

## УРОК 38. REVISION

|  |    |
|--|----|
| ОСНОВНЫЕ ТИПЫ ДАННЫХ И МАТЕМАТИЧЕСКИЕ ОПЕРАЦИИ | 2  |
| IF И WHILE                                     | 4  |
| ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ                        | 5  |
| МЕТОДЫ ДЛЯ РАБОТЫ СО СТРОКАМИ                  | 6  |
| FOR И ФУНКЦИИ                                  | 8  |
| ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ                        | 9  |
| ПРАКТИЧЕСКАЯ РАБОТА                            | 10 |
| ПОЛЕЗНЫЕ МАТЕРИАЛЫ                             | 11 |



# ОСНОВНЫЕ ТИПЫ ДАННЫХ И МАТЕМАТИЧЕСКИЕ ОПЕРАЦИИ

Основные типы данных в Python:

| Тип данных         | Значение  | Определение в Python | Вариант использования   |
|--------------------|---|----------------------|---|
| Целые числа        | -3, -2, -1, 0, 1, 2, 3                                  | int                  | <code>a = int(input())</code>                                       |
| Вещественные числа | -1.5, -1.1, 0.6, 1.7                                    | float                | <code>a = float(input())</code>                                     |
| Булевы значения    | True, False   | True, False          | <code>flag = True</code>  |
| Строка             | 'abracadabra'   | str                  | <code>a = str(5)</code>   |
| Список             | [1, 2, 3], ['a', 'b', 'c']                              | list                 | <code>a = list(('a', 'b', 'c'))</code>                              |
| Кортеж             | ('red', 'blue', 'green')                                | tuple                | <code>a = tuple(('red', 'blue', 'green'))</code>                    |
| Множество          | {'black', 'blue', 'white'}, {1, 3, 9, 7}                | set                  | <code>a = set(('black', 'blue', 'white'))</code>                    |
| Словарь            | {'color': 'red', 'model': 'VC6', 'dimensions': '30×50'} | dict                 | <code>a = dict(color='red', model='VC6', dimensions='30×50')</code> |



Основные математические операции в Python:

| Оператор | Операция  | Пример              | Результат         |
|----------|-----------|---------------------|-------------------|
| +        | Сложение  | 478 + 32<br>"a"+"b" | 510<br>"ab"       |
| -        | Вычитание | 999 – 135           | 864               |
| *        | Умножение | 123 * 321<br>"hi"*3 | 39483<br>"hihihi" |
| /        | Деление   | 39 / 2              | 19.5              |



## IF И WHILE

Полный синтаксис условного оператора if:

```
Python
if условие1:
    # выполняемые действия, если условие1 истинно
elif условие2:
    # выполняемые действия, если условие2 истинно
else:
    # выполняемые действия, если все условия ложны
```



**Цикл while в Python - это цикл с предусловием, то есть он выполняет блок кода, пока условие истинно.**

Формат цикла while выглядит следующим образом:

```
Python
while условие:
    # выполняемые действия
    # возможно использование break и continue
else:
    # выполняемые действия, если цикл завершился без break
```



## ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Объясните, что происходит в этом фрагменте программы. Перечислите ситуации, когда цикл продолжает выполняться, а когда завершается:

Python

```
list_of_tables = ["users", "product", "sales"]
str_of_tables = ', '.join(list_of_tables)
tables=["empty"]

while not set(tables).issubset(list_of_tables):
    tables = input(f"Введите таблицу или список таблиц из набора
{str_of_tables}: ").split()
    if set(tables) == {"users", "product"}:
        print("Эти таблицы не связаны. Сделайте другой выбор")
        tables=["empty"]
```



## МЕТОДЫ ДЛЯ РАБОТЫ СО СТРОКАМИ

| Функция или метод       | Назначение                                       |
|-------------------------|--|
| <b>S.split</b> (символ) | Разбиение строки по разделителю                  |
| <b>S.isdigit</b> ()     | Состоит ли строка из цифр                        |
| <b>S.isalpha</b> ()     | Состоит ли строка из букв                        |
| <b>S.islower</b> ()     | Состоит ли строка из символов в нижнем регистре  |
| <b>S.isupper</b> ()     | Состоит ли строка из символов в верхнем регистре |
| <b>S.upper</b> ()       | Преобразование строки к верхнему регистру        |
| <b>S.lower</b> ()       | Преобразование строки к нижнему регистру         |
| <b>S.join</b> (список)  | Сборка строки из списка с разделителем S         |



**S.strip**([chars])

Удаление пробельных символов в начале и в конце строки



## FOR И ФУНКЦИИ

Цикл `for` в Python используется для выполнения повторяющихся операций над элементами последовательности. Он позволяет пройти по всем элементам списка, строки, кортежа или другой итерируемой структуры данных. Каждая итерация цикла представляет собой обработку одного элемента.

Синтаксис:

```
Python
for i in итератор:
    # выполняемые действия
    # возможно использование break и continue
else:
    # выполняемые действия, если цикл завершился без break
```

Функция - это блок кода, который выполняет определенную задачу и может быть многократно использован в программе.

Синтаксис:

```
Python
def имя_функции(аргументы):
    # выполняемые действия
    return возвращаемые значения
```





## ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Объясните, что происходит в этом фрагменте программы.

Python

```
def input_columns(table):
    columns = {"users" : ["id", "name", "age"],
               "product": ["pid", "prod", "quantity"],
               "sales": ["sid", "pid", "id"]}
    field = input(f"Выберите одно из полей {columns[table]} этой таблицы или
введите * для выбора всех: ").strip()
    if field not in columns[table] and field != "*":
        print("Введено некорректное значение. Будет использовано значение *")
        field = "*"
    return "SELECT " + field + " "
```



## ПРАКТИЧЕСКАЯ РАБОТА

В базе данных ich\_edit три таблицы. Users с полями (id, name, age), Products с полями (pid, prod, quantity) и Sales с полями (sid, id, pid). Написать мини-интерфейс к базе данных, который умеет выполнять разные команды.

1. Выбрать таблицу для запроса. Предусмотреть возможность выбрать несколько таблиц. Вывести результат их соединения, если это возможно, или сообщение об ошибке.
2. Выбрать одно поле из выбранной таблицы и искомое значение этого поля. Вывести все подходящие строки



## ПОЛЕЗНЫЕ МАТЕРИАЛЫ

1. [Python, введение в БД](#)
2. [Руководство по работе с базами данных в Python](#)