

# Кофе для друзей: начало

# 1

## В ЭТОЙ ГЛАВЕ

- ✓ Эрику приходит идея
- ✓ Эрик и Саймон обсуждают будущее приложение
- ✓ Эрик устанавливает редактор кода и пробует запустить первую программу на Python
- ✓ Саймон объясняет Эрику, как использовать переменные
- ✓ Эрик пишет первый диалог на Python

Однажды солнечным летним днем Эрику пришла в голову мысль открыть кофейню и готовить кофе для друзей. Кто бы мог подумать, что для этого он создаст свое онлайн-приложение?

«Сделаю как в Starbucks — с разными вкусами и топпингами, — подумал Эрик. — У меня есть все что нужно: кофе, три-четыре добавки для вкуса и шоколадные взбитые сливки для топпинга. Просто отлично!»

— Где мой iPad? — спросил он старшего брата Саймона.

— Там, где ты его оставил. Зачем он тебе?

— Собрать заказы для моей кофейни!

Через пару минут Эрик появился с записанными на iPad заказами. Он приготовил четыре напитка для друзей и снова ушел.

— Разве не отличная идея? — спросил Эрик у брата, вернувшись домой с четырьмя пустыми пластиковыми стаканчиками.

— Да, неплохая, — ответил Саймон. — Но...

— Что «но»? — спросил Эрик. Ему показалось, что брат, как обычно, хочет испортить ему настроение.

— Ты принимаешь заказы на iPad, но используешь его как обычный бумажный блокнот. Можно ведь создать простое приложение для заказов.

— Типа интернет-магазина? С меню и всем этим вот? — Эрик тут же представил себе страницу *собственного* интернет-магазина с огромным названием: «Кофейня Эрика».

— Конечно. У тебя же был онлайн-курс по Python, разве нет?

— Да, но я уже мало что помню. У нас были какие-то задания... Но мне кажется, сделать настоящий интернет-магазин будет сложно.

— Ничего, — сказал Саймон. — Будем продвигаться постепенно. У нас были похожие темы в кружке робототехники.

**ПРИМЕЧАНИЕ** Не волнуйся, если у тебя нет вообще никакого опыта программирования. Эрик почти ничего не помнит со своих курсов, так что мы начнем с самого начала.

Саймон заканчивает школу. Он выучил Python несколько лет назад и использовал его в школьном кружке информатики, а теперь и робототехники.

— Так что, мы правда можем сделать настоящее приложение? — не поверил Эрик.

— Да, конечно. Если ты не будешь сбегать с моих уроков, — улыбнулся Саймон, — то напишешь его за пару недель. Тогда твои клиенты смогут выбирать напитки с разными вкусами...

— И топпингами! — добавил Эрик.

— Да, и топпингами. Когда пользователь подтвердит заказ, ты сразу увидишь его на странице заказов. Ты точно будешь знать, что и кому приготовить. Примерно вот так, — Саймон взял лист бумаги и начал рисовать простую веб-страницу. — Вот страница заказов.

Кофейня Эрика

Введите свое имя

Выберите напиток

Кофе  
 Горячий шоколад  
 Кофе без кофеина

Выберите вкус

Карамель  
 Зефир  
 Клубника

Выберите топпинг

Шоколад  
 Карамель  
 Корица  
 Ваниль

— А это — список заказов.

Кофейня Эрика

Заказы

Имя	Напиток	Вкус	Топпинг
Алекс	Кофе	Карамель	Ваниль
Дрю	Кофе без кофеина	Клубника	Шоколад

— Круто! И что, у нас правда получится? — все еще не верил брату Эрик.

— Конечно! Я же сказал, только не сбегай. У тебя каникулы и полно времени, ты все успеешь.

**ПРИМЕЧАНИЕ** Если тебя не привлекает кофейня, то в книге ты найдешь другие идеи для проектов. Некоторые мы обсудим позже, когда к Эрику присоединится его подруга Эмили, а остальные собраны в приложении А.

