перед блоком).

1. Создай скрипт **if_age.py** в папке **Lesson10**.
2. Напиши код:

```
age = 13
if age > 20:
    print('Как-то вы староваты!')
    print('Что вы здесь делаете?')
    print('Почему не ходите на работу?')
```

3. Запусти его на выполнение.
4. Что произошло? А ничего! Поскольку значение **age** не больше 20, Python не будет выполнять блок с командой **print()**.
5. Измените значение переменной **age** в коде так, чтобы блок текста выводился в оболочку.

Оператор **if ... else** (если ... то) двойного выбора

Оператор **if ... else** (если ... то) двойного выбора обеспечивает два варианта продолжения выполнения программы. Выбор осуществляется исходя из результата проверяемого условия.

Общий вид оператора **if ... else** двойного выбора имеет вид:

```
if условие :  
    инструкция1  
else :  
    инструкция2
```

Вместо одиночных инструкций 1 или 2 могут быть блоки инструкций:

```
if условие :  
    БлокИнструкций  
else :  
    БлокИнструкций
```

Задание 5 **your_age.py**

С помощью команды **if** можно что-то сделать, если условие не дает Истину (то есть дает Ложь). Например, если ваш возраст (**age**) равен 12, мы можем вывести на экран одно сообщение, а если не равен 12 — другое. Для этого служит конструкция

if-else,

которая работает по принципу:

«если условие дает Истину, сделай это, иначе сделай то».

1. Создай скрипт **your_age.py** в папке **Lesson10**.

2. Введи код:

```
#Спросить пользователя его возраст:
age = int(input("Сколько тебе лет? \n"))

# Задать условие выполнения программы:
if age >= 12:
    print('Вы слишком стары для моих шуток!')
else:
    print('Что нельзя съесть на завтрак?')
    print('Обед и ужин!')
```

3. Запусти его на выполнение.

4. Попробуй вводить разный возраст и посмотрите на результат.

5. Сделай выводы о работе программы.

Задание 6 *

1. Открой ранее созданный скрипт **password.py**.

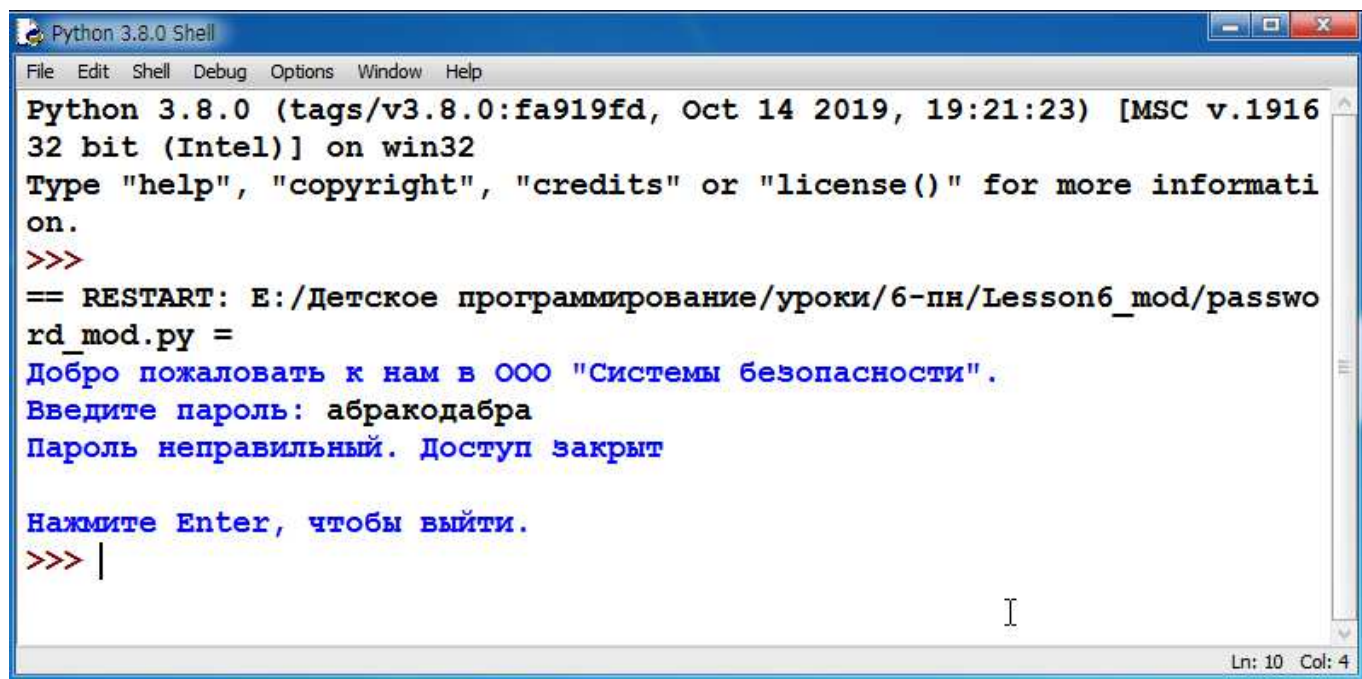
2. Произведи модернизацию программы **password.py**.

