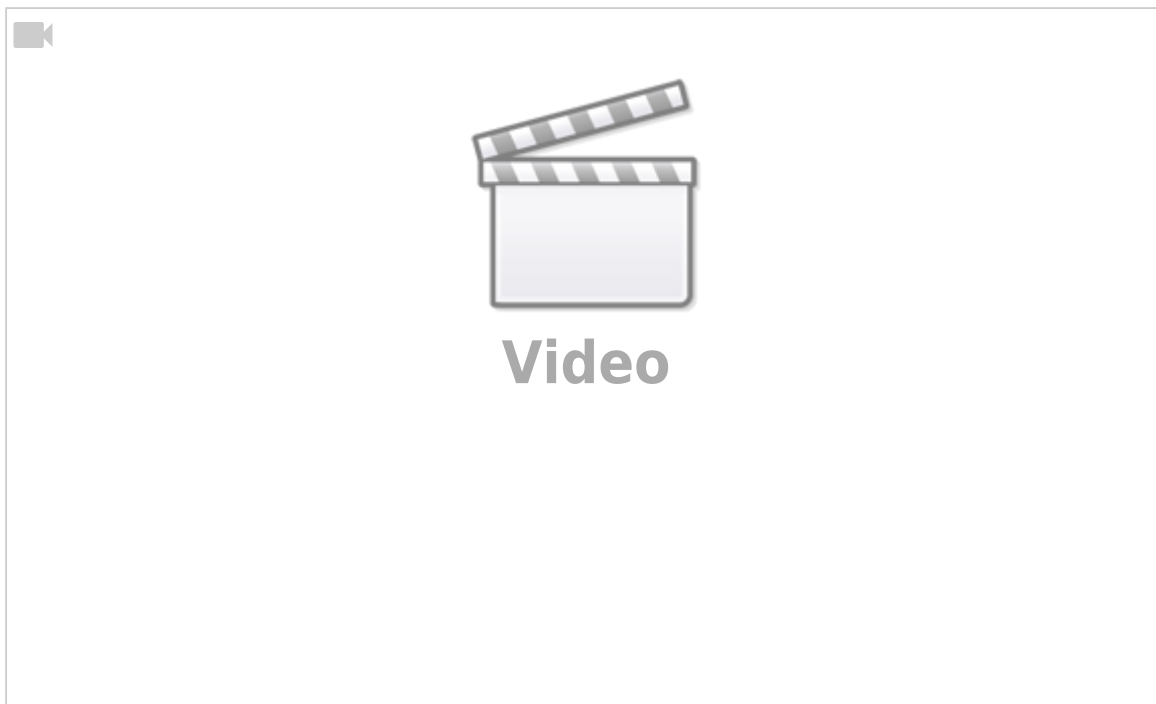
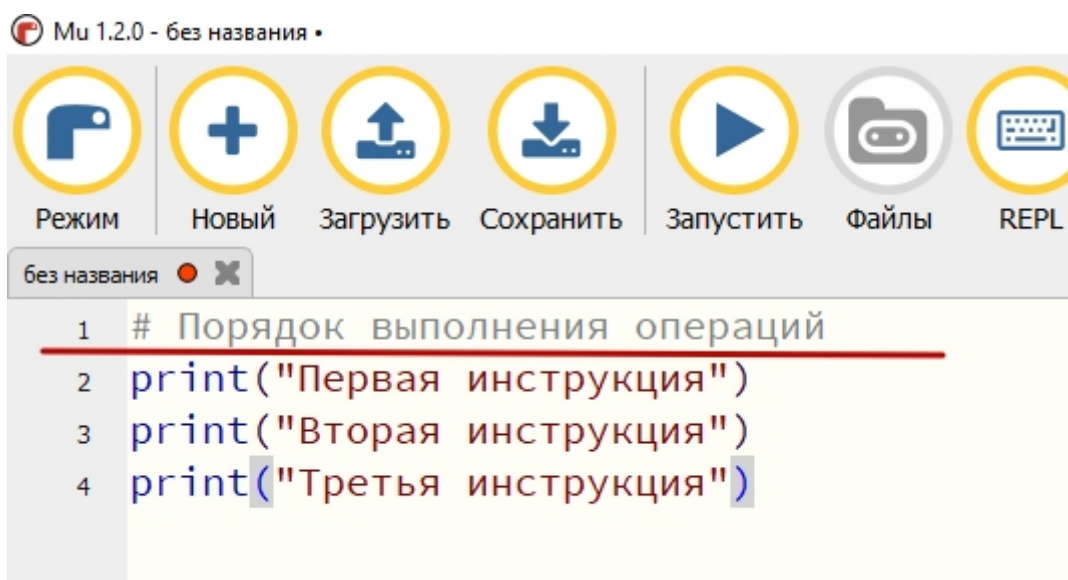


Урок 2. Порядок инструкций. Операции с числами



Убедившись что мы уже умеем программировать (ведь мы написали первую программу), отправимся дальше.