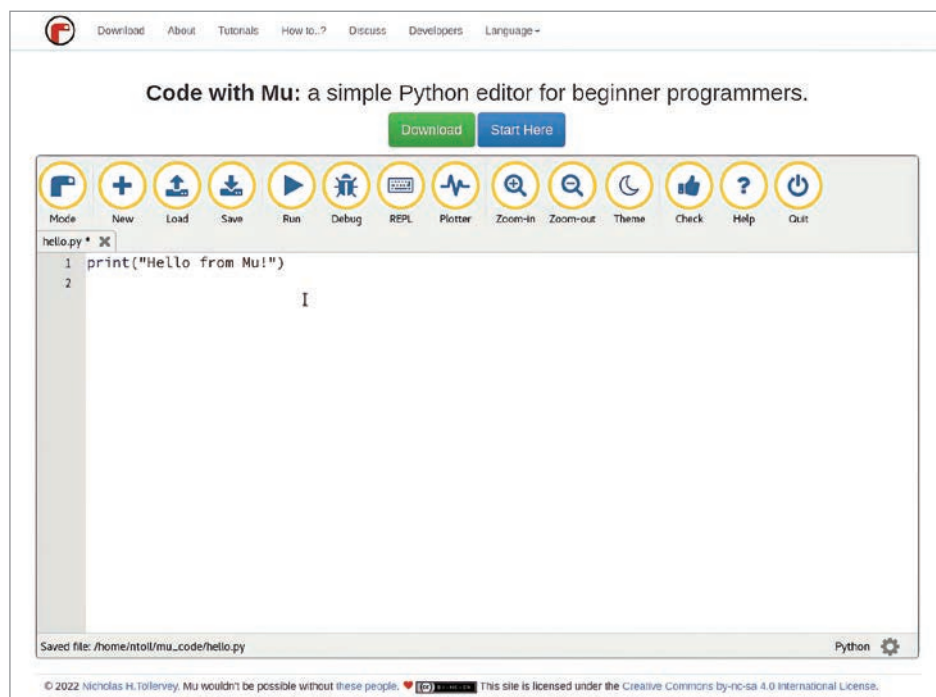
## Перво-наперво: установка

— Начнем с простого. Ты быстро вспомнишь Python. У тебя он есть на ноутбуке? — спросил Саймон.

— Скорее всего, нет.

— Вот отличный редактор Python, специально для новичков, как ты. Называется Му. Попробуй найти его и установить. Уверен, ты справишься.

Эрик нашел сайт, с которого можно загрузить Му: <https://codewith.mu/>.



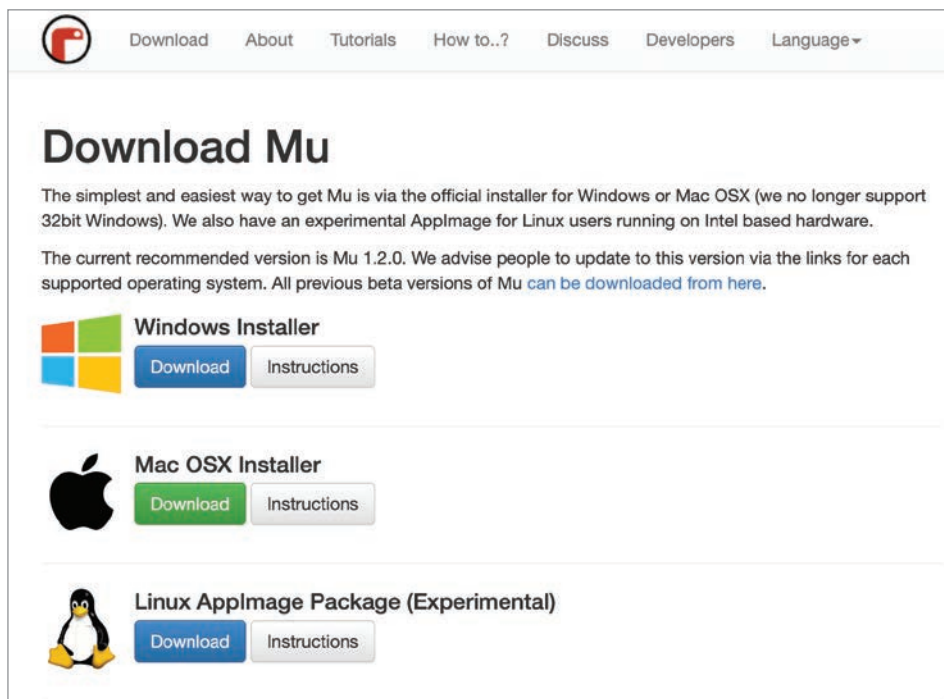
Эрик открыл страницу Download (<https://codewith.mu/en/download>) и загрузил установочную программу.

