

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В.Н. Татищева»
(Астраханский государственный университет им. В.Н. Татищева)

Кафедра английской филологии

Письменный перевод

по книге Learn to code by solving problems

название книги на иностранном языке

выходные данные No Starch Press, 2021

(место издания, год)

перевод стр. с xxi по 32

для сдачи кандидатского экзамена
по иностранному языку
(английский)

Выполнил:

Шведова Светлана Витальевна

Кафедра цифровых технологий

Астрахань – 2023 г.

Introduction

We use computers to accomplish tasks and to solve problems.

For example, perhaps you've used a word processor to write an essay or letter.

Perhaps you've used a spreadsheet program to organize your finances.

Perhaps you've used an image editor to touch up a picture.

It's hard to imagine doing these things these days without a computer.

We get a lot of mileage out of our word processors and spreadsheet programs and image editors.

Those programs are written as general-purpose tools to accomplish a wide variety of tasks.

Ultimately, though, they're programs written by others, not by us.

What do we do when an off-the-shelf program doesn't quite do what we need?

In this book, our goal is to learn how to take control of our computer by going

Вступление

Компьютеры используются для выполнения задач и решения проблем.

Например, возможно, вы использовали текстовый редактор для написания эссе или письма.

Возможно, вы использовали программу для работы с электронными таблицами, чтобы систематизировать свои финансы.

Возможно, вы использовали графический редактор, чтобы отредактировать картинку.

В наши дни трудно представить, как можно заниматься подобными вещами без компьютера.

Текстовый редактор, программы для работы с электронными таблицами и графических редакторов очень полезны в повседневной жизни.

Данные программы написаны как инструменты общего назначения для выполнения широкого спектра задач.

В конечном счете, однако, данные программы написаны другими специалистами, а не нами.

Что мы делаем, когда готовая программа не соответствует нашим ожиданиям?

В данной книге основной целью является научиться управлять нашим компьютером, выходя за рамки того, что может сделать

beyond what can be done by an end user using preexisting programs.	конечный пользователь с помощью уже существующих программ.
We're going to write our own programs.	Мы собираемся писать наши собственные программы.
We won't write a word processor or a spreadsheet or an image editor.	Мы не будем писать текстовый редактор, электронную таблицу или графический редактор.
Those are huge tasks that, fortunately, people have already done.	Данные глобальные задачи, к счастью, уже решены.
Rather, we're going to learn how to write small programs to solve problems that we otherwise wouldn't be able to solve.	Скорее всего, мы собираемся научиться писать небольшие программы для решения проблем, которые мы бы не смогли решить при их отсутствии.
I want to help you learn to communicate instructions to a computer; these instructions will tell the computer how to carry out your plan for solving a problem.	Я хочу помочь вам научиться давать инструкции компьютеру; данные инструкции подскажут компьютеру, каким образом, согласно намеченному плану, решить вашу проблему.
To give instructions to a computer, we write code in a programming language.	Чтобы давать инструкции компьютеру, мы пишем код на языке программирования.
A programming language specifies the rules for the code we write and dictates what the computer does in response to that code.	Язык программирования определяет правила для кода, который мы пишем, и диктует, что компьютер делает в ответ на данный код.
We're going to learn to program in the Python programming language.	Мы собираемся научиться программировать на языке программирования Python.
That's a concrete skill that you'll take away from this book, one that you can put on your résumé.	Данный конкретный навык, который вы обретете после прочтения данной книги и который сможете указать в своем резюме.

More than Python, though, you'll learn the type of thinking required to solve problems using a computer.	Однако, помимо Python, вы изучите тип мышления, необходимый для решения проблем при помощи компьютера.
Programming languages come and go.	Языки программирования появляются и исчезают.
The way that we solve problems does not.	Однако, проблемы остаются.
I hope that this book helps you on your way from end user to programmer and that you have fun exploring what's possible.	Я надеюсь, что данная книга поможет вам на вашем пути от конечного пользователя к программисту и что вы получите удовольствие, исследуя предоставляемые вам возможности.
Online Resources	Онлайн-ресурсы
Supplementary resources for the book, including downloadable code and additional exercises, are available at https://nostarch.com/learn-code-solving-problems/	Дополнительные ресурсы для книги, включая загружаемый код и дополнительные упражнения, доступны по адресу https://nostarch.com/learn-code-solving-problems/
Who This Book Is For	Для кого предназначена данная книга?
This book is for anyone who wants to learn how to write computer programs to solve problems.	Данная книга будет полезна всем, кто хочет научиться писать компьютерные программы для решения определенных задач.
I have three particular types of people in mind.	Я имею в виду три конкретных типа людей.
First, you may have heard about the Python programming language and want to learn how to write code in Python.	Во-первых, возможно, вы слышали о языке программирования Python и хотите научиться писать код на Python.