В первой строке кода в прошлом примере мы использовали комментарий - это строка, которая игнорируется контроллером при исполнении кода. Комментарии нужны для того, чтобы сделать код понятнее и нагляднее. То есть это пояснения для нас самих, или тех, кто будет после нас читать наш код. Если в начале строки поставить символ "#", то строка будет являться комментарием.



```
Ми 1.2.0 - без названия •
Режим | Новый | Загрузить | Сохранить | Запустить | Файлы | REPL
без названия ● ✕
1 # Порядок выполнения операций
2 print("Первая инструкция")
3 print("Вторая инструкция")
4 print("Третья инструкция")
```

Инструкцией будем называть законченную часть кода (можно сказать что это команда контроллеру). Например, в первом примере у нас была инструкция "Напечатайте мне фразу

Привет, Мир!", которая на языке выражалась как

```
print("Привет, Мир!")
```

Если мы захотим выполнить несколько инструкций, то каждую из них нужно указывать на новой строке. В конце строки (и инструкции) никакие дополнительные символы (такие, как характерные для других языков, например, ";") не требуются.

Инструкции выполняются сверху вниз. То есть если мы захотим выполнить несколько инструкций, то нужно их писать в порядке, начиная сверху. Например, выведем подряд три фразы: "Первая инструкция", "Вторая инструкция", "Третья инструкция". Код будет следующим:

```
# Порядок выполнения операций  
print("Первая инструкция")  
print("Вторая инструкция")  
print("Третья инструкция")
```

Выполнив данный код, мы увидим в окне вывода (терминале - то что пишет нам контроллер) три строки, которые выведены именно в том порядке, который мы и указали.



Обратите внимание, что загрузка кода напрямую (то как мы загружаем скрипты в этом разделе) может происходить не с первого раза. Поэтому если не получилось загрузить скрипт сразу, попробуйте несколько раз нажать кнопку Запустить.

ESP MicroPython REPL

Первая инструкция

Вторая инструкция

Третья инструкция

☒☒>

MicroPython v1.19.1 on 2022-06-18; ESP module with ESP8266
Type "help()" for more information.

>>>

Мы уже познакомились со встроенной функцией `print`, которая выводит текст, указанный в первом параметре в скобках. Но помимо текста, она также может выводить и числа, и результат над основными операциями с числами.

Основные операции с числами выполняются знакомыми нам символами, но помимо стандартных операций: сложения, вычитания, умножения и деления, есть ещё несколько

полезных. Итак, если у нас есть два числа x и y то операции над ними будут выглядеть следующим образом:

$x + y$	Сложение
$x - y$	Вычитание
$x * y$	Умножение
x / y	Деление
$x // y$	Целая часть от деления
$x \% y$	Остаток от деления
$x ** y$	Возведение числа x в степень y
$-x$	Смена знака числа

Попробуем посмотреть результат выполнения этих операций над числами 5 и 2.

```
# Операции с числами
print(5 + 2)
print(5 - 2)
print(5 * 2)
print(5 / 2)
print(5 // 2)
print(5 % 2)
print(5 ** 2)
print(-5)
```

В окне вывода мы увидим последовательный вывод результата наших операций:

```
ESP MicroPython REPL

7
3
10
2.5
2
1
25
-5
☒☒>
MicroPython v1.19.1 on 2022-06-18; ESP module with ESP8266
Type "help()" for more information.
>>> |
```

Что и является результатом выполнения математических операций над числами которые мы указали.

[Предыдущий урок](#)

[Следующий урок](#)

From:
<https://know.gikkon.ru/> -

Permanent link:
https://know.gikkon.ru/main/gikkon_start/p1_l2

Last update: **2023/08/03 07:37**

