

# Python, ЗНАКОМСТВО С КОНСОЛЬЮ



Гежин Олег



**Олег Гежин**

Python - разработчик, специалист SQL, фрилансер.



# План занятия

1. [Почему Python?](#)
2. [Онлайн-интерпретатор repl.it](#)
3. [Числа и арифметические операции](#)
4. [Переменные](#)
5. [Строки](#)
6. [Комментарии](#)



# Почему Python?

---

# Python 3.x


Python – интерпретируемый язык с динамической типизацией.

Плюсы:

- Прекрасно подходит для новичков
- Широкая область применения
- Богатое и дружелюбное сообщество разработчиков
- Востребованность

Сферы применения:

- Веб-разработка (Google, Yandex, Youtube..)
- Анализ данных и машинное обучение (Google, Microsoft)
- Игры (Wargaming, Eve)
- Прототипирование
- Тестирование

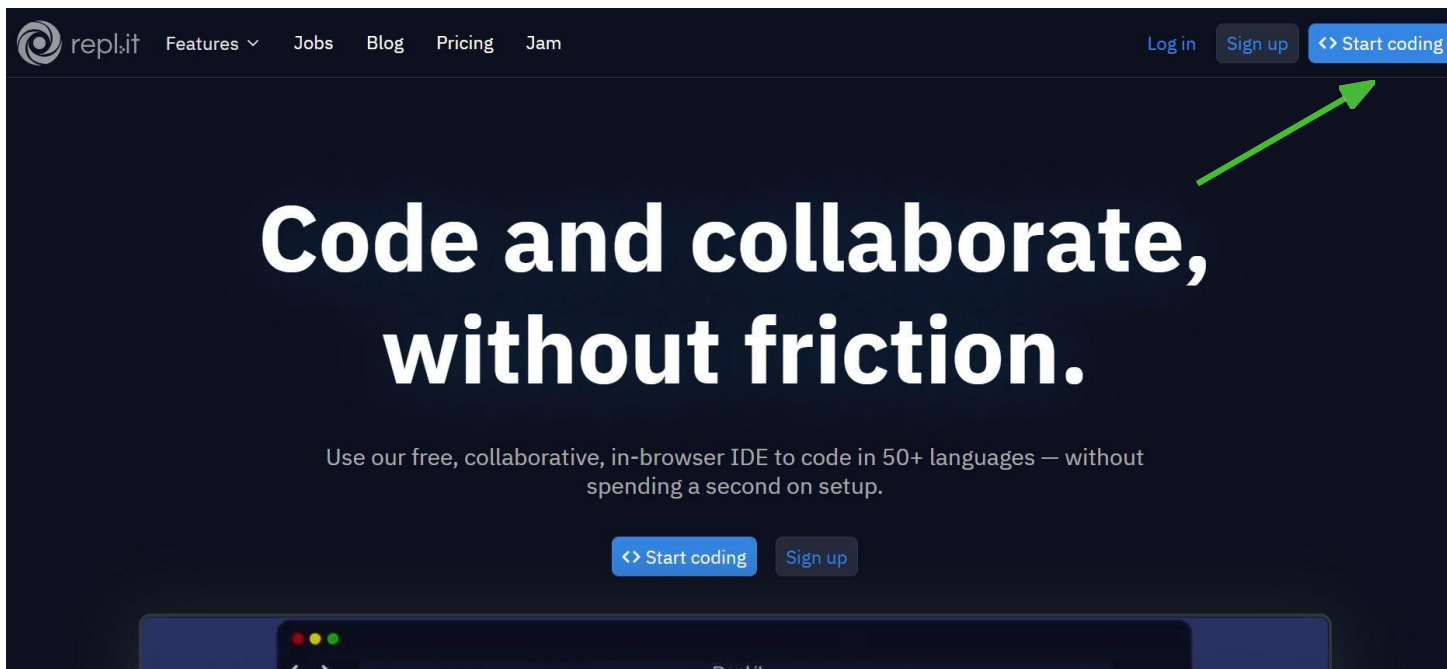


# Онлайн-интерпретатор repl.it

# Создание программы в Repl.it

## Шаг 1

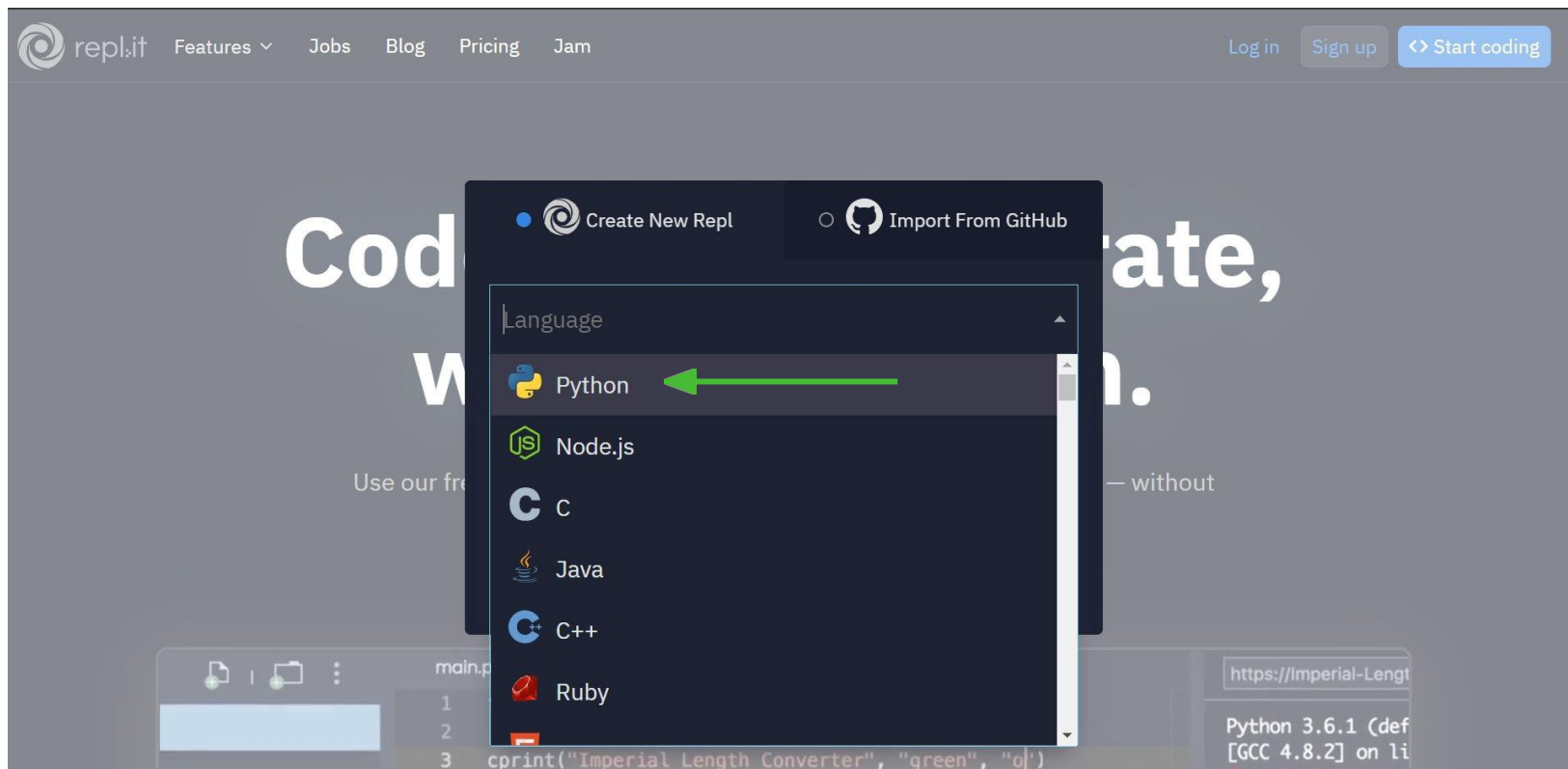
Открываем в браузере сайт <https://repl.it/> и нажимаем кнопку **Start coding** (правый верхний угол).