I'll explain in the next section why Python is a great choice as a first programming language to learn.

You'll learn a lot about Python in this book, and you'll be in a position to read more advanced books on Python if that's your next step.

Second, if you haven't heard of Python or just want to learn what programming is all about, don't worry: this book is for you, too!

This book will teach you how to think about programming.

Programmers have particular ways of breaking down problems into manageable pieces and expressing solutions to those pieces in code.

At this level, it doesn't matter what programming language is being used, because the way that programmers think is not tied to a particular language.

Finally, you may be interested in learning some other programming language such as C++, Java, Go, or Rust.

Much of what you learn as a byproduct of learning Python will be useful when

В следующем разделе я объясню, почему Python представляет собой отличный выбор в качестве первого языка программирования для изучения.

Из данной книги вы многое узнаете о Python и сможете читать более продвинутые книги по Python, если решите перейти на следующий уровень.

Во-вторых, если вы никогда не слышали о Python или просто хотите узнать, что такое программирование, не волнуйтесь: данная книга тоже для вас!

Данная книга научит вас понимать процесс программирования.

У программистов есть особые способы делить проблемы на управляемые части и представлять решения данных частей в виде кода.

На данном уровне не имеет значения, какой язык программирования используется, потому что мышление программистов не привязано к определенному языку.

Наконец, вам может быть интересно изучить какой-нибудь другой язык программирования, такой как C++, Java, Go или Rust.

Многое из того, что вы узнаете дополнительно, помимо изучения Python,

you study those other programming languages. будет полезно при изучении вышеперечисленных других языков программирования.

Plus, Python is certainly worth learning in its own right. Let's turn to why next. Кроме того, Python, безусловно, стоит изучать сам по себе. Давайте перейдем к следующему вопросу «с какой целью».

Why Learn Python? Зачем изучать Python?

Years of teaching introductory programming have demonstrated to me that Python is a great choice for a first programming language. Годы преподавания вводного программирования показали мне, что Python - отличный выбор для первого языка программирования.

Compared to other languages, Python code is often more structured and readable. По сравнению с другими языками код на Python часто более структурирован и удобочитаем.

Once you get used to it, you might agree that parts of it almost read like English! Как только вы привыкнете к нему, вы, возможно, согласитесь, что некоторые его части читаются почти как текст на английском языке!

Python also comes with many features that are not available in other languages, including powerful tools to manipulate and store data. Python также обладает многими функциями, недоступными на других языках, включая мощные для обработки и хранения данных)

We'll use many of these features throughout the book. Мы будем использовать многие из данных функций на протяжении всей книги.

Not only is Python an excellent teaching language, but it's also one of the most in-demand programming languages in the world. Python не только является отличным языком обучения, но и одним из самых востребованных языков программирования в мире.

Programmers use it to write web applications, games, visualizations, machine learning software, and more.

There we have it: a language well-suited for teaching that also carries a professional advantage for you.

I can't ask for any more than that!

Installing Python

Before we can program in Python, we need to install it.

Let's do that now.

Python 3 was a major evolution from Python 2, but Python continues to change even within version 3.

The first version of Python 3 was Python 3.0.

Then Python 3.1 was released, then Python 3.2, and so on.

At the time of writing, the latest version of Python 3 is Python 3.9.

Versions as old as Python 3.6 will suffice for this book, but I encourage you to install and work with the latest version of Python.

Follow these steps for your operating system to install Python.

Windows

Программисты используют его для написания веб-приложений, игр, визуализаций, программного обеспечения для машинного обучения и многого другого.

Итак, что мы имеем: язык, хорошо подходящий для обучения, предоставляющий вам профессиональные преимущества.

Большого сложно даже представить!

Установка Python

Прежде чем мы сможем программировать на Python, нам нужно установить его.

Давайте сделаем это сейчас.

