

Ссылка на урок:

<https://coreapp.ai/app/player/lesson/66fa23d2d280b778ee9e0c53> (для учеников).

Занятие 3.

Тема. Инструкции ЯП Python. Первая программа

1. Теоретическая часть

Для того, чтобы программа могла работать и выдавать ожидаемый результат, мы должны предоставить интерпретатору Python чёткие и пошаговые указания. Это делается с помощью инструкций.

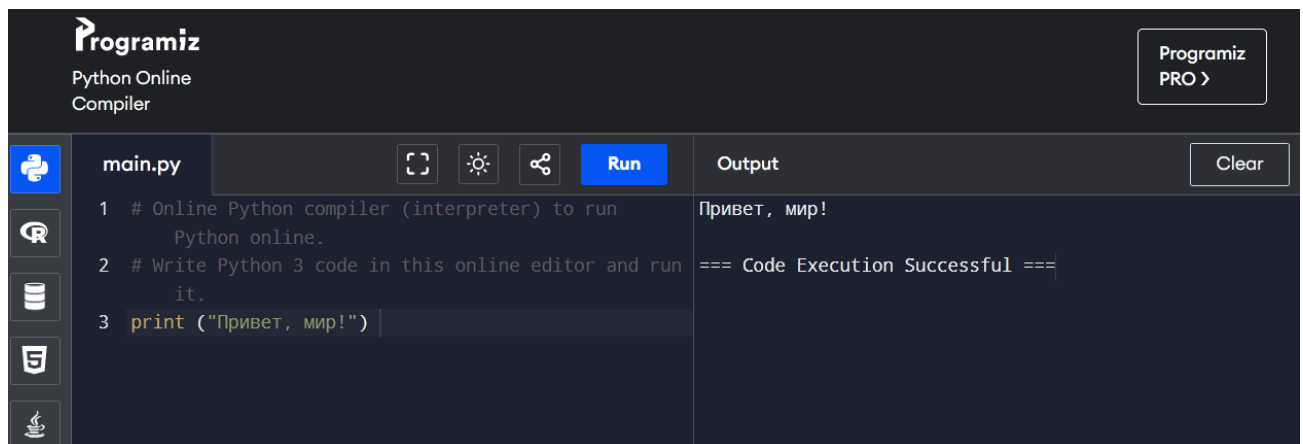
Инструкция – это код, содержащий в себе логическую последовательность действий над какими-либо объектами (величинами), выполнение которой приведёт к желаемому результату.

Величиной может считаться, например, число, символ, строка, таблица, графический объект. Величины, с которыми работает компьютер принято называть **данными**.

Инструкции также можно называть **операторами**. По сути, это команды, управляющие процессом выполнения программы. Любая компьютерная программа представляет собой последовательность операторов. Инструкции (операторы) могут быть простыми или составными.

Создадим свою первую программу, выдающей надпись: «Привет, мир!»

Для этого нужно зайти на сайт <https://www.programiz.com/python-programming/online-compiler/> (обучающая он-лайн платформа для начинающих программистов) и в левой части окна набрать `print ("Привет, мир!")`, щ. ЛКМ по кн. Run и в правой части окна (окно вывода) появится результат: надпись «Привет, мир!».



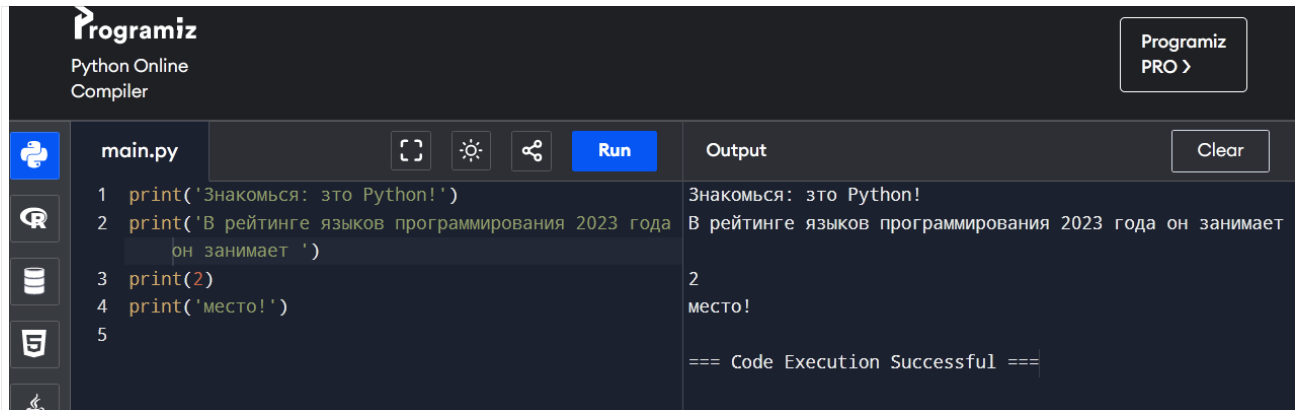
Ну конечно-же, серьёзные программы не состоят из одной строки. Сгенерированный человеком код может содержать в себе до миллиона строк в одном модуле! И, чтобы программы были «читабельными» для человека, есть определённые правила их составления. Вы постепенно с ними познакомитесь.

