

# Методичка по Python.

Информатика Умскул ([https://vk.com/umsch\\_i](https://vk.com/umsch_i))

Телега и все другие соцсети @infa\_vikusya - [https://t.me/infa\\_vikusya](https://t.me/infa_vikusya)

## Основные типы данных в Python:

**int()** — целые числа;  
**float()** — числа с плавающей точкой;  
**str()** — строки;  
**list()** — списки;  
**dict()** — словари;  
**tuple()** — кортежи;  
**set()** — множества.

## Условные операторы в Python:

Существует 3 условных оператора: if, elif, else.

if — оператор, являющийся началом условной конструкции, который может вернуть два значения: True или False.

Операторов elif для одного блока условного оператора может быть несколько, а может не быть совсем.

Последним идёт оператор else, который не содержит условия, а выполняется, только если ни одно из предыдущих условий в if и elif не выполнилось. Оператор else не является обязательным.

## Циклы в Python:

Цикл for всегда используется в сочетании с итерируемым объектом, таким как список или диапазон. Циклы for используются, когда у вас есть блок кода, который вы хотите повторить фиксированное количество раз.

Цикл for от 0 до 2 включительно, следовательно, выполняется 3 раза.

```
for x in range(0, 3):  
    print("Hello world!")
```

Цикл while (“пока”) позволяет выполнять одну и ту же последовательность действий, пока проверяемое условие истинно.

Цикл While от 1 до бесконечности, поэтому выполняется вечно.

```
x = 1  
while True:  
    print(x)  
    x += 1
```

## Методы в Python, которые пригодятся для ЕГЭ:

**str.split(sep = None, maxsplit = -1)** — метод, который возвращает список слов (подстрок) в строке, используя sep в качестве разделителя строки. Если задан maxsplit, то выполняется не более maxsplit разбиений, таким образом, список будет иметь не более maxsplit + 1 элементов. Если maxsplit не указан или равен -1, то метод разделяет строку максимальное количество раз.

**str.index(element)** — метод, который возвращает индекс первого появления element в строке.

**str.rindex(element)** — метод, который возвращает индекс последнего появления element в строке.

`str.replace(old, new, count)` — метод, который возвращает копию строки, в которой значения старой строки были заменены новым значением `count` раз.

`list.append(element)` — метод, который добавляет `element` в конец списка.

`list.remove(element)` — метод, который удаляет первый `element`, появляющийся в списке.

`list.sort(reverse = False)` — метод, который сортирует список элементов. Параметр `reverse` необязателен и изначально равен `False`, поэтому он сортирует по возрастанию.

`.count(element)` — метод, который возвращает количество повторений элемента `element` в списке или строке.

### **Функции в Python, которые пригодятся для ЕГЭ:**

`len(object)` — функция, которая возвращает длину объекта `object`;

`sum(object, start)` — функция, которая возвращает сумму элементов объекта `object`, начиная со значения `start`. Данная функция ожидает, что все элементы `object` являются числами;

`bin(object)` — функция, которая возвращает число в двоичной записи;

`int(object, base = 10)` — функция возвращает целочисленный объект из указанного ввода. Возвращаемый объект `int` всегда будет с основанием 10.

`sorted(reverse = False)` — функция, которая возвращает новый отсортированный список. По умолчанию она сортирует по возрастанию.

### **Библиотеки в Python, которые пригодятся для ЕГЭ:**

*itertools* — библиотека, которая хранит в себе полезные итераторы.

Примеры функций:

`product(iterable, repeat = 1)` — функция, которая возвращает итератор, хранящий в себе декартово произведение, эквивалентное вложенному циклу `for`.

`permutations(iterable, r)` — функция, которая возвращает итератор, хранящий в себе перестановки длиной `r` из `iterable`.

`combinations(iterable, r)` — функция, которая возвращает итератор, хранящий в себе комбинации длиной `r` из `iterable` без повторяющихся элементов.

*functools* — библиотека, которая хранит в себе высокие функции.

Примеры функций и декораторов:

`@lru_cache`

`reduce()`

*math* — библиотека, которая предоставляет обширный функционал для работы с числами.

Примеры функций:

`ceil(x)` — округление до ближайшего большего числа.

`sqrt(x)` - квадратный корень из `x`.