Python 3 был значительно усовершенствован по сравнению с Python 2, но Python продолжает меняться даже в версии 3.

Первой версией Python 3 был Python 3.0.

Затем был выпущен Python 3.1, затем Python 3.2 и так далее.

На момент написания последней версией Python 3 является Python 3.9.

Для данной книги будет достаточно таких старых версий, как Python 3.6, но я рекомендую установить последнюю версию Python и работать с ней.

Выполните следующие действия для установки Python в вашей операционной системе.

Операционная система Windows

Windows doesn't come with Python by default.

To install it, go to <https://www.python.org/> and click Downloads.

This should offer you the option of downloading the latest version of Python for Windows.

Click the link to download Python and then run the installer.

On one of the first screens in the installation process, click either Add Python 3.9 to PATH or Add Python to environment variables; this makes running Python much easier.

(If upgrading Python, you may need to click "Customize installation" to find this option.)

macOS

macOS doesn't come with Python 3 by default.

To install it, go to <https://www.python.org/> and click Downloads.

This should offer you the option of downloading the latest version of Python for macOS.

Click the link to download Python and then run the installer.

Windows по умолчанию не поставляется с Python.

Чтобы установить его, перейдите по ссылке <https://www.python.org/> и нажмите кнопку «Загрузки».

Далее в диалоговом окне вам будет предложено загрузить последнюю версию Python для Windows.

Нажмите на ссылку, чтобы загрузить Python, а затем запустите программу установки.

На одном из первых экранов в процессе установки нажмите либо «Добавить Python 3.9 в ДИРЕКТОРИЮ», либо «Добавить» Python выбранную вами директорию; это значительно упрощает запуск Python.

(При обновлении Python вам может потребоваться нажать «Настроить установку», чтобы найти данную опцию.)

Операционная система macOS

macOS по умолчанию не поставляется с Python 3.

Чтобы установить его, перейдите по ссылке <https://www.python.org/> и нажмите кнопку «Загрузки».

Появляется диалоговое окно с возможностью загрузить последнюю версию Python для macOS.

Нажмите на ссылку, чтобы загрузить Python, а затем запустите программу установки.

Linux

Linux comes with Python 3, but it may be an older version of Python 3.

Installation instructions will vary depending on which Linux distribution you're using, but you should be able to install the newest version of Python using your favorite package manager.

How to Read This Book.

Reading this book from cover to cover in one sitting is likely to teach you very little.

It would be like trying to learn piano by inviting someone into your house to play piano for a few hours and then kicking them out, lowering the lights, and serenading.

That's not how a practice-based skill is learned.

Here is my advice for making your way through the book:

Space out your work.

Massing your practice into a small number of sessions is far less effective than spacing your practice out.

When you feel tired, take a break.

Операционная система Linux

Linux поставляется с Python 3, но это может быть более старая версия Python 3.

Инструкции по установке будут отличаться в зависимости от того, какой дистрибутив Linux вы используете, но вы должны иметь возможность установить новейшую версию Python с помощью вашего любимого менеджера распаковки.

Как читать данную книгу?

Прочтение данной книги полностью за один раз, скорее всего, мало чему вас научит.

Это было бы похоже на попытку научиться играть на фортепиано, пригласив кого-то в свой дом поиграть на фортепиано в течение нескольких часов, а затем, когда он уйдет, приглушить свет и исполнить серенаду.

Практико-ориентированное обучение происходит не таким образом.

Мой совет по прочтению данной книги заключается в следующем:

Выделите место для своей работы.

Деление практических занятий на небольшое количество сеансов гораздо менее эффективно, чем разделение их на более длительные периоды.

Когда вы почувствуете усталость, сделайте перерыв.