3. Введи в программу оператор **else**, с помощью которого на экран будет выводиться сообщение "**Пароль неправильный. Доступ закрыт**" при вводе неправильного пароля.

4. Как добавлять **else** посмотри в скрипте **your_age.py**. Сделайте по аналогии.

5. Запусти скрипт из окна редактора.

6. Проверь работу программы при вводе различных паролей.



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916
32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more informati
on.
>>>
== RESTART: E:/Детское программирование/уроки/6-пн/Lesson6_mod/passwo
rd_mod.py =
Добро пожаловать к нам в ООО "Системы безопасности".
Введите пароль: абракодабра
Пароль неправильный. Доступ закрыт

Нажмите Enter, чтобы выйти.
>>> |
```


Задание 7 *

1. Создай новый файл и сохраните его под именем **money.py** в папке **Lesson10**.
2. При выполнении задания ориентируйся на программу **your_age.py**
3. Напишите программу (по аналогии с предыдущей), которая будет спрашивать у пользователя:

Сколько у тебя денег?

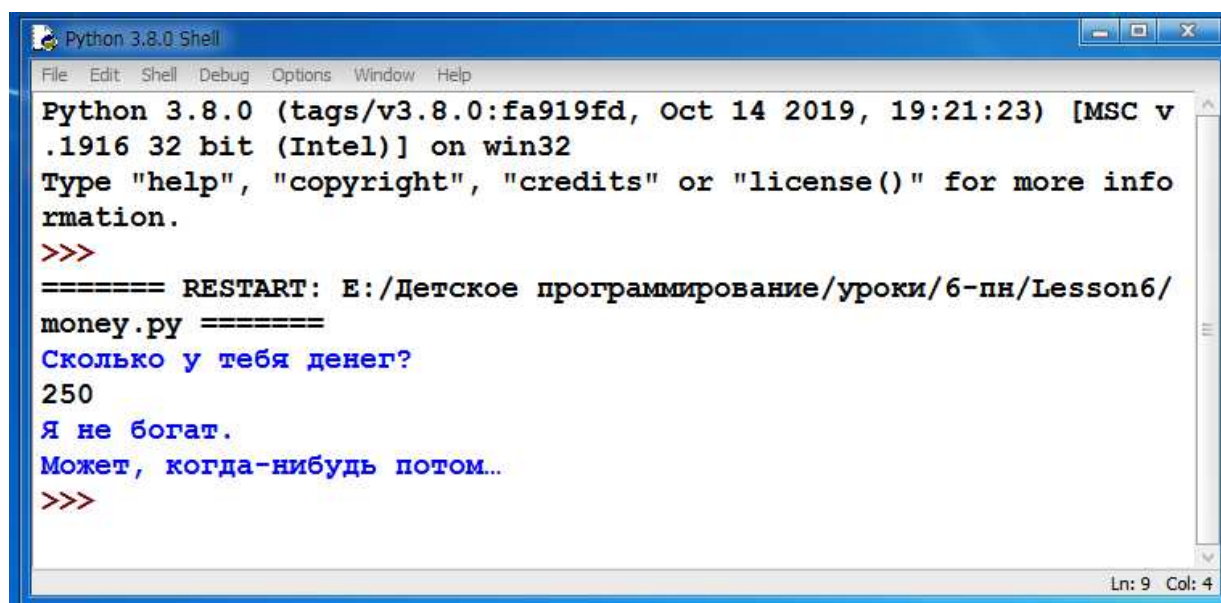
В случае, если пользователь введет число больше 1000, на печать выводит фразу:

Я богат!

а противном случае:

Я не богат!

Может, когда-нибудь потом...



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v
.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more info
rmation.
>>>
===== RESTART: E:/Детское программирование/уроки/6-пн/Lesson6/
money.py =====
Сколько у тебя денег?
250
Я не богат.
Может, когда-нибудь потом...
>>>
```