Он нажал кнопку Instructions (Инструкции) и выбрал инструкцию для своего компьютера, в которой были описаны все шаги установки.

— Не волнуйся, это отличный редактор. И вовсе не игрушечный, — сказал Саймон. — Мы используем его в кружке робототехники для работы

с микроконтроллерами. Как видишь, у него есть версии для Windows, macOS и Linux. У нас в кружке установлена Linux-версия.

**ПРИМЕЧАНИЕ** Ссылки и инструкции см. в приложении В.



Эрик не хотел слепо следовать указаниям брата и спросил:

— А есть еще какие-нибудь редакторы Python?

— Да, конечно, их много. Еще один хороший редактор для новичков — Thonny. Посмотри тут: <https://thonny.org/>.

— Круто, мне нравится! — сказал Эрик. — Забавное название.

— Конечно, есть и другие редакторы, которые совместимы с разными платформами:

- VS Code (<https://code.visualstudio.com/>);
- Sublime Text (<http://www.sublimetext.com/>).

## 24 Поговорим на Python

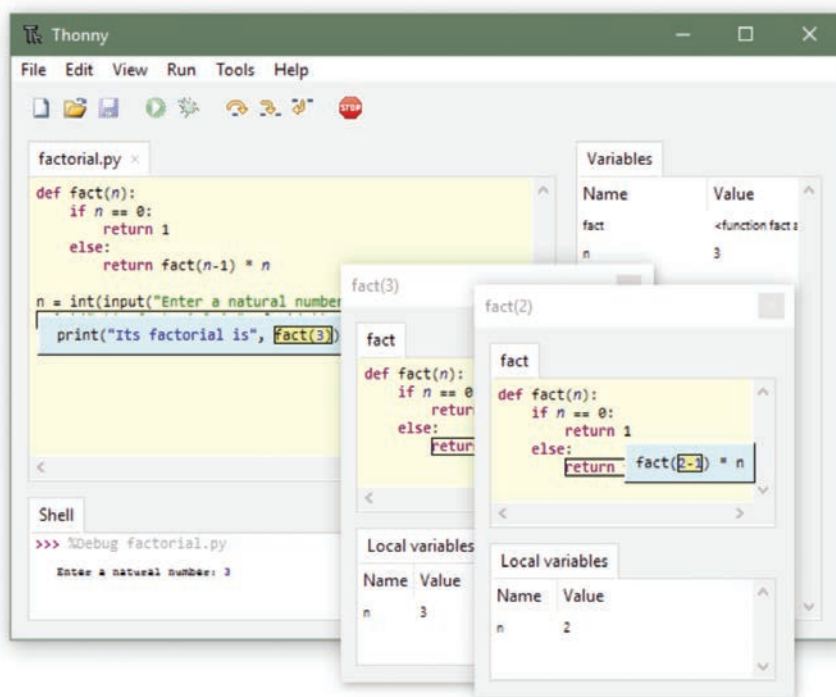
— Все они отлично работают с Python. Python поддерживают даже такие старые редакторы, как Vim (<https://www.vim.org/>) и Emacs (<https://www.gnu.org/software/emacs/>), но, чтобы их использовать, нужно быть ну очень хорошим программистом, — Саймон подмигнул брату и продолжил: — В Mu и Thonny Python уже есть. Но чтобы использовать Python и в *других* редакторах, ты должен сначала его *установить*. В некоторых операционных системах, таких как Linux и macOS, Python уже установлен по умолчанию. На Windows тебе придется установить его самостоятельно. Если захочешь, позже я покажу тебе как.

# Thonny

Python IDE for beginners



Download version [4.0.1](#) for  
[Windows](#) • [Mac](#) • [Linux](#)