No one can tell you how much time to work before taking a break.	Никто не может сказать вам, сколько времени нужно поработать, прежде чем сделать перерыв.
No one can tell you how long it should take you to finish the book.	Никто не может сказать вам, сколько времени вам потребуется, чтобы дочитать данную книгу до конца.
It's up to your own mind and body.	Это зависит от ваших умственных и физических возможностей.
Pause to test your understanding.	Сделайте паузу, чтобы проверить усвоенный материал.
Reading about something can give us the illusion that we understand it better than we do.	Чтение о чем-то может создать у нас иллюзию, что мы понимаем материал лучше, чем на самом деле.
Applying the material forces what we know and what we think we know into alignment.	Использование физических способностей приводит в соответствие то, что мы знаем, с тем, что мы думаем, что знаем.
For that reason, at key points in each chapter, I've included multiple-choice "concept check" questions that ask you to make a prediction.	По данной причине в ключевые моменты каждой главы я включил вопросы для проверки концепции с несколькими вариантами ответов, которые вам необходимо выбрать.
Take these seriously!	Отнеситесь к этому серьезно!
Read each question and commit to a response without checking anything using your computer.	Прочитайте каждый вопрос и зафиксируйте ответ, ничего не проверяя с помощью своего компьютера.
Then, read my answer and explanation.	Затем прочтите мой ответ и объяснение.
This is an opportunity to confirm that you're on the right track.	Данный процесс является возможностью подтвердить, что вы на правильном пути.

<p>If you answer incorrectly or answer correctly but for the wrong reason, take time to remedy your understanding before continuing.</p> <p>This could involve playing around a little more with the relevant Python feature being discussed, rereading material from the book, or searching for additional explanations and examples online.</p> <p>Practice programming.</p> <p>Making predictions while you read will help solidify your understanding of key concepts.</p> <p>But you need more than that to become an adept problem-solver and programmer.</p> <p>You need to practice using Python to solve new problems, whose solutions you haven't read about in the book. Each chapter concludes with a list of practice exercises.</p> <p>Please work through as many of these exercises as you can.</p> <p>Learning how to program takes time. Don't be discouraged if you progress slowly or make a lot of mistakes.</p>	<p>Если вы отвечаете неправильно или отвечаете правильно, но по неправильной причине, найдите время, чтобы еще раз ознакомиться с материалом, прежде чем продолжить изучение.</p> <p>Вы можете еще несколько раз проиграть использование соответствующей версии Python, изучить функции Python, еще раз перечитать материала книги или найти дополнительные объяснения и примеры в интернете.</p> <p>Практикуйтесь в программировании.</p> <p>Составление прогнозов во время чтения поможет укрепить ваше понимание ключевых концепций.</p> <p>Но вам нужно нечто большее, чтобы искусно решать поставленные задачи и стать программистом.</p> <p>Вам нужно попрактиковаться в использовании Python для решения новых задач, о решениях которых вы не читали в книге.</p> <p>Каждая глава завершается списком практических упражнений.</p> <p>Пожалуйста, выполняйте как можно больше данных упражнений.</p> <p>Обучение программированию требует времени.</p>
---	--

Don't be intimidated by any blustering peacocks that you might encounter online.

Не расстраивайтесь, если вы продвигаетесь медленно или допускаете много ошибок.

Не пугайтесь никаких «псевдо» суперпрограммистов, с которыми вы можете

Surround yourself with people and resources that can help you learn.

столкнуться в Интернете.

Окружите себя людьми и ресурсами, которые могут помочь вам учиться.

Using Programming Judges

Использование хостинга для проектов

I've decided to structure this book around problems from programming judge websites.

Я решил структурировать данную книгу в соответствии с задачами, связанными с сайтами-хранилищами для хранения архивов различных версий проектов.

A programming judge website offers a repository of programming problems that can be solved by programmers around the world.

Сайт-хранилище предлагает репозиторий для задач в программировании, которые могут быть решены программистами по всему миру.

You submit your solution – your Python code – and the site runs tests on your code.

Вы размещаете свое решение – свой код на Python – и сайт запускает тестирование вашего кода.

If your code produces the correct answer for each test case, then it's likely that your solution is correct.

Если ваш код выдает правильный ответ для каждого тестового примера, то, скорее всего ваше решение правильное.

If instead your code produces the wrong answer for one or more test cases, then your code is incorrect, and revisions are required.

Если вместо этого ваш код выдает неправильный ответ для одного или нескольких тестовых случаев, значит, ваш код неверен и требуются изменения.

