

Python, ЗНАКОМСТВО С КОНСОЛЬЮ



Евгений
Шмаргунов



Евгений Шмаргунов

Ведущий специалист
по автоматизированному тестированию
Ozon



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

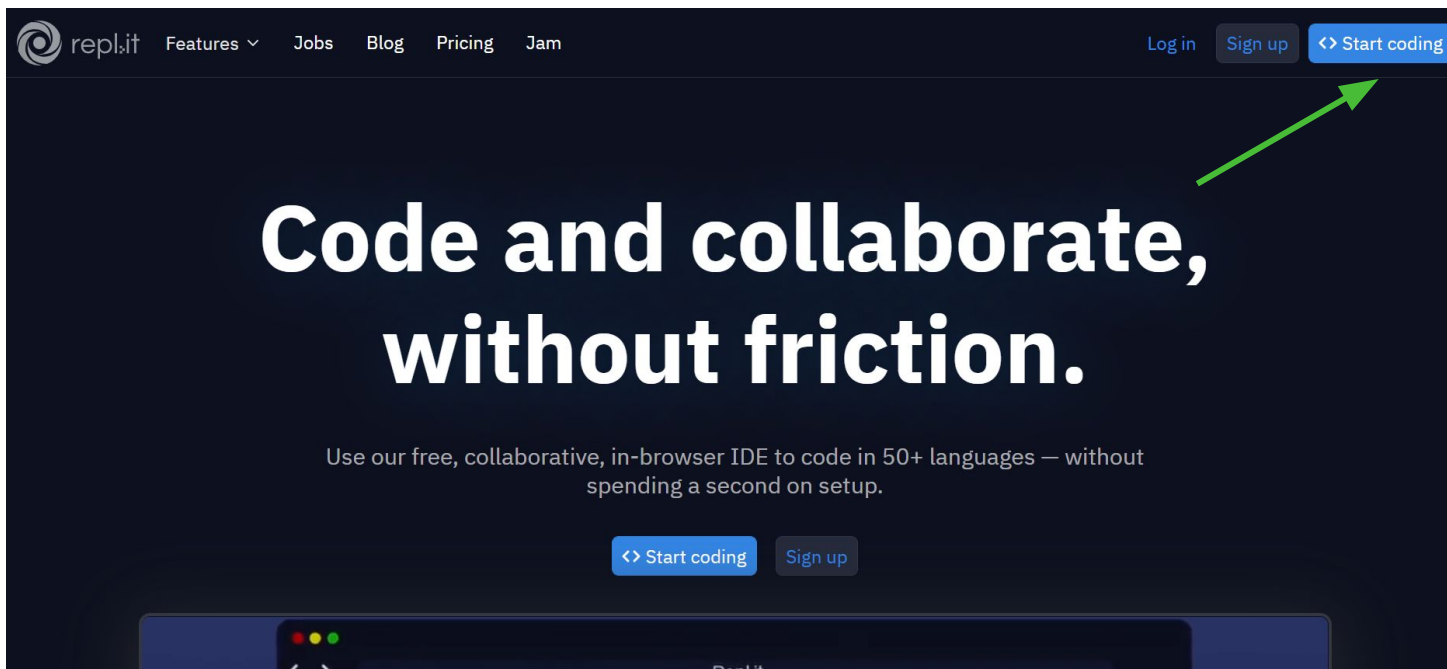