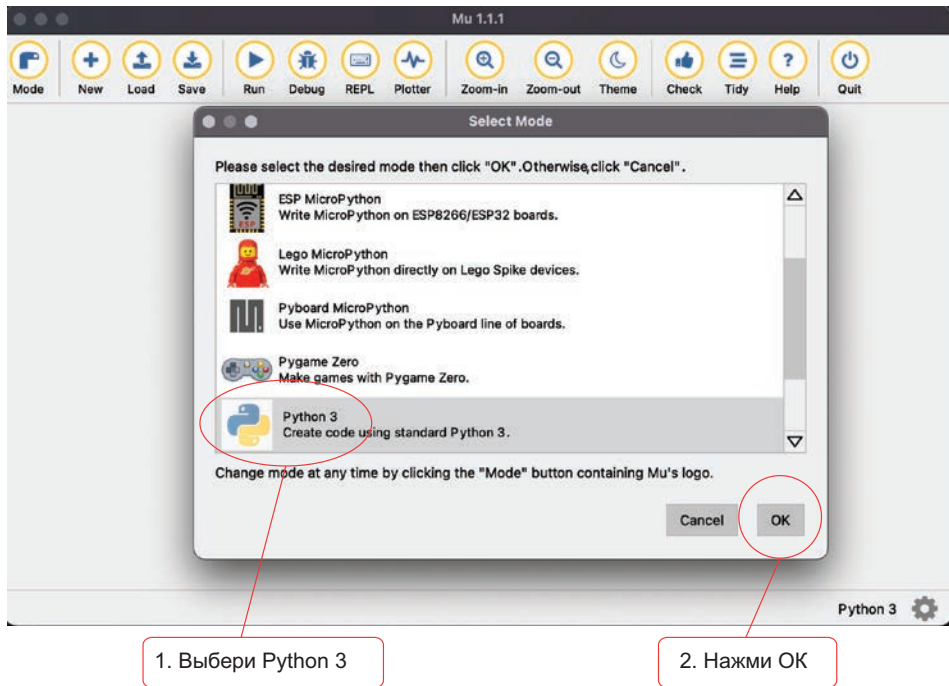
**ТВОЙ ХОД! Установи среду для работы с Python**

Теперь твоя очередь. Запусти ноутбук или стационарный компьютер и установи редактор Mu. Полные инструкции для каждой платформы (Windows, macOS и Linux) см. в приложении В.

Если ты предпочитаешь другой редактор, можешь установить его вместо Mu. Не бойся экспериментировать!

**Как говорить с компьютером**

— Давай откроем Mu и начнем писать программу для кофейни, — сказал Саймон. Эрик запустил Mu и увидел первое окно.

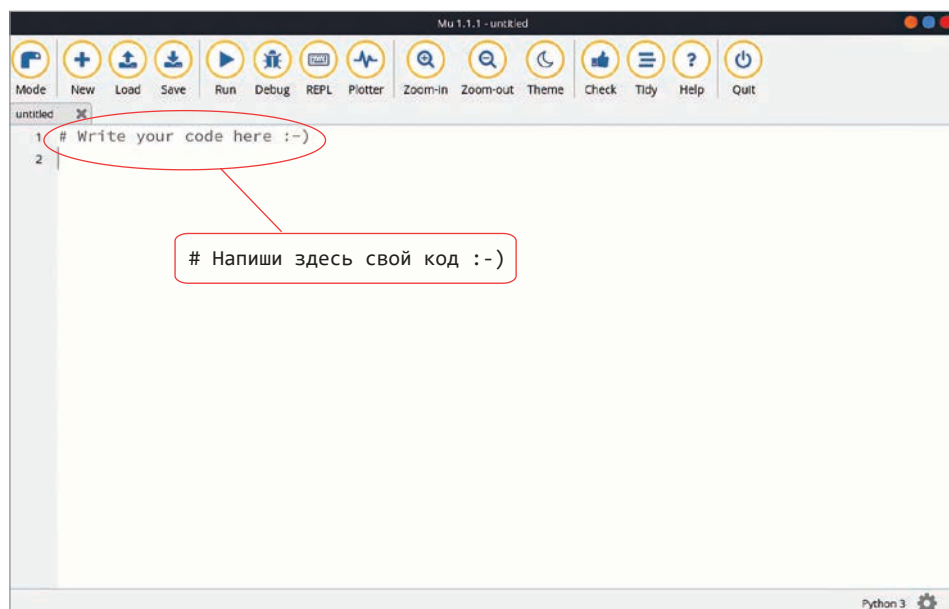


— В главном меню выбери Python 3 и нажми ОК, — предложил Саймон. Эрик так и сделал.

## 26 Поговорим на Python

— Теперь, — продолжил Саймон, — Ми запомнит, что ты работаешь с Python 3. Ты, наверное, заметил, что там есть и другие режимы, их можно использовать для работы с микроконтроллерами, создания веб-приложений и многого другого. Но нам пока нужен только Python 3.