There are several reasons why I think programming judges are particularly

Есть несколько причин, по которым я считаю, что онлайн-хранилища кода представляют

suitable sites for learning programming:	собой соответствующие сайты для обучения программированию:
Rapid feedback is crucial in the early stages of learning to program.	Быстрая обратная связь. Быстрая, целенаправленная обратная связь имеет решающее значение на ранних стадиях обучения программированию.
Programming judges provide feedback as soon as you submit your code.	Онлайн-хранилища кода предоставляют обратную связь, как только вы разместите свой код.
High-quality problems.	Проблемы высокого качества.
I find the problems on programming judges to be high quality.	Я нахожу, что задачи онлайн-хранилища заключаются в предоставлении материалов высокого качества.
Many problems are originally from competitive programming contests.	Многие задачи изначально взяты из конкурсных соревнований программистов.
Other problems are written by individuals who are associated with a programming judge or simply want to help others learn.	Другие задачи написаны людьми, которые связаны хранилищами кода или просто хотят помочь другим учиться.
See the Problem Credits appendix for the source of each problem that we'll study.	Обратитесь к приложению «Примерные задачи» для получения доступа к источнику описания каждой изученной проблемы.
Quantity of problems.	Количество проблем.
The programming judges contain hundreds of problems.	В хранилищах кода содержатся сотни задач.
I've selected only a small number for this book.	Я отобрал для данной книги лишь небольшое количество.
If you need more practice, trust me: the programming judges can provide it.	

Community features.

The programming judges enable users to read and respond to comments.

If you're stuck on a problem, browse the comments for hints dropped there by others.

If you're still stuck, consider posting a comment of your own asking for help.

Once you've successfully solved a problem, your learning is not done!

Many programming judges allow you to view the code that others have submitted.

Dig through a few such submissions to see how they compare to yours.

There are always multiple ways to solve a problem.

Perhaps your way is most intuitive to you right now, but opening yourself to other possibilities is an important step toward programming mastery.

Making Your Programming Judge Accounts

Если вам нужно больше практики, поверьте мне: хранилища кода помогут решить данную проблему.

Особенности сообщества.

Хранилища кода позволяют пользователям прочитать комментарии и ответить на них.

Если вы застряли на какой-то проблеме, просмотрите комментарии в поисках подсказок, оставленных там другими.

Если вы все же не нашли решения, подумайте о том, чтобы опубликовать свой собственный комментарий с просьбой о помощи.

После успешного решения проблемы, обучение еще не закончено!

Многие хранилища кода позволяют вам просматривать код, представленный другими.

Просмотрите несколько таких материалов, чтобы увидеть, как они соотносятся с вашими.

Всегда есть несколько способов решения проблемы.

Возможно, прямо сейчас ваш путь вам наиболее интуитивно понятен, но открытие для себя других возможностей - важный шаг к овладению программированием.

Создание учетных записей в онлайн-хранилищах кода.

We'll use several programming judge websites throughout the book.

That's because each programming judge hosts some problems that are not found on other programming judges; we need multiple programming judges to cover all of the problems that I've chosen.

Here are the programming judges that we'll use:

Each programming judge requires that you create an account before you can submit code.

Let's go through the process of creating your accounts now and learn a little about the judges while we're at it.

The DMOJ Judge

The DMOJ judge is the judge we'll use most often in the book.

More than for any other judge, it's worth your time exploring the DMOJ website and learning about what the judge offers.

To create an account on the DMOJ judge, go to <https://dmoj.ca/> and click Sign up.

На протяжении всей книги мы будем пользоваться несколькими сайтами хранилищ программного кода.

Это потому, что у каждого хранилища программного кода есть некоторые проблемы, которых нет у других хранилищ программного кода; нам нужно несколько хранилищ программного кода, чтобы охватить все выбранные мною задачи.

Вот хранилища программного кода, которые мы будем использовать:

Каждое хранилище программного кода требует создания учетной записи до отправления кода.

Давайте пройдемся по процессу создания ваших аккаунтов прямо сейчас и немного узнаем о хранилищах, пока мы этим занимаемся.

Хранилище DMOJ

Хранилище DMOJ — это хранилище программного кода, которое мы будем чаще всего использовать в книге.

Больше, чем для любого другого хранилища, вам стоит потратить время на изучение веб-сайта DMOJ и узнать о том, что предлагает хранилище.

Чтобы создать учетную запись на хранилище DMOJ, перейдите по <https://dmoj.ca/> и нажмите «Зарегистрироваться».

On the registration page that comes up, enter your username, password, and email address.

This page also allows you to set your default programming language.

We'll exclusively be using the Python programming language in this book, so I suggest clicking Python 3 here. Click Register! to create your account.

Once registered, you can use your username and password to log in to DMOJ.

Each problem in the book begins by indicating the judge website where the problem can be found and the problem code that you should use to access it.