Задание 8**

1. Создай новый файл и сохраните его под именем **old_enough.py** в папке **Lesson10**.
2. Напиши программу которая спрашивает у пользователя:

Какой минимальный возраст для получения прав?

После того как вы ответите, компьютер опять задает вопрос:

Сколько вам лет?

Затем при условии, если возраст пользователя больше или равен возрасту получения прав, то компьютер выводит на печать:

Вы уже достаточно взрослый для получения прав!

Если же возраст меньше, то на печать выводится следующая фраза:

Извините, вы не сможете водить еще в течение ... лет.

Подсказка.

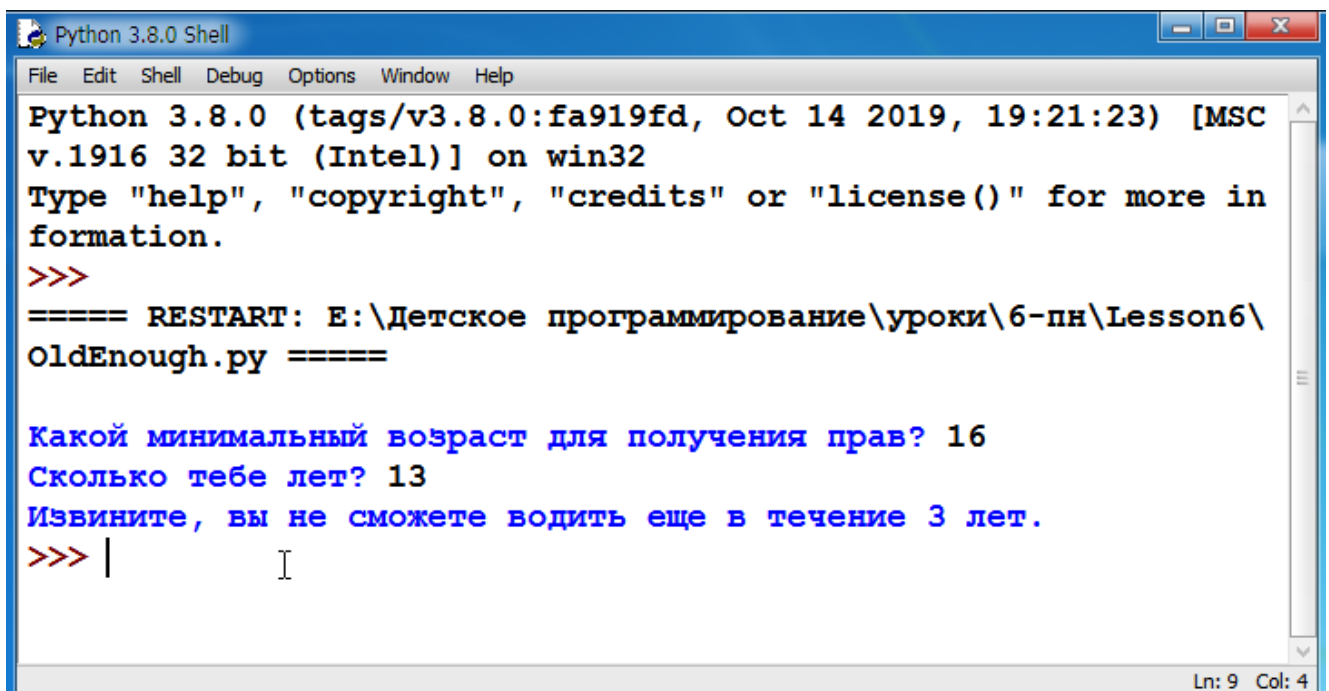
Используйте переменные:

driving_age – возраст получения прав

your_age – возраст пользователя

В блоке **else** на печать необходимо будет вывести в **print()** через запятую:

- строку **'Извините, вы не сможете водить еще в течение'**
- разность переменных **driving_age** и **your_age**
- строку **'лет'**



```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC
v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more in
formation.
>>>
===== RESTART: E:\Детское программирование\уроки\6-пн\Lesson6\
OldEnough.py =====

Какой минимальный возраст для получения прав? 16
Сколько тебе лет? 13
Извините, вы не сможете водить еще в течение 3 лет.
>>> |
```

Оператор if ... elif ... else множественного выбора

Оператор **if ... elif ... else** множественного выбора позволяет реализовать множественный выбор, исходя из результатов проверяемых условий.

