

# Алгоритм. Линейный алгоритм. Условные оператор if. Elif.

# Повторение

- Что такое переменная?
- Функция для вывода на экран какого-то текста, или значения.
- Как обозначается целый тип данных.
- Как обозначается вещественный тип данных.
- Как обозначается строковый тип данных.
- Как узнать тип данных?
- Функция для ввода данных в программу?

# Алгоритм

- Алгоритм — это последовательность команд для достижения цели.
- Пример простого линейного алгоритма с прошлого занятия. Команды выполняются последовательно, одна за одной.

```
>>> mum = 2019-1965
>>> papa = 2019-1964
>>> sister = 2019-1990
>>> me = 2019-1991
>>> print("Маме", mum, "лет")
Маме 54 лет
>>> print("Папе", papa, "лет")
Папе 55 лет
>>> print("Сестре", sister, "лет")
Сестре 29 лет
>>> print("Мне", me, "лет")
Мне 28 лет
>>> |
```



# Условный алгоритм.

- При написании программ часто приходится задавать вопросы, требующие ответа «да» или «нет», и в зависимости от этого совершать какие-то действия. Например, вы можете спросить: вам больше 20 лет? И если пользователь ответит «да», вывести сообщение: как-то вы староваты! Подобные вопросы называют условиями.
- Условный алгоритм, это последовательность команд, которая содержит в себе условие, в зависимости от выполнения которого будут выполняться те или иные дальнейшие действия.



# Логический тип данных bool.

- Когда мы задаем вопросы, т.е. условия, мы ждем ответа либо да, либо нет. В Python за такой ответ отвечает тип bool.
- Если мы вспомним другие типы данных, которые уже выучили, например тип int, то в нем мы можем хранить любые числа, т.е. любые значения. В нашем сегодняшнем случае, у нас только два варианта ответа, да или нет. Вот тип bool только эти два ответа и может хранить, и ничего больше. Только называются они немного по-другому. True (да) или False (нет). Правда или ложь.

# Операторы сравнения.

- Когда мы отвечаем на вопрос да или нет (правда или ложь) мы как-бы сравниваем результат (ответ) с тем что для нас является правдой, истинным результатом. В Python сравнивать нам помогают следующие простые операторы сравнения

Оператор	Значение
==	Равно
!=	Не равно
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно

# Оператор If.

- В программах на Python условия и ответы обрабатываются с помощью условной конструкции if. Эти конструкции используют условные выражения и в зависимости от их значения направляют выполнение программы по одному из путей.
- В самом простом виде после ключевого слова if идет логическое выражение. И если это логическое выражение возвращает True, то выполняется последующий блок инструкций, каждая из которых должна начинаться с новой строки и должна иметь отступы от начала строки:

# Оператор If.

Если наше условие возвращает

```
True: = 22
age = 22
if age > 21:
    print("Доступ разрешен")
print("Завершение работы")
```

```
Доступ разрешен
Завершение работы
```

Если наше условие возвращает

```
False: = 20
age = 20
if age > 21:
    print("Доступ разрешен")
print("Завершение работы")
```

```
Завершение работы
```

Обратите внимание на отступы. Если мы поставим второй print() на одном уровне с первым, у нас будет другой результат.

```
age = 20
if age > 21:
    print("Доступ разрешен")
print("Завершение работы")
```

В случае, если условие возвратит False, мы ничего не увидим на экране. Почему так произошло?





# Блоки программы

- В нашем случае, если условие истинно, True, мы выполняем одно действие.
- А что если наше условие `if age > 21`: истинно, и нам понадобится выполнить не одно действие, а несколько. Для этого мы группируем все эти команды в один блок.
- **Блок** — это набор сгруппированных программных конструкций (команд). В Python они выделяются отступами.