# Создание программы в Repl.it

## Шаг 2

В открывшемся окне **выбираем язык Python.**

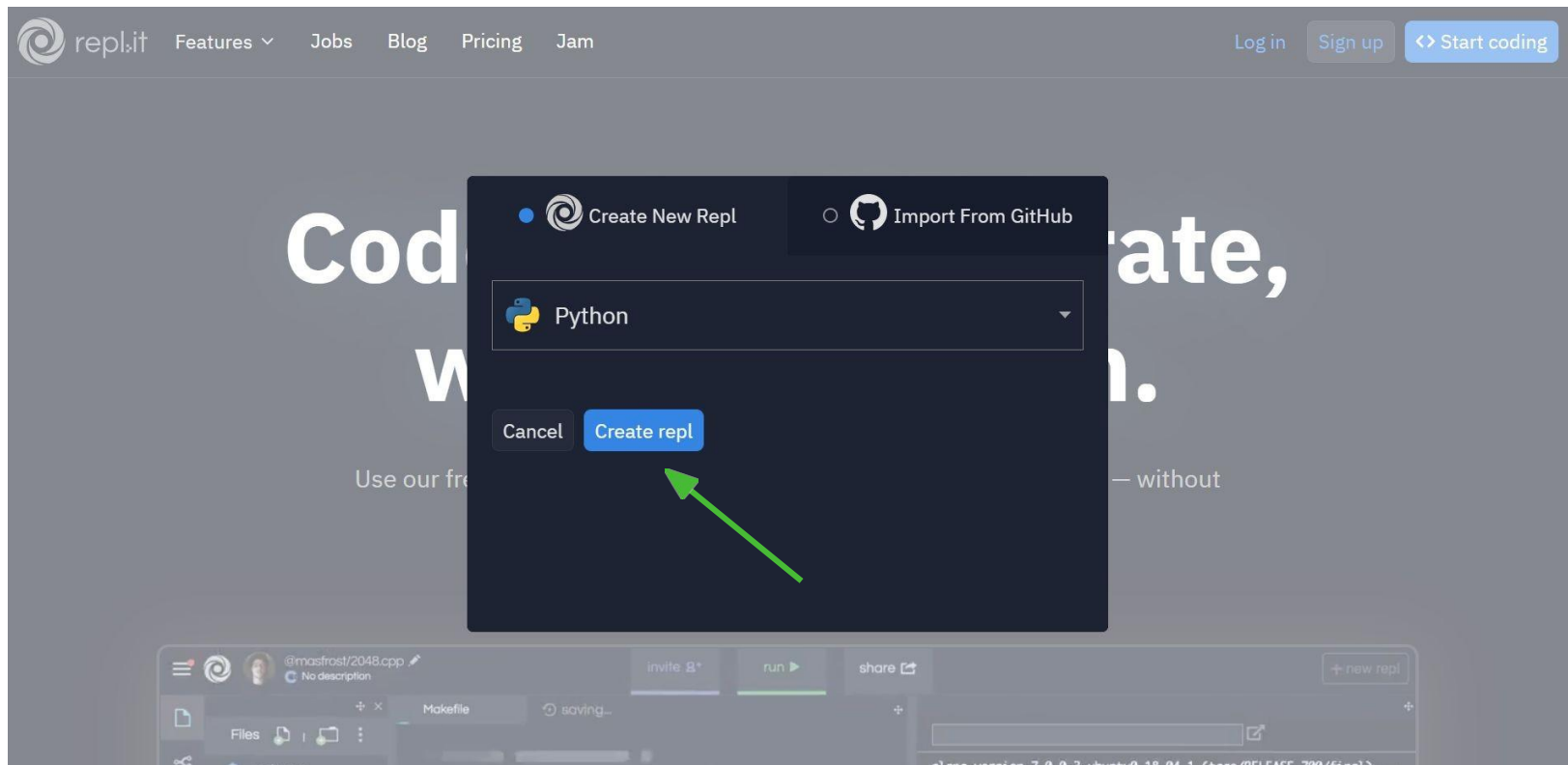




# Создание программы в Repl.it

## Шаг 3

