

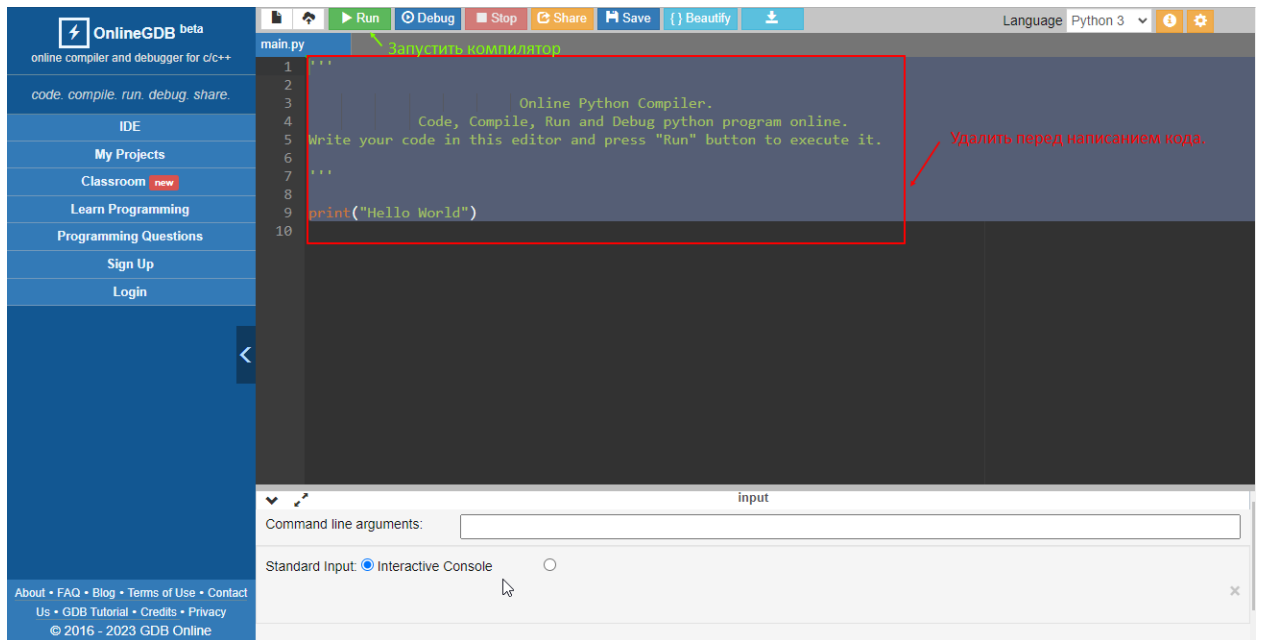
## Знакомство с Python

Для написания кода можно воспользоваться онлайн-компилятором по ссылке:

[https://www.onlinegdb.com/online\\_python\\_compiler](https://www.onlinegdb.com/online_python_compiler)

Для работы с компилятором необходимо удалить текст перед написанием кода

Для выполнения кода нажать кнопку Run



После выполнения задания делать скриншоты. Оформить отчет в текстовом редакторе

Шпаргалка:

## Знакомство с Python

Клиент и сервер взаимодействуют в интернете.  
Бэкенд-разработчик пишет программы для серверов.

Бэкенд разрабатывают на различных языках программирования, в том числе на Python.

### Первая программа на Python

```
In
# Приветствие миру – традиционная
# первая строка в освоении
# нового языка программирования
print('Привет, Мир!')
```

Out Привет, Мир!

Любая строка, начинающаяся с символа # – это комментарий, примечание для разработчика.

```
# эта команда показывает "Zen of Python"
import this
```

### Типы переменных

Переменные в Python могут быть разного типа – целые числа, дробные числа, строки:

```
In
one_hundred = 100
five_hundred = 500
print(one_hundred + five_hundred)
```

Out 600

```
In
first = 87.2
second = 50.2
third = 50.242
print(first + second + third)
```

Out 187.642

```
In
first = '87.2'
second = '50.2'
third = '50.242'
# оператор + объединяет строки:
print(first + second + third)
```

Out 87250.250.242

Числа приводятся к строкам функцией `str()`:

```
# объявляем две переменные разных типов:
number = 100
rubles = ' рублей'
In
# сложить число и строку невозможно,
# поэтому приводим number к строке:
print(str(number) + rubles)
```