For example, the first problem that we'll work on in Chapter 1 is found on DMOJ and is problem `dmopc15c7p2`.

To find this problem on DMOJ, click Problems, type `dmopc15c7p2` in the search box, and click Go.

You should see the problem as the only result.

If you click the problem title, you should see the problem itself.

На открывшейся странице регистрации введите свое имя пользователя, пароль и адрес электронной почты.

Данная страница также позволяет вам установить язык программирования по умолчанию.

В данной книге мы будем использовать исключительно язык программирования Python, поэтому я предлагаю нажать здесь на Python 3.

Нажмите «Зарегистрироваться!», чтобы создать свою учетную запись.

После регистрации вы можете использовать свое имя пользователя и пароль для входа в DMOJ.

Каждая задача в книге начинается с указания веб-сайта хранилища, на котором можно найти решение, и код проблемы, который вы должны использовать для доступа к нему.

Например, первая проблема, над которой мы будем работать в главе 1, находится на DMOJ и является проблемой `dmopc15c7p2`.

Чтобы найти данную проблему на DMOJ, нажмите «Проблемы», введите `dmopc15c7p2` в поле поиска и нажмите «Перейти».

Вы должны будете увидеть задачу как единственный результат.

Если вы нажмете на заголовок задачи, вы должны увидеть саму проблему.

When you're ready to submit Python code for a problem, find the problem and click Submit solution.

On the resulting page, paste your Python code into the text box and click Submit!

Your code will then be judged and the results shown.

The Timus Judge

To create an account on the Timus judge, go to <https://acm.timus.ru/> and click Register.

On the registration page that comes up, enter your name, password, email address, and other requested information.

Click Register to create your account.

Then, check your email for a message from Timus containing your judge ID.

You'll need your judge ID whenever you submit Python code.

There's currently no way to set your default programming language, so be sure to choose the available version of Python 3 whenever you submit Python code.

Когда вы будете готовы отправить код Python для решения задачи, найдите задачу и нажмите «Отправить решение».

На полученной странице вставьте свой код Python в текстовое поле и нажмите «Отправить»!

Затем ваш код будет оценен и будут показаны результаты.

Хранилище «Тимус»

Чтобы создать учетную запись на хранилище Timid, перейдите по <https://acm.timus.ru/> и нажмите кнопку «Зарегистрироваться».

На открывшейся странице регистрации введите свое имя, пароль, адрес электронной почты и другую запрашиваемую информацию.

Нажмите «Зарегистрироваться», чтобы создать свою учетную запись.

Затем проверьте свою электронную почту на наличие сообщения от Timus, содержащего ваш идентификатор хранилища.

Вам понадобится ваш идентификатор хранилища всякий раз, когда вы отправляете код на Python.

В настоящее время нет возможности установить язык программирования по умолчанию, поэтому обязательно выбирайте доступную версию Python 3 всякий раз, когда отправляете код на Python.

We use the Timus judge only once, in Chapter 6, so I won't say any more about it here.

The USACO Judge

To create an account on the USACO judge, go to <http://usaco.org/> and click Register for New Account.

On the registration page that comes up, enter your username, email address, and other requested information.

Click Submit to create your account.

Then, check your email for a message from USACO containing your password.

Once you have your password, you can use your username and password to log in to USACO.

There's currently no way to set your default programming language, so be sure to choose the available version of Python 3 whenever you submit Python code.

You'll also need to choose the file containing your Python code rather than paste your code into a text box.

Мы используем хранилище Timus только один раз, в главе 6, поэтому я больше не буду говорить об этом здесь.

Хранилище USACO

Чтобы создать учетную запись на сайте хранилища USACO, перейдите по <http://usaco.org/> и нажмите «Зарегистрироваться» для новой учетной записи.

На открывшейся странице регистрации введите свое имя пользователя, адрес электронной почты и другую запрашиваемую информацию.

Нажмите кнопку «Отправить», чтобы создать свою учетную запись.

Затем проверьте свою электронную почту на наличие сообщения от USACO, содержащего ваш пароль.

Как только у вас будет свой пароль, вы можете использовать свое имя пользователя и пароль для входа в USACO.

В настоящее время нет возможности установить язык программирования по умолчанию, поэтому обязательно выбирайте доступную версию Python 3 всякий раз, когда отправляете код на Python.