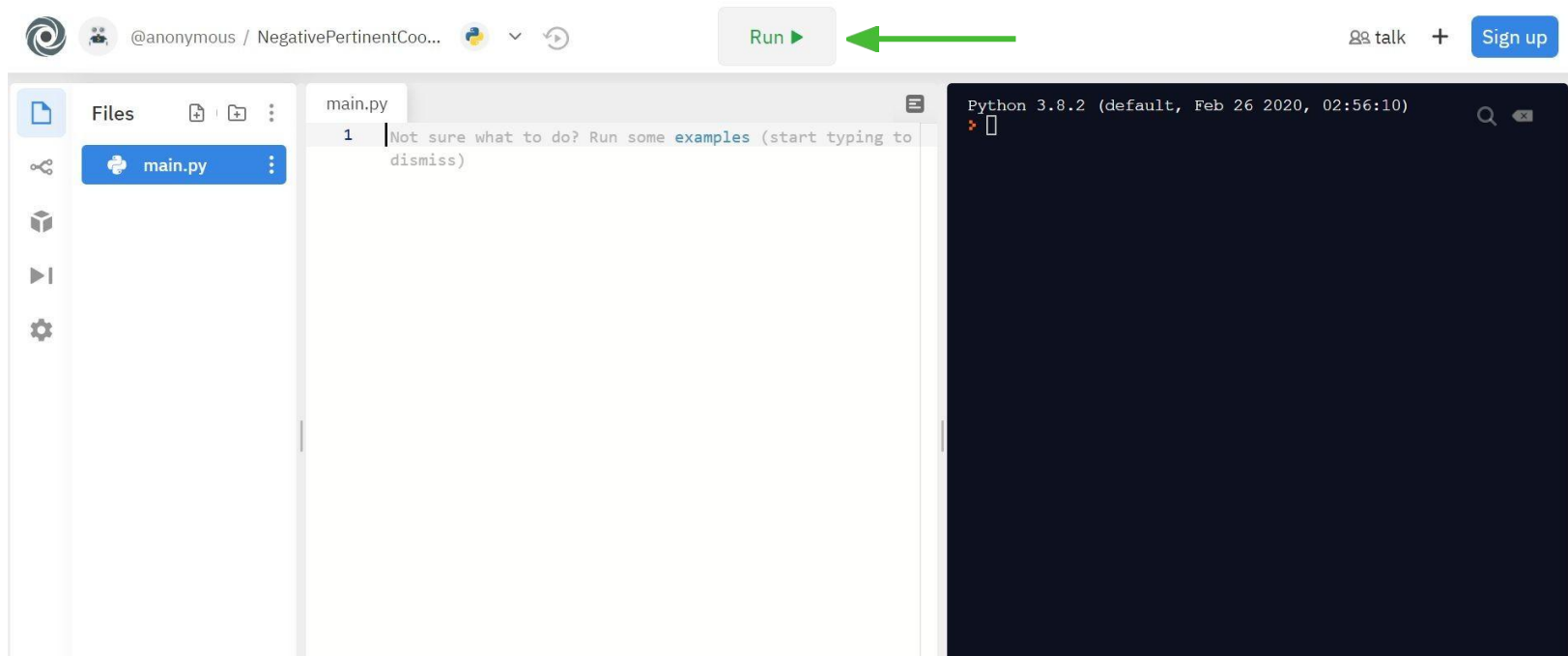
Нажимаем кнопку **Create Repl**.



# Создание программы в Repl.it

## Шаг 4

Для запуска программы нужно нажать кнопку **Run**.