Общий вид оператора **if ... elif ... else** следующий:

```
if условие1 :
    инструкция1
elif условие2 :
    инструкция2
.....
elif условиеN :
    инструкцияN
else :
    инструкция (N+1)
```

Оператор выполняет серию последовательных проверок условий до тех пор, пока не будет установлено следующее:

- 1) одно из условий в **if**-части или в **elif**-части является истиной. В этом случае выполняются соответствующие этим условиям инструкции;
- 2) ни одно из вложенных условий не является истиной. Программа выполняет инструкции в последней **else**-части, если она имеется.

Задание 9 if.py

Рассмотрим программу, которая принимает варианты чисел от пользователя и проверяет, совпадают ли они с заранее заданным числом.

1. Создай скрипт **if.py** в папке **Lesson10**.
2. Введи код:

```
number = 23
guess = int(input("Введите целое число: "))

if guess == number :
    print("Поздравляю, вы угадали!")
elif guess < number :
    print("Нет, загаданное число больше этого.")
else :
    print("Нет, загаданное число меньше этого.")

print("Завершено") # Эта последняя инструкция выполняется
                  # всегда после выполнения оператора if
```

3. Запусти скрипт из окна редактора.
4. Проверь работу программы при вводе разных чисел (больше и меньше 23).
5. Присвой переменной **number** другое значение, запусти программу и убедись, что она работает верно.

Задание 10*

Измени программу **if.py** следующим образом:

Если введенное число больше, то программа должна печатать:

Нет, загаданное число больше этого на ...

Если введенное число меньше, то программа должна печатать:

Нет, загаданное число меньше этого на ...

The screenshot shows a Python IDE with two windows. The top window displays the source code for a program that checks if a user's guess is correct, less than, or greater than a target number (23). The bottom window shows the program's execution, where the user enters 20 and 10, and the program provides feedback and completion messages.

```
if_mod.py - E:\Детское программирование\уроки\6-пн\Lesson6_mod\if_mod.py (3.8.0)
File Edit Format Run Options Window Help

number = 23
guess = int(input("Введите целое число: "))

if guess == number :
    print("Поздравляю, вы угадали!")
elif guess < number :
    print("Нет, загаданное число больше этого на 3")
else :
    print("Нет, загаданное число больше этого на 13")
print("Завершено")

Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information
>>>
===== RESTART: E:\Детское программирование\уроки\6-пн\Lesson6_mod\if_mod.py =====
Введите целое число: 20
Нет, загаданное число больше этого на 3
Завершено
>>>
===== RESTART: E:\Детское программирование\уроки\6-пн\Lesson6_mod\if_mod.py =====
Введите целое число: 10
Нет, загаданное число больше этого на 13
Завершено
>>>
```

Задание 11 else_if.py

1. Создай скрипт `else_if.py` в папке **Lesson10**.
2. Введи код:

```
#Спросить пользователя его возраст:
age = int(input("Сколько тебе лет? \n"))

# Задать условия выполнения программы:
if age == 10:
    print('Что выйдет, если клюква наденет штаны?')
    print('Брюква!')

elif age == 11:
    print('Что сказала зеленая виноградаина синей виноградине?')
    print('Дыши! Дыши!')

elif age == 12:
    print('Что сказал 0 числу 8?')
    print('Привет, ребята!')

elif age == 13:
    print('Что такое: на потолке сидит и хохочет?')
    print('Муха-хохотуха')
else:
    print('Сколько-сколько?')
```

3. Запусти скрипт его на выполнение.
4. Попробуй вводить разный возраст и посмотри на результат.

Пояснение к коду.

Переменной **age** присваивается значение, которое ввел пользователь.