Перед Эриком появилось окно редактора кода.



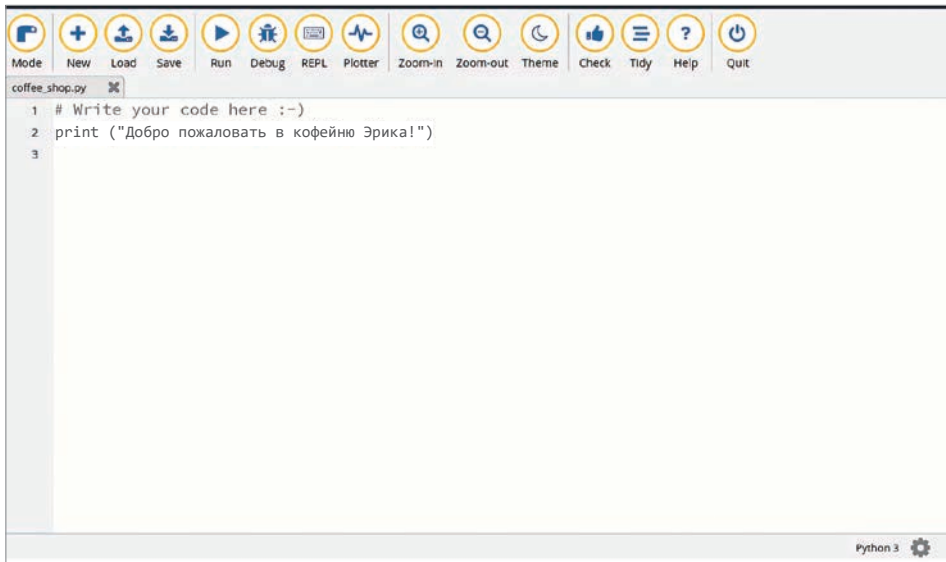
— Что мне здесь написать? — спросил Эрик.

— А что твоя программа должна сделать в первую очередь?

— Сказать: «Добро пожаловать в кофейню Эрика!»

— Отлично! Так и напиши. Помнишь функцию `print()` в Python?

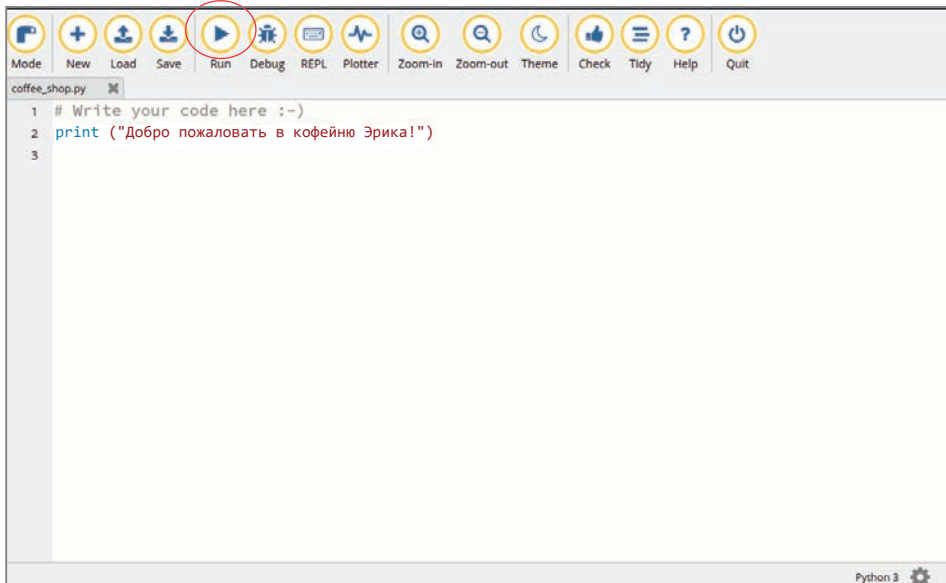
Эрик начал печатать. Первый шаг оказался простым.



— Что теперь?

— Теперь запусти. Нажми Run (Запустить).