# **Основы программирования**

# Что такое программа?

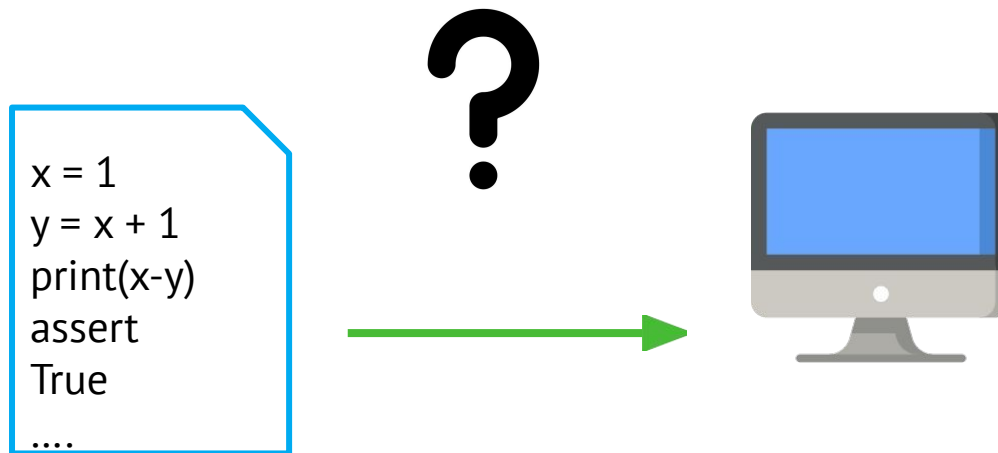


Программа = Инструкции + Данные

# Выполнение программы

Мы пишем программы на языке программирования, а компьютер понимает 0 и 1.

Как же выполняются наши программы?



# Выполнение программы

	Отличия	Примеры
Компилируемые	<ul style="list-style-type: none"><li>● Специальная программа (компилятор) преобразует исходный код в машинный</li><li>● Полученный в результате файл готов к исполнению без дополнительных инструментов</li></ul>	<ul style="list-style-type: none"><li>● C++</li><li>● Golang</li><li>● Rust</li></ul>
Интерпретируемые	<ul style="list-style-type: none"><li>● Код исполняется строка за строкой специальной программой (интерпретатор)</li><li>● Исполнение возможно только при наличии этой программы</li></ul>	<ul style="list-style-type: none"><li>● Python</li><li>● Lua</li><li>● JavaScript</li></ul>



# Числа и арифметические операции

---

# Арифметические операции

- сложение (+);
- вычитание (-);
- умножение (\*);
- деление (/);
- целочисленное деление (//);
- возведение в степень (\*\*);
- взятие остатка от деления (%).

Приоритет операций аналогичен стандартным математическим правилам.

Можно использовать скобки.





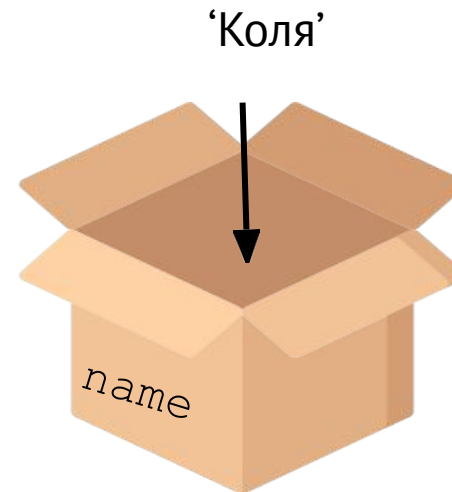
# Переменные

# Переменные

Переменная – это объект, которому дано имя. Необходимо для хранения данных и промежуточных результатов вычислений.

Объект – это:

- число;
- строка;
- практически что угодно в Python.



Python – язык с динамической типизацией. Это значит, что он самостоятельно определяет тип объекта, который мы хотим сохранить в переменной.