Вам также нужно будет выбрать файл, содержащий ваш код на Python, а не вставлять его в текстовое поле.

We won't be using the USACO judge until Chapter 7, so I won't say any more about it here.

About This Book

Each chapter in the book is driven by two or three problems from a programming judge website.

In fact, I start each chapter by posing our first problem, before teaching any new Python at all!

My goal in doing this is to motivate you to want to learn the Python features that we need to solve the problem.

Don't worry if you're not sure how to solve a problem after reading its description.

(If you can't solve the problem yet, then you're reading the right book!)

If you understand what the problem is asking you to do, then you're all set.

We're going to learn Python and solve the problem together.

Subsequent problems in the chapter may introduce further Python features or ask us to extend what we learned in the first problem.

Мы не будем использовать хранилище USACO до главы 7, поэтому я не буду больше говорить об этом здесь.

Об этой книге.

Каждая глава в книге основана на двух или трех задачах с веб-сайта хранилища программного кода.

На самом деле, я начинаю каждую главу с постановки нашей первой проблемы, прежде чем вообще учить какой-либо новый Python!

Моя цель при этом - побудить вас к изучению функции Python, которые нам нужны для решения задачи.

Не волнуйтесь, если вы не уверены, как решить задачу, прочитав ее описание.

(Если вы пока не можете решить задачу, значит, вы читаете правильную книгу!)

Если вы понимаете, каким образом решить задачу, тогда у вас все готово.

Мы собираемся выучить Python и решить задачу вместе.

Последующие задачи в данной главе могут ввести дополнительные возможности Python или попросить нас расширить то, что мы узнали в первой задаче.

Каждая глава завершается упражнениями, которые вы должны решить самостоятельно,

Each chapter concludes with exercises that you should solve on your own to practice what you just learned.	чтобы применить на практике то, что вы только что узнали.
Getting started.	Приступая к работе.
Programming involves writing code to solve a problem.	Программирование включает в себя написание кода для решения проблемы.
As such, I want to solve problems with you from the outset.	Таким образом, я хочу решить проблемы с вами с самого начала.
That is, rather than learning Python concept by concept and then solving a problem, we're going to use a problem to dictate the concepts we need to learn.	То есть, вместо того, чтобы изучать Python концепцию за концепцией, а затем решать проблему, мы собираемся использовать проблему, чтобы диктовать концепции, которые нам нужно изучить.
In this chapter, we'll solve two problems: determining the number of words in a line (like the word-count feature in a word processor) and calculating the volume of a cone.	В данной главе мы решим две задачи: определим количество слов в строке (подобно функции подсчета слов в текстовом процессоре) и вычислим объем конуса.
Solving these problems requires a tour of quite a few Python concepts.	Решение этих задач требует ознакомления с довольно многими концепциями Python.
You may feel that you need more details to fully understand some of what I introduce here and how it all fits together in the design of a Python program.	Вы можете почувствовать, что вам нужно больше деталей, чтобы полностью понять некоторое из того, что я здесь представляю, и то, как все это сочетается в дизайне программы на Python.
Don't worry: we'll revisit and elaborate on the most important concepts in later chapters.	Не волнуйтесь: мы вернемся к наиболее важным концепциям и подробно остановимся на них в последующих главах.
What We'll Be Doing	Что мы будем делать?

<p>As described in the introduction, we'll be solving competitive programming problems using the Python programming language.</p> <p>The competitive programming problems can each be found on an online judge website.</p> <p>I assume that you've followed the instructions in the introduction to install Python and make your judge accounts.</p> <p>For each problem, we'll write a program to solve it.</p> <p>Each problem specifies the kind of input that our program will be provided, and the kind of output (or result) that is expected.</p> <p>Our program correctly solves the problem if it can take any valid input and produce the correct output.</p> <p>In general, there will be millions or billions of possible inputs.</p> <p>Each such input is referred to as a problem instance.</p> <p>For example, in the first problem that we'll solve, the input is a line of text, like hello there or bbaabbb aa abab.</p>	<p>Как описано во введении, мы будем решать задачи соревновательного программирования с использованием языка программирования Python.</p> <p>С каждой из задач соревновательного программирования можно ознакомиться на веб-сайте онлайн-хранилища.</p> <p>Я предполагаю, что вы следовали инструкциям во введении, чтобы установить Python и создать свои учетные записи на хранилищах.</p> <p>Для каждой задачи мы напишем программу для ее решения.</p> <p>Каждая задача определяет тип входных данных, которые будут предоставлены нашей программе, и тип выходных данных (или результата), который ожидается.</p> <p>Наша программа правильно решает задачу, если она может принимать любые допустимые входные данные и выдавать правильные выходные данные.</p> <p>В общем, будут миллионы или миллиарды возможных входных данных.</p> <p>Каждый такой ввод упоминается как экземпляр задачи.</p> <p>Например, в первой задаче, которую мы решим, вводом является строка текста, например, «hello there» или «bbaabbb aa abab».</p>
--	---