Сначала делается проверка, равно ли значение переменной **age** числу 10. Соответственно, команды **print** в блоке будут выполнены, если **age** равняется 10. Если переменной **age** не равна числу 10, компьютер переходит на следующую строку, к следующей проверке, где **age** сравнивается с числом 11. Если **age** не равно 11, то компьютер идет дальше, чтобы проверить, равняется ли **age** числу 12. И так далее. Если ни одно из условий не даст истину, компьютер выполнит блок после слова **else** и напечатает: **Сколько-сколько?**

При вводе этого кода IDLE будет ставить отступы автоматически, поэтому не забывайте нажимать **Delete** или **Backspace** после блоков с командами **print**, так чтобы перед командами **if**, **elif** и **else** отступов (пробелов) не было.

Задание 12*

1. Открой скрипт **else_if.py** в папке **Lesson10**.
2. Измени программу следующим образом:

Перед условием:

```
else:  
    print('Сколько-сколько?')
```

добавь еще два условия **elif**.

Если возраст меньше 10, то пусть программа выводит на печать:

1 июня – День защиты детей.

Помните, дети: лучшая защита – это нападение.

Если возраст больше 13, то пусть программа выводит на печать:

Пропала собака, ротвейлер.

Нашедшему... царство небесное...

Рассмотрим программу, в которой пользователя просят ввести, хочет ли он, чтобы был нарисован многоугольник (треугольник, квадрат, пятиугольник и так далее) или же розетка с определенным количеством сторон или окружностей. В зависимости от выбора, сделанного пользователем (**м** для многоугольника и **р** для розетки), программа нарисует правильную фигуру.

Задание 13 PolygonOrRosette.py

1. Создай скрипт `else_if.py` в папке `Lesson10`.
2. Введи код:

```
import turtle
window = turtle.Screen( )
t= turtle.Pen( )
    # Запросить у пользователя количество сторон или
окружностей
number = int(turtle.numinput("Количество сторон или
окружностей", \
    "Сколько сторон или окружностей будет у фигуры? "))
    # Спросить у пользователя, хочет ли он отобразить
многоугольник
    # или розетку
shape = turtle.textinput("Какую фигуру вы хотите?", \
    "Введите 'м' для многоугольника или 'р' для розетки")
for k in range(number):
    if shape == 'р' :    # Пользователь выбрал розетку
        t.circle(100)
    else :    # Многоугольник по умолчанию
        t.forward(150)
        t.left(360 / number)
```

3. Запусти скрипт из окна редактора.
4. Проверь работу программы вводя разные значения.
5. Установи толщину линий пера 5 пикселей, цвет холста зеленый, а цвет линий многоугольника красный, а окружности – синий.

Пояснение.

`turtle.numinput()` – это функция для вызова диалогового окна для ввода числа.

Синтаксис: `turtle.numinput (title, prompt, default=None, minval=None, maxval=None)`

turtle.numinput () – это функция для вызова диалогового окна для ввода строки.

Синтаксис: **turtle.textinput** (**title**, **prompt**)

Параметры:

title – строка

prompt – строка

default – число (не обязательно)

minval – число (не обязательно)

maxval – число (не обязательно)

Числовой ввод должен находиться в диапазоне **minval** ... **maxval**, если они указаны.

В нашем примере их нет.

(В переводе с англ. *title* - заголовок, *prompt* –подсказка, *default* – значение по умолчанию, *minval* – минимальное значение для ввода, *maxval* – максимальное значение для ввода)

Итак, вы познакомились с тремя схожими условными конструкциями для выбора с помощью оператора **if**.

Кратким напоминанием послужит вам таблица.

Конструкция	Описание
if <условие> : БлокИнструкций	Простейшая условная конструкция. Если <условие> истинно, то БлокИнструкций выполняется, в противном случае - пропускается
if <условие> : БлокИнструкций_1 else : БлокИнструкций_2	Условная конструкция с условием else . Если <условие> истинно, выполняется БлокИнструкций_1, если ложно – БлокИнструкций_2
if <условие_1> : БлокИнструкций_1 elif <условие_2> : БлокИнструкций_2 elif <условие_N> : БлокИнструкций_N else : БлокИнструкций_N+1	Условная конструкция с дополнительными условиями elif и необязательным else в конце. Будет исполнен блок после первого же условия, которое окажется истинным. Если ни одно условие не принимает значение True, будет исполнен блок после заключительного else .