## Таблица с примерами значений

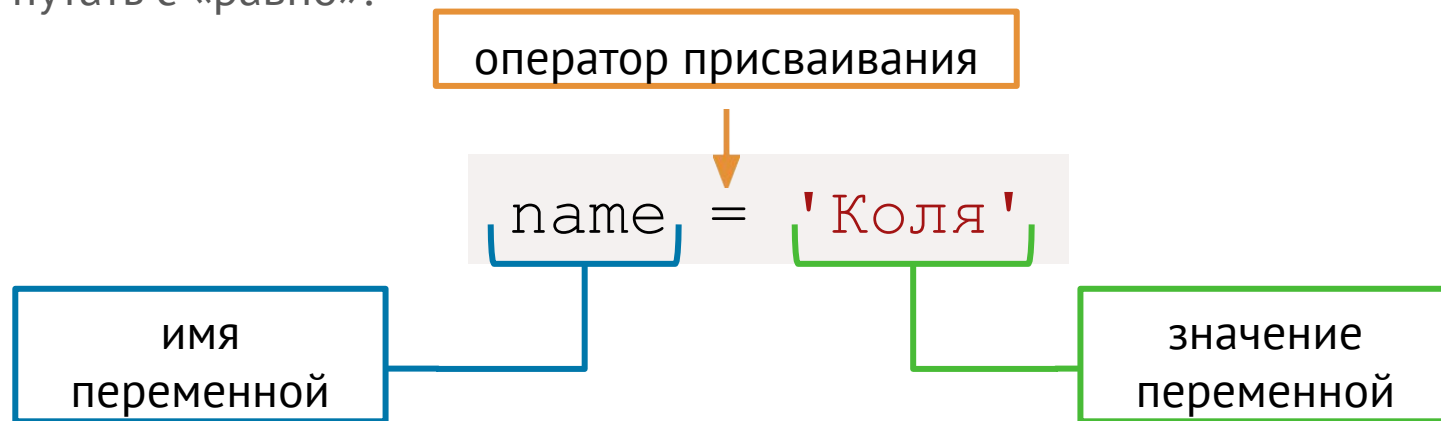
Тип переменной	Пример
bool	True или False, 0 или 1
int (целое число)	2021, 10, 15
float (число с плавающей точкой)	3.29, 1.234567
str (строка)	"привет", "hello", "2021", "True"

Строковые значения всегда заключаются в кавычки: одинарные или двойные. В одной строке **нельзя** использовать разные типы кавычек.

# Переменные

Чтобы сохранить значение в переменной, используется **оператор присваивания «=»**.

Не путать с «равно»!



Переменной name присвоено значение 'Коля'.

# Именованние переменных

Правила именованния:

- Имя переменной может состоять только из цифр, латинских букв и знака подчеркивания.
- Имя переменной **не может** начинаться с цифр.

Рекомендации именованния:

- Имя переменной должно описывать её суть.
- Лучше использовать **snake\_case** (слова с маленькой буквы и разделять подчеркиванием).
- Переменные именуем только на английском языке. Транслит не используется.

Определите, какие имена правильные:

1. zarplata = 1000   2. age = 25   3. 5element = True   4. familia = 45

5. month = 'May'   6. qqk = 'Spartak'   7. is\_number = True

# Именованние переменных

Правила именованния:

- Имя переменной может состоять только из цифр, латинских букв и знака подчеркивания.
- Имя переменной **не может** начинаться с цифр.

Рекомендации именованния:

- Имя переменной должно описывать её суть.
- Лучше использовать **snake\_case** (слова с маленькой буквы и разделять подчеркиванием).
- Переменные именуем только на английском языке. Транслит не используется.

Проверьте себя:

1. `zarplata = 1000`
2. `age = 25`
3. `Selement = True`
4. `familia = 45`
5. `month = 'May'`
6. `qqq = 'Spartak'`
7. `is_number = True`

# Вывод в консоль

Функция **print()** в Python выводит заданные объекты на экран.

```
print(100)
print('Hello,
World!')
print('Привет, ', 'мне', 20, 'лет')
```

Результат выполнения:

```
100
Hello, World!
Привет, мне 20
лет
```

Можно задавать разделение слов с помощью параметра **sep**. И  
окончание строки с помощью параметра **end**.

```
print('Hello', 'Python',
sep='+') print(1, 2, end='\t')
```

# input

**input** – функция для ввода данных от пользователя.

```
input('Сколько тебе лет?')
```

Чтобы использовать полученное значение в программе, сохраните его в переменной:

```
age = input('Сколько тебе лет?')
```

Функция **input()** сохраняет строку.





# Строки

---

# Строки

Строка – последовательность символов.

Свойства строк:

- Можно применять некоторые арифметические операции (+, \*).
- Поддерживают индексацию.
- У строк есть множество методов (**find**, **split**).
- Про остальные методы вы можете узнать из этой [статьи](#).