Онлайн-интерпретатор repl.it

Создание программы в Repl.it

Шаг 1

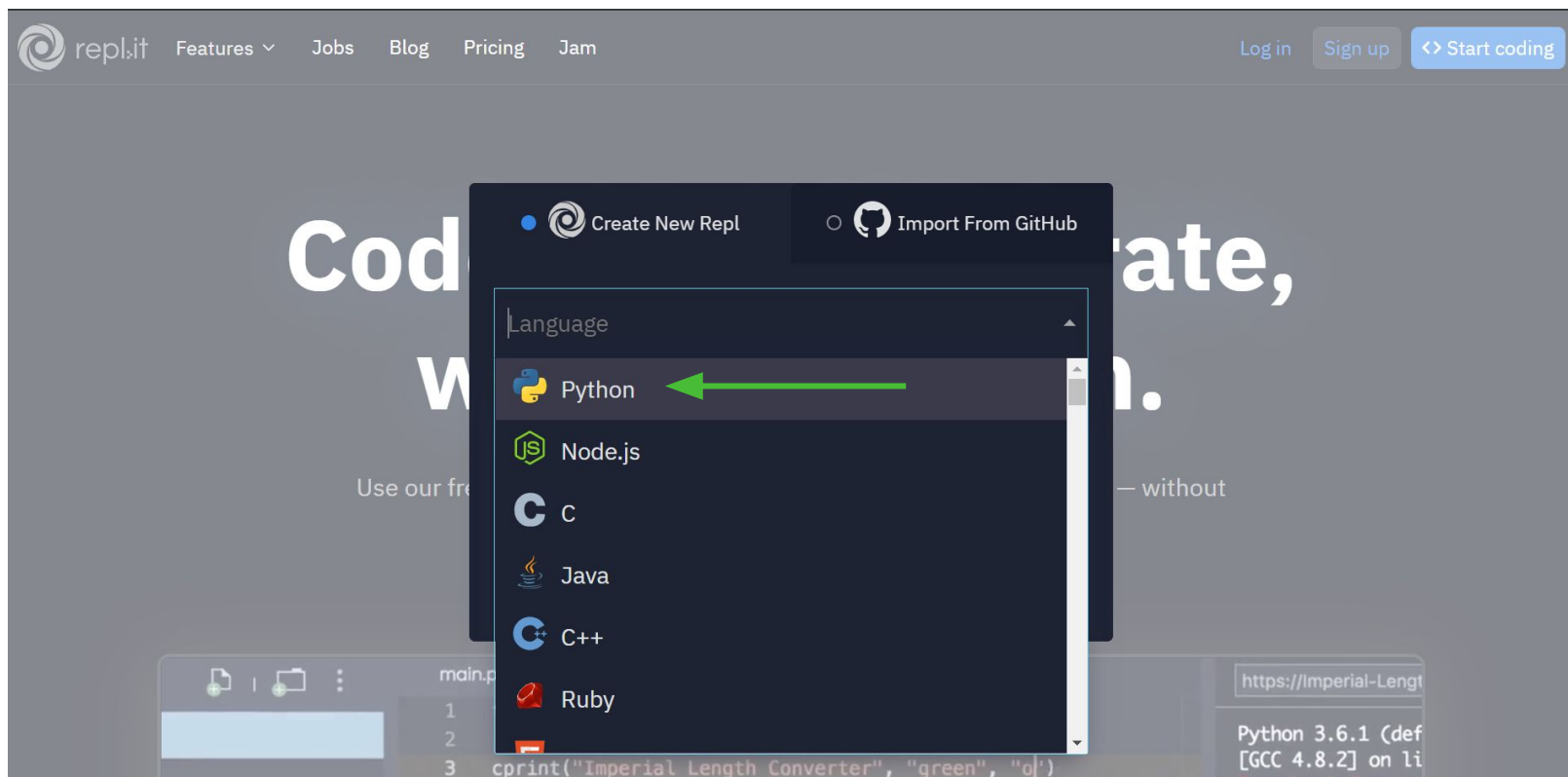
Открываем в браузере сайт <https://repl.it/> и нажимаем кнопку **Start coding** (правый верхний угол).