А вот и правило первое:

Каждая инструкция должна начинаться с новой строки.

Посмотрим на пример кода с четырьмя инструкциями. При его запуске на экран последовательно выводятся четыре предложения:

```
print('Знакомься: это Python!')
print('В рейтинге языков программирования 2023 года он
занимает ')
print(2)
print('место!')
```



Здесь **print** — функция вывода данных на экран. В результате выполнения данной инструкции на экран будут выведены *аргументы* функции, записанные в круглых скобках. Если аргументом функции является строка символов, то она заключается в кавычки, если — число, то кавычки не пишутся.

ЗАДАНИЕ.

Выполните данную программу на платформе Programiz.com. Придумайте свой вариант предложений.

Выше мы говорили, что инструкции отделяются друг от друга переходом на новую строку. Но есть и другой способ: их можно разделить точкой с запятой (;), например:

```
print('Mother of Dragons. '); print('Drakarys!')
```

Технической разницы между первым и вторым вариантом нет — интерпретатор поймет инструкции одинаково. Разница только в том, что человеку будет неудобно читать второй вариант.

Арифметические операторы (операции)

На базовом уровне компьютеры оперируют только числами. Арифметика в программировании практически не отличается от школьной арифметики. Перечислим арифметические операции, используемые в языке Python:

- 1) + — сложение
- 2) - — вычитание
- 3) * — умножение
- 4) ** — возведение в степень
- 5) / — деление
- 6) // — целочисленное деление
- 7) % — остаток от деления

Теперь выведем на экран результат деления, результат возведения в степень и результат получения остатка от деления:

```
print(8 / 2)    # => 4.0 (При делении двух чисел получается
тип данных float)
print(3 ** 2)   # => 9
print(9 % 2)    # => 1
```

ЗАДАНИЕ.

Напишите и выполните программу на платформе Programiz.com, которая считает и последовательно (по одному значению в каждой строке) выводит на экран значения следующих математических выражений:

- 3 в степени 5.
- -8 разделить на -4.
- Остаток от деления 100 на 3.
- Сумму трех предыдущих выражений.

Подсказка: при делении чисел в этом упражнении нужно использовать обычное деление, а не целочисленное.

! Решение будет на следующей странице

Решение:

```
print(3 ** 5)          243
print(-8 / -4)         2.0
print(100 % 3)         1
print((3 ** 5) + (-8 / -4) + (100 % 3)) 246.0
```

Операнды

Вспомним определение инструкции. В нём сказано, что это код, содержащий в себе логическую последовательность действий над какими-либо *объектами*.

Объекты, над которыми выполняются операции (действия), называются **операндами**. Операндами могут быть константы, переменные, функции, выражения.

Математические (арифметические) операции, выполняемые над операндами, можно считать инструкциями, то есть операторами. Например, в выражении `print(8 + 2)` знак « + » — это оператор, а числа 8 и 2 — это операнды.

Все операторы всегда должны отделяться пробелами от операндов.

Когда мы выполняем операцию сложения, у нас есть два операнда: один слева, другой справа от знака +. Операции, которые требуют наличия двух операндов, называются **бинарными**. Если пропустить хотя бы один операнд, например, 3 +, то программа завершится с синтаксической ошибкой.

Примеры операндов:

2 + 5 (числа 2 и 5 – операнды);

x – y (переменные x и y – операнды);

A2 + B**3** (переменные A и B – операнды, выражения A^2 и B^3 – тоже операнды).

Операции бывают не только бинарными, но и **унарными** — с одним операндом, и **тернарными** — с тремя операндами. Причем операторы могут выглядеть одинаково, но обозначать разные операции. Символы **+** и **-** используются не только как операторы. Когда речь идет про отрицательные числа, то знак минуса становится частью числа: `print(-3) # => -3`.

Это пример применения унарной операции к числу 3. Оператор минус перед тройкой говорит интерпретатору взять число 3 и найти противоположное, то есть -3. Это может сбить с толку, потому что -3 — это одновременно и число само по себе, и оператор с операндом. Но у языков программирования такая структура.

Язык Python поддерживает следующие типы операторов:

- Арифметические операторы.
- Операторы сравнения (отношения).
- Операторы присваивания.
- Логические операторы.
- Побитовые операторы.
- Операторы членства.
- Операторы идентификации.
- Операторы описания алгоритмических конструкций.

В процессе изучения данного курса вы с ними познакомитесь.

2. Просмотр учебного видеоролика «Операторы в Питоне» (с целью расширения знаний, наглядного представления и закрепления материала).

<https://rutube.ru/video/53c777ae9f64257ee5ba198aed54755e/?r=plwd> (длина ролика: 1:20).

3. Закрепление и проверка знаний

- 1) выполнение задания по составлению программы на он-лайн платформе Programiz.com;
- 2) ответы на вопросы с автоматической проверкой.