строка кода  
строка кода  
строка кода

строка кода  
строка кода  
строка кода

строка кода  
строка кода  
строка кода

строка кода  
строка кода

блок 3

блок 2

блок 1



## Условие возвращает

**False:**

```
age = 20
if age > 21:
    print("Доступ разрешен")
    print("Завершение работы конструкции if")
print("Действия после конструкции if")
print("Мы проверили условие, и дальше последовательно продолжаем выполнять программу")
```

Действия после конструкции if

Мы проверили условие, и дальше последовательно продолжаем выполнять программу

## Условие возвращает

**True:**

```
age = 22
if age > 21:
    print("Доступ разрешен")
    print("Завершение работы конструкции if")
print("Действия после конструкции if")
print("Мы проверили условие, и дальше последовательно продолжаем выполнять программу")
```

Доступ разрешен

Завершение работы конструкции if

Действия после конструкции if

Мы проверили условие, и дальше последовательно продолжаем выполнять программу

Как видим, когда наше условие возвращает True, мы выполнили действия и в блоке if и в блоке после него. В случае, когда условие вернуло нам False, мы выполнили только действия после блока if.



# Оператор If-Else, Elif.

- Если вдруг нам надо определить альтернативное решение на тот случай, если условное выражение возвратит False, то мы можем использовать блок else, что выведет на экран?:
- Если выражение `age > 21` возвращает True, то выполняется блок if, иначе выполняется блок else.

```
age = 18
if age > 21:
    print("Доступ разрешен")
else:
    print("Доступ запрещен")
```

```
age = 18
if age > 21:
    print("Доступ разрешен")
else:
    print("Доступ запрещен")
print("Завершение работы")
```

А если так, что выведет?



# Дополнительный блок Elif.

- Если необходимо ввести несколько альтернативных условий, то можно использовать дополнительные блоки elif, после которого идет блок инструкций.

```
age = 18
if age > 21:
    print("Доступ разрешен")
elif age >= 18:
    print("Доступ частично разрешен")
else:
    print("Доступ запрещен")
```



# Вложенные условия.

- Конструкция `if` в свою очередь сама может иметь вложенные конструкции `if`.
- Стоит учитывать, что вложенные выражения `if` также должны начинаться с отступов, а инструкции во вложенных конструкциях также должны иметь отступы. Отступы, расставленные не должным образом, могут изменить логику программы.

```
age = 18
if age >= 18:
    print("Больше 17")
    if age > 21:
        print("Больше 21")
    else:
        print("От 18 до 21")
```



# Практика

- Условная конструкция, которая печатает «Вы слишком стары для моих шуток!», если возраст пользователя (age) больше 10 лет

```
age = 10
if age > 10:
    print('Вы слишком стары для моих шуток!')
```

- Что выведет? Почему так?
- Как изменить программу чтобы при значении age=10 нам все же вывелся текст.



# Практика

- С клавиатуры считываем число, и в зависимости от того какое число было введено выводим в каком диапазоне это число. Не забываем, что функция ввода текста (что за она?) возвращает нам текст. И для того чтобы иметь возможность сравнивать его с другими числами, нам необходимо его преобразовать.

```
Введите любое число: 36
```

```
Вы ввели число больше 30 но меньше 40
```

```
Введите любое число: -6
```

```
Вы ввели число меньше 0
```

```
Введите любое число: 125
```

```
Вы ввели число больше 100
```





# Решение

```
numberStr = input("Введите любое число: ")
number = int(numberStr)
if number > 50:
    if number > 100:
        print("Вы ввели число больше 100")
    else:
        print("Вы ввели число больше 50 но меньше 100")
elif number > 30:
    if number > 40:
        print("Вы ввели число больше 40 но меньше 50")
    else:
        print("Вы ввели число больше 30 но меньше 40")
elif number > 10:
    if number > 20:
        print("Вы ввели число больше 20 но меньше 30")
    else:
        print("Вы ввели число больше 10 но меньше 20")
else:
    if number < 0:
        print("Вы ввели число меньше 0")
    else:
        print("Вы ввели число больше 0 но меньше 10")
```



# ДЗ

- Определить существует ли треугольник по трем сторонам. Сумма любых двух сторон должна быть больше третьей.
- Определить количество дней в году. В обычном году 365 дней, а в високосном - 366. Каждый четвертый год является високосным, кроме столетий, которые не делятся нацело на 400. Например, 2000 год високосный, а 1900 - нет, т.к. не делится нацело на 400. Если год не делится нацело на 4, значит сразу можно сделать вывод, что он не является високосным. Если все же делится, надо исключить столетия за исключением тех, которые нацело делятся на 400.
- Какое число: однозначное, двухзначное, трехзначное или более, положительное или отрицательное?  $n = 345$  - Положительное трехзначное или более число  $n = -23$  - Отрицательное двузначное число