Создание программы в Repl.it

Шаг 2

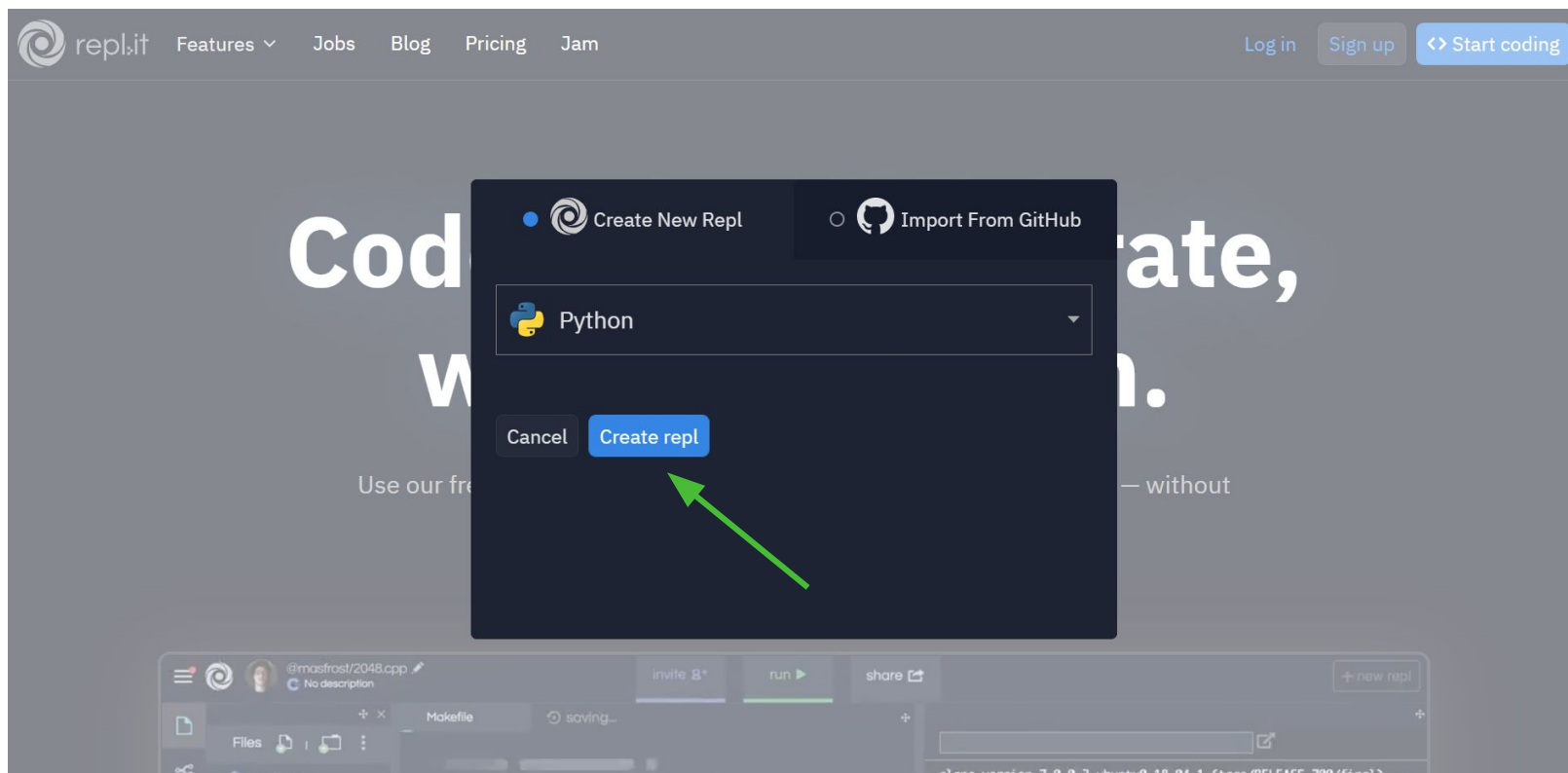
В открывшемся окне **выбираем язык Python.**