Out 100 рублей

Дробные числа приводятся к целым функцией `int()`. Она не округляет числа, а отбрасывает дробную часть:

```
In
# округление вниз, как привыкли
print(int(3.14))
```

Out 3

```
In
# а здесь всё равно округление вниз,
# хотя вроде бы так быть не должно
print(int(2.72))
```

Out 2

## Списки

Списки — это последовательности, похожие на массивы из других языков программирования. Они записываются в квадратных скобках через запятую:

```
In bremen_musicians = ['Трубадур', 'Кот', 'Пёс', 'Осёл', 'Петух']  
    print(bremen_musicians)
```

```
Out ['Трубадур', 'Кот', 'Пёс', 'Осёл', 'Петух']
```

Чтобы подсчитать, сколько в списке элементов, вызывают стандартную функцию `len()`:

```
In count = len(bremen_musicians)  
    print(count)
```

```
Out 5
```

### Задание 1

С этого задания вы начинаете создавать собственного персонального помощника, вроде Алисы, Google Assistant, Siri или Alexa. Назовём её Анфиса.

Для начала научите Анфису здороваться: код уже подготовлен, но Python не станет его выполнять, ведь он скрыт за символом комментария, «закомментирован».

Сделайте так, чтобы Python увидел и выполнил код. Для этого уберите символ комментария и пробел в начале строки (это называют «раскомментировать» строку).

Код Python Скопировать текст, вставить в компилятор, исправить согласно заданию и выполнить код. Сделать скриншот	# print('Привет, я Анфиса!')
--	------------------------------

### Задание 2

Анфиса поздоровалась, но для светского разговора этого маловато. Пусть Анфиса немножко расскажет о себе.

После строки `print('Привет, я Анфиса!')` добавьте новую строку кода, которая напечатает текст 'Я маленькая, но уже программа!'.

Лайфхак: функцию `print()` (да и любую функцию вообще) можно вызывать в коде сколько угодно раз.

### Задача 3

Чтобы Анфиса стала более дружелюбной, научите её рассказывать о себе. Для начала напечатайте на экране фразу:

**'Привет, я Анфиса, твой персональный помощник!'**

Вместо многоточий подставьте в код переменные `name` и `job`. Будьте внимательны, не пропустите запятую, восклицательный знак или пробел: при проверке задания это будет учитываться.

Код Python	name = 'Анфиса'	# Имя
------------	-----------------	-------

Скопировать текст, вставить в компилятор, исправить согласно занюю и выполнить код. Сделать скриншот	<pre> job = 'персональный помощник' # Профессия  # Подставьте имена переменных вместо многоточий print('Привет, я ' + ... + ', твой ' + ... + '!')</pre>
--	--

#### Задача 4

Если бы Анфиса поселилась в фитнес-трекере, она могла бы подсчитывать количество шагов пользователя и сообщать ему результат.

Научите Анфису подставлять в сообщение сумму шагов, пройденных за два дня.

В переменных `steps_today` и `steps_yesterday` записано, сколько шагов прошёл незнакомый вам Геннадий вчера и сегодня.

Вычислите в коде сумму шагов за два дня и сохраните результат в переменную `steps_sum`. А программа напечатает получившийся результат.

<p>Код Python</p> <p>Скопировать текст, вставить в компилятор, исправить согласно занюю и выполнить код. Сделать скриншот</p>	<pre> steps_yesterday = 8452 steps_today = 6783 # Вычислите сумму здесь: вместо многоточия напишите выражение, # суммирующее количество шагов за два дня steps_sum = ... print('Сколько шагов сделал Геннадий за два дня?') print(steps_sum)</pre>
---	--

#### Задание 5

Научим Анфису информировать вас о новых сообщениях, которые вы могли бы получить.

Выведите на экран строку 'У вас 8 новых сообщений', составленную

из строки 'У вас ',

значения переменной `count`

и из строки ' новых сообщений'.

<p>Код Python</p> <p>Скопировать текст, вставить в компилятор, исправить согласно занюю и выполнить код. Сделать скриншот</p>	<pre> count = 8 message = 'У вас ' + str(...) + ... # Допишите ваш код здесь print(message)</pre>
---	---