Нажми Run

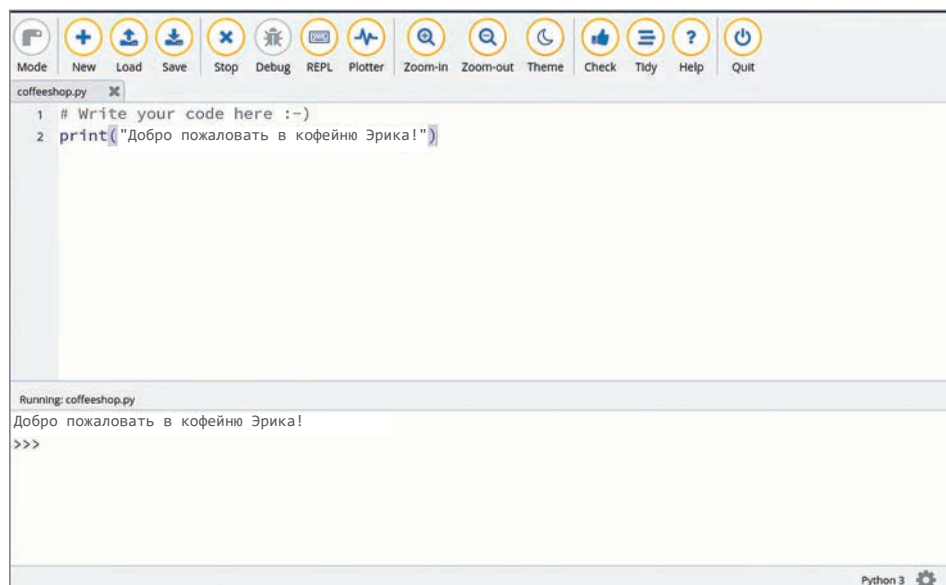


## 28 Поговорим на Python

Эрик нажал кнопку Run, и появилось другое окно, в котором спрашивалось, хочет ли он сохранить программу. Это тоже было легко. Эрик набрал имя файла — `coffeeshop` — и уже хотел нажать Enter, чтобы сохранить файл, но Саймон сказал:

— Стой... Не забудь добавить к имени файла расширение `.py`. Так текстовый редактор поймет, что это программа на Python. Му автоматически добавляет расширение, но другие редакторы этого не делают, так что убедись, что все твои файлы Python имеют расширение `.py`.

Эрик добавил `.py` к имени файла и сохранил его. Сразу после этого в нижней части редактора появилось другое окно. Это было приветствие от кофейни — все как хотел Эрик!



— Работает! — обрадовался Эрик.

— Конечно, работает. Почему бы ему не работать? — спросил Саймон. — Ты вроде хотел принимать заказы?

— Да, я хотел спрашивать у клиента, как его зовут и что он хочет заказать.

— А потом? — Саймон, конечно, уже знал ответ, но хотел, чтобы Эрик догадался сам.

— А потом на экране должно выводиться: «Привет! Ваш заказ», имя клиента, вкус, который он выбрал, и топпинг, как в настоящем счете.

— Отлично, — сказал Саймон. — Только смотри: когда ты пишешь программу, ты не знаешь, что закажет твой друг, так? Поэтому в программе ты не можешь написать: «Вы заказали карамель». К тому же разные люди заказывают разное. Алекс может заказать карамель, а Эмили — клубнику. Как видишь, для разных заказов вкусы и имя клиента *меняются*. Ничего это слово не напоминает из программирования?

— *Переменная!* — Эрик был страшно рад, что вспомнил это название из курса Python, который он прошел несколько месяцев назад.

— Точно! — обрадовался Саймон. — Переменная похожа на коробку: ты кладешь в нее что-то, а потом открываешь и смотришь, что там лежит. А еще можно заменить содержимое коробки чем-то другим. Мы создадим коробку с названием `answer` (ответ), она будет хранить то, что ты услышал от клиента. Ты спросишь у клиента его имя, и он, например, ответит: «Алекс». Тогда ты положишь этот ответ в коробку `answer` и будешь хранить его там. Когда ты захочешь вывести ответ на экран, ты скажешь Python: «Напечатай то, что сейчас находится в коробке с названием `answer`». Следующим клиентом может быть Эмили, а значит, ты положишь в коробку имя «Эмили». И в следующий раз Python выведет на экран уже не «Алекс», а «Эмили», потому что в коробке `answer` будет храниться «Эмили». Давай теперь напишем код.

— Прямо здесь? В этом же файле? — спросил Эрик.