Создание программы в Repl.it

Шаг 3

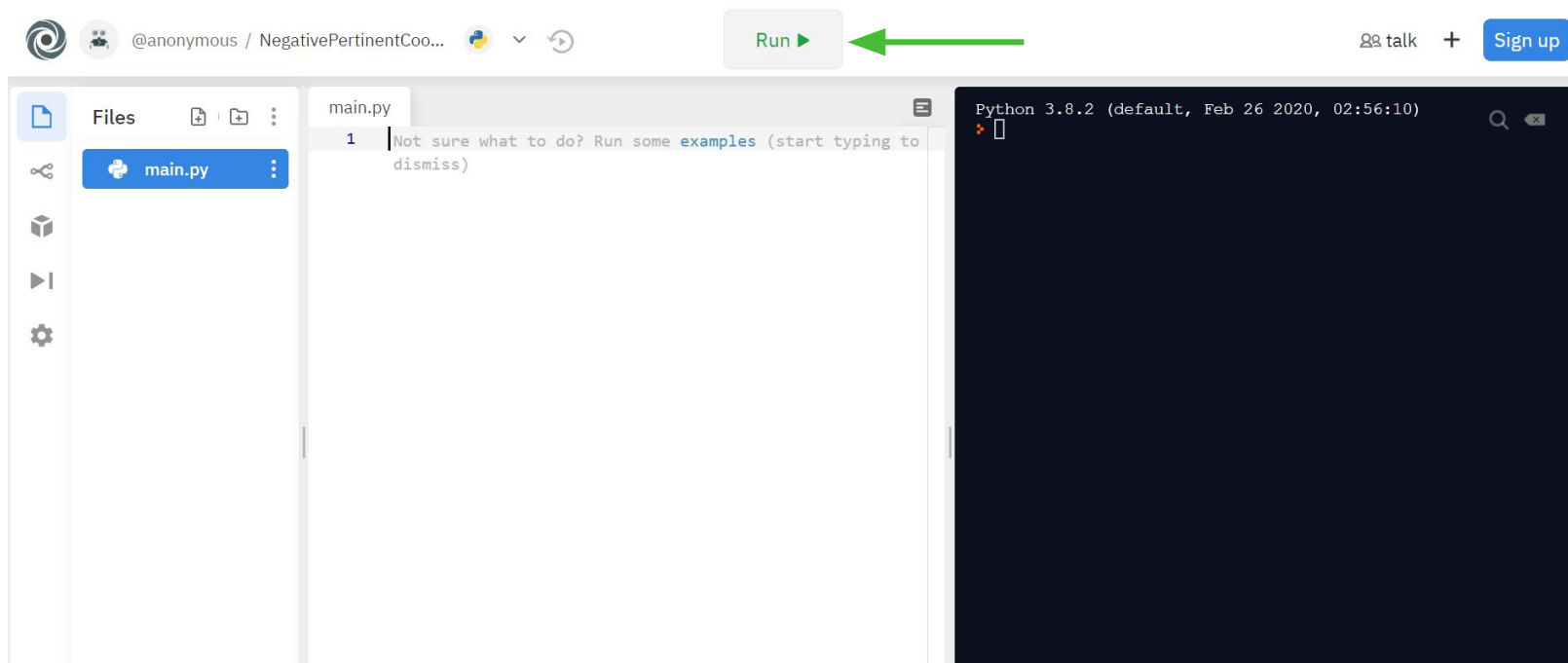
Нажимаем кнопку **Create Repl**.



Создание программы в Repl.it

Шаг 4

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





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.