# Преобразование типов

**int()** – преобразование в число.

**str()** – преобразование в строку.

**bool()** – преобразование в булево значение: True или False (подробнее будет рассмотрено на следующих лекциях).

Примеры:

```
age = int(input('Сколько тебе лет?')) # вернёт число,  
введенное пользователем
```

```
number = 10  
print(str(number)) # вернёт строку '10'
```



# Комментарии

# Комментарии

Это текст, который присутствует в коде программы, но игнорируется интерпретатором. Используются для того, чтобы добавить объяснение для определенного блока кода.

Написание комментария начинается с символа #.

```
name = input('Enter your name') # сохраняем имя пользователя
```



## Задания с автопроверкой

[hackerrank](#)

**HackerRank** 



## Домашнее задание

# Задача №1

Квадрат и прямоугольник.

Пользователь вводит длину и ширину фигуры.

Программа выводит их периметр и площадь.

**Пример:**

```
1 Введите длину стороны квадрата: 3
2
3 Вывод:
4 Периметр: 12
5 Площадь: 9
6
7 Введите длину прямоугольника: 3
8 Введите ширину прямоугольника: 4
9
10 Вывод:
11 Периметр: 14
12 Площадь: 12
```

Программа запрашивает у пользователя длину стороны квадрата и выводит его периметр и площадь. Сразу после этого пользователю предлагается ввести длину и ширину прямоугольника, для которого рассчитывается периметр и площадь. Обратите внимание, что программа должна работать корректно при любых введённых значениях длины и ширины фигуры.



# Задача №2

Нужно разработать приложение для финансового планирования.

Приложение учитывает, какой процент от заработной платы уходит на ипотеку и ежемесячные расходы.

Для этого пользователю предлагается ввести следующие данные:

1. Зарботную плату в месяц.
2. Какой процент(%) от зп уходит на ипотеку.
3. Какой процент(%) от зп уходит “на жизнь”.

Программа подсчитывает и выводит, сколько денег тратит пользователь на ипотеку и сколько он накопит за год (остаток от заработанной платы).

## Пример:

- 1 Введите заработную плату в месяц: 100000
- 2 Введите, какой процент(%) уходит на ипотеку: 30
- 3 Введите, какой процент(%) уходит на жизнь: 50
- 4
- 5 Вывод:
- 6 На ипотеку было потрачено: 360000 рублей
- 7 Было накоплено: 240000 рублей

# Задача №3

Разделитель. Пользователь вводит какой-нибудь символ. Нужно разделить первую задачу и вторую задачу строкой из этого символа. Длина строки должна быть суммой периметра квадрата и площади прямоугольника.

## Пример:

```
1 #  
2 Вывод :  
3 #####
```

# Задача №4(необязательное)

Зарегистрироваться на сайте [hackerrank](#).

Решить задачу [“Say “Hello, World!” With Python”](#).

Решить задачу [“Arithmetic Operators”](#).

Ссылку на *аккаунт* приложите в качестве решения к домашнему заданию. Чтобы получить ссылку, на сайте hackerrank в правом верхнем углу нажмите на своё имя, перейдите в раздел “Profile” и скопируйте ссылку из адресной строки браузера.

*Пример:* ссылка на аккаунт eshmargunov - [hackerrank.com/eshmargunov](https://hackerrank.com/eshmargunov)

## **Инструкция по выполнению домашнего задания:**

1. Зарегистрируйтесь на сайте Repl.IT.
1. Перейдите в раздел my repls.
2. Нажмите кнопку Start coding now!, если приступаете впервые, или New Repl, если у вас уже есть работы.
3. В списке языков выберите Python.
4. Код пишите в левой части окна.
5. Посмотреть результат выполнения файла можно, нажав на кнопку Run. Результат появится в правой части окна.
6. После окончания работы скопируйте ссылку на ваш repl в адресной строке браузера.
7. В личном кабинете на сайте netology.ru в поле комментария к домашней работе вставьте скопированную ссылку и отправьте работу на проверку.

*Помимо ссылки, пожалуйста, прикрепите к вашему решению любой файл (например, скриншот любой части вашего кода).*

⌘ нетология

**Задавайте вопросы и  
пишите отзыв о лекции!**

Олег Гежин