— Конечно, продолжай в этом же файле. Чтобы получить что-то от клиента, мы будем использовать *функцию*, которая называется `input()`. Когда ты вызываешь эту функцию, компьютер ждет, что пользователь введет какие-то данные. Пользователь набирает что-нибудь на клавиатуре и нажимает Enter, а функция *возвращает* нам эти данные.

— Погоди, — остановил брата Эрик. — Что значит «возвращает»? И что это за *функции*? Нет, я, конечно, знаю, что это, но лучше объясни, что *ты* имеешь под этим в виду, — Эрик не хотел признаваться, что он не помнит о функциях почти ничего.

— Функция — это фрагмент кода, который что-то *делает*. Почти каждый код что-то делает, но какой-то код мы используем чаще. Позже ты напишешь свои функции, но пока возьмем те, что уже написаны другими людьми. Какие-то действия выполняются особенно часто, например вывод на экран. Ты даже не заметил, что уже использовал функцию, когда написал `print()`. В программировании это называется *вызвать* функцию.

— А, понятно, — сказал Эрик. — Вот это с круглыми скобками называется «функция».

## 30 Поговорим на Python

— Да. Если ты поместишь что-то внутри скобок, то функция что-то с этим сделает. Например, выведет на экран в виде сообщения. То, что ты передаешь функции, называется *аргументом*. Аргументы могут быть строками, числами, а иногда аргументов может быть несколько.

— Что такое строка? — спросил Эрик.

— Строка — это слово или несколько слов. Хорошо, что ты спросил, мы часто будем использовать строки.

— Мы говорим, что *передаем аргументы* функции, — продолжил Саймон. — Функция делает что-то с этими аргументами и получит какой-то *результат*. Например, она может что-то вычислить или изменить строку, которую ты передал, например привести ее к ВЕРХНЕМУ РЕГИСТРУ или зашифровать. Потом функция *возвращает* результат основной программе.

— А как я увижу результат? — спросил Эрик. — Функция выведет его на экран?

— Нет, не выведет. Тут нам как раз понадобятся *переменные*. Мы скажем Python: «Вызови такую-то функцию с такими-то аргументами и положи то, что она вернет, в такую-то коробку, точнее *переменную*». Все это можно сделать с помощью обычного знака «равно»: =. Например, если ты хочешь вызвать функцию `input()` и положить ее результат в переменную `answer`, ты просто пишешь так:

```
answer = input()
```

Сначала ты сохраняешь ответ клиента, а потом можешь вывести его на экран. Для этого вызови функцию `print()` и передай ей свою переменную в качестве аргумента.

— Отлично, — сказал Эрик. — Теперь я понял, как это написать.

Эрик начал печатать в редакторе и за пару минут написал вот это:

```
print("Добро пожаловать в кофейню Эрика!")  
  
answer = input()  
print(answer)
```

— Запустить? — спросил Эрик.

— Конечно, давай, жми Run .

Эрик нажал Run .

— Тут написано: «Добро пожаловать в кофейню Эрика!» и все.

— А ты что хотел? — спросил Саймон.

— Чтобы программа спросила, как меня зовут.

— Но ты же не сказал Python, что он должен что-то спросить. Теперь он ждет, когда ты что-нибудь введешь. Напиши что-нибудь.

Эрик набрал на клавиатуре «Эрик» и нажал Enter.

Python вывел на экран Эрик.

```

1 # Write your code here :-)
2 print("Добро пожаловать в кофейню Эрика!")
3
4 answer = input()
5 print(answer)

```

Running: coffeshop.py  
Добро пожаловать в кофейню Эрика!  
Эрик  
Эрик  
>>>

— Работает! — воскликнул Эрик.

### **ТВОЙ ХОД! Напиши свой первый диалог**

Напиши такую же диалоговую программу, как у Эрика. Это короткая программа, так что я рекомендую не копировать ее из книги, а набрать самостоятельно. Придумай название кофейни и используй его в приветствии. Или придумай любой магазин, какой захочешь. Что ты будешь продавать? Мороженое? Пиццу? Цветы? Игрушки для животных?

Попробуй запустить свою программу. Делает ли она то, что ты ожидал? Если нет, то скопируй код из книги или с сайта (<https://github.com/pavelanni/pythonicadventure-code/tree/main/ch01>) и запусти снова. Программа должна заработать.