Our task will be to output the number of words in the line.	Нашей задачей будет вывести количество слов в строке.
One of the most powerful ideas in programming is that often a small amount of general-purpose code can solve a seemingly endless number of problem instances.	Одна из самых мощных идей в программировании заключается в том, что часто небольшой объем кода общего назначения может решить, казалось бы, бесконечное количество примеров задач.
Whether the line has 2 words or 3 or 50, it won't matter.	Будет ли в строке 2 слова, или 3, или 50, это не будет иметь значения.
Our program will get it right every time.	Наша программа каждый раз сделает все правильно.
Our programs will perform three tasks:	Наши программы будут выполнять три задачи:
Read input.	Чтение входных данных.
We need to determine the specific instance of the problem that we're solving, so we first read the provided input.	Нам нужно определить конкретный экземпляр задачи, которую мы решаем, поэтому мы сначала читаем предоставленные входные данные.
Process.	Процесс.
We process the input to determine the correct output.	Мы обрабатываем входные данные, чтобы определить правильный вывод.
Write output.	Запись выходных данных.
Having solved the problem, we produce the desired output.	Решив проблему, мы получаем желаемый результат.
The boundaries between these steps may not always be crisp – we might have to interleave some processing with producing some output, for	Границы между этими этапами не всегда могут быть четкими – например, нам, возможно, придется чередовать некоторую обработку с получением некоторого

example – but it will be helpful to keep these three broad steps in mind.	результата, – но будет полезно помнить об этих трех общих шагах.
You likely use programs on a daily basis that follow this input-process-output model.	Скорее всего, вы ежедневно используете программы, которые следуют этой модели ввода-вывода.
Consider a calculator program: you type in a formula (the input), the program crunches your numbers (process), and the program displays the answer (output).	Рассмотрим программу-калькулятор: вы вводите формулу (ввод), программа вычисляет ваши цифры (обработка), и программа отображает ответ (вывод).
Or consider a web search engine: you type in a search query (input), the search engine determines the most relevant results (process), and it displays them (output).	Или рассмотрим поисковую систему в Интернете: вы вводите поисковый запрос (ввод), поисковая система определяет наиболее релевантные результаты (обрабатывает) и отображает их (вывод).
Contrast these kinds of programs with interactive programs, which fuse input, processing, and output.	Сравните эти виды программ с интерактивными программами, которые объединяют ввод, обработку и вывод.
For example, I’m typing this book using a text editor.	Например, я печатаю эту книгу с помощью текстового редактора.
When I type a character, the editor responds by adding that character to my document.	Когда я набираю символ, редактор отвечает, добавляя этот символ в мой документ.
It doesn’t wait for me to type the entire document before displaying it to me; it interactively displays it as I build it.	Он не ждет, пока я введу весь документ целиком, прежде чем отображать его мне; он интерактивно отображает его по мере того, как я его создаю.
We won’t be writing interactive programs in this book.	В этой книге мы не будем писать интерактивные программы.

If you're interested in writing such programs after studying this book, you'll be happy to hear that Python is certainly up for the task.	Если вы заинтересованы в написании таких программ после изучения данной книги, вы будете рады услышать, что Python, безусловно, подходит для данной задачи.
The text for each problem is found both here and on the online judge.	Текст для каждой задачи можно найти как здесь, так и на онлайн-хранилище.
However, the text won't match, because I've rewritten it for purposes of consistency throughout the book.	Однако текст не будет соответствовать, потому что я переписал его в целях согласованности на протяжении всей книги.
Don't worry: what I've written conveys the same information as the official problem statement.	Не волнуйтесь: то, что я написал, содержит ту же информацию, что и официальное описание задачи.