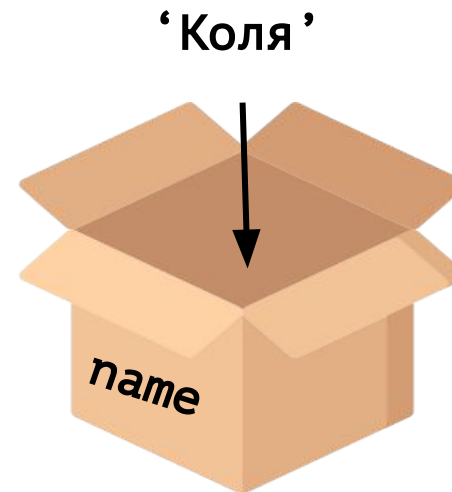
Переменные

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

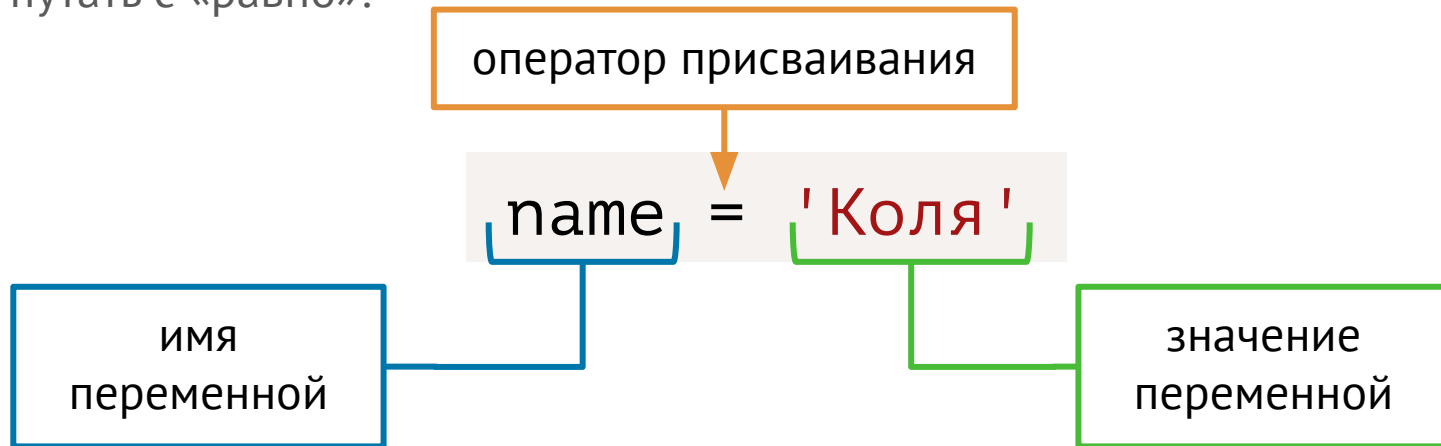
Тип переменной	Пример
bool	True или False, 0 или 1
int (целое число)	2021, 10, 15
float (число с плавающей точкой)	3.29, 1.234567
str (строка)	"привет", "hello", "2021", "True"

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 именования:

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

Рекомендации именования:

- 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.

Именованние переменных

Правила именования:

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

Рекомендации именования:

- 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.

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](#).

Операции со строками

1. Конкатенация (объединение) строк возможна при помощи `+` ;
2. Умножение строки на число позволит повторить ее нужное количество раз;
3. `.upper()` приводит строку к верхнему регистру;
4. `.lower()` приводит строку к нижнему регистру;
5. `.capitalize()` приводит первую букву к верхнему регистру;
6. `.replace('что заменить', 'на что заменить')` заменяет элемент в строке на указанный;
7. `len(my_string)` позволяет определить длину строки (количество символов в ней);

Со всеми методами работы со строками можно ознакомиться [по ссылке](#).

Форматирование строк (f-строки)

Добавляя префикс `f` к строке, можно встраивать в нее произвольные выражения при помощи фигурных скобок `{ }`.

```
name = 'Коля'
age = 13
print(f'Меня зовут {name}. Я родился в {2020-age} году.')
```


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 игнорируются интерпретатором. Используются для того, чтобы добавить объяснение для определенного блока кода.

Написание комментария начинается с символа #.

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

Домашнее задание

Чтобы проверить, как вы поняли тему лекции, и улучшить свои практические навыки программирования, рекомендую выполнить **задания в тренажёре** в личном кабинете.

В тренажёре вы сможете потренироваться с написанием кода для **3 заданий**:

1. Рассчитать площадь и периметр квадрата
2. Рассчитать площадь и периметр прямоугольника
3. Создать приложение для финансового планирования

Вопросы по домашней работе задавайте в чате группы

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

Евгений Шмаргунов