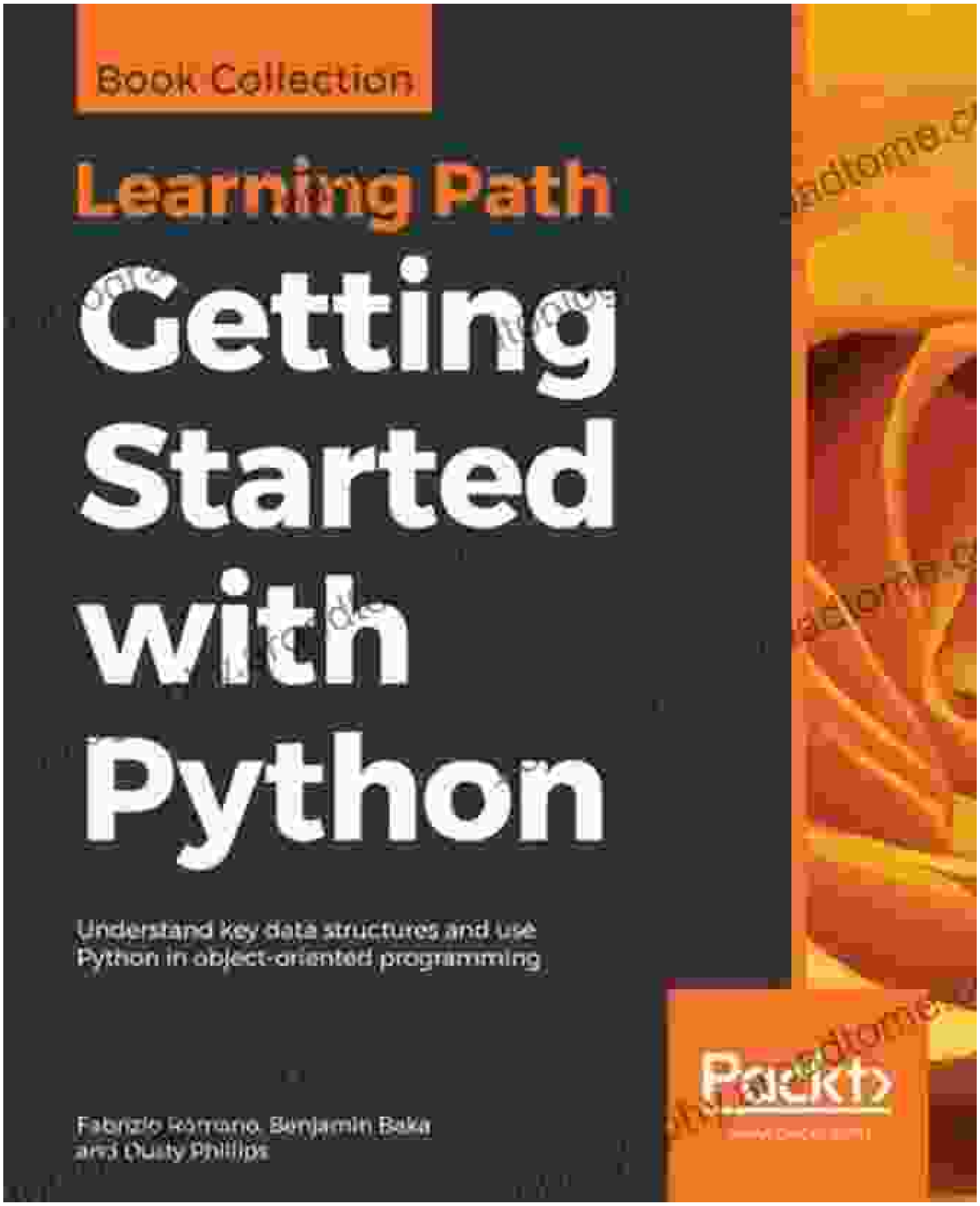
Getting Started With Python: The Ultimate Guide for Beginners

Getting Started with Python: Understand key data structures and use Python in object-oriented programming

★★★★★ 5 out of 5

Language	: English
File size	: 12222 KB
Text-to-Speech	: Enabled
Screen Reader	: Supported
Enhanced typesetting	: Enabled
Print length	: 724 pages


Book Collection
Learning Path
Getting Started with Python
Understand key data structures and use Python in object-oriented programming
Fabrizio Romano, Benjamin Baka and Dusty Phillips
Packt

Python is one of the most popular programming languages in the world, and for good reason. It's powerful, versatile, and easy to learn. If you're a beginner who's interested in learning Python, this guide is for you.

In this guide, we'll cover everything you need to know to get started with Python, from installation to data science applications. We'll start with the basics, like variables, data types, and operators. Then, we'll move on to more advanced topics, like functions, loops, and classes.

By the end of this guide, you'll be able to write your own Python programs and use them to solve real-world problems. So what are you waiting for? Let's get started!

Chapter 1: Installation

The first step to getting started with Python is to install it on your computer. Python is available for all major operating systems, including Windows, Mac, and Linux.

To install Python, visit the official Python website and download the latest version. Once the download is complete, run the installer and follow the on-screen instructions.

Once Python is installed, you can open a Python shell by typing "python" into your terminal or command prompt.

Chapter 2: Variables and Data Types

Variables are used to store data in Python. You can think of them as named boxes that can hold different types of data, such as numbers, strings, and lists.

To create a variable, simply assign it a value. For example:

```
python >>> my_name ="John Doe" >>> my_age = 30 >>> my_list = [1, 2, 3]
```

The data type of a variable is determined by the type of data it holds. Python has several built-in data types, including:

- Integers
- Floats
- Strings
- Lists
- Tuples
- Dictionaries

You can use the `type()` function to check the data type of a variable.

Chapter 3: Operators

Operators are used to perform operations on variables and values. Python has a variety of operators, including:

- Arithmetic operators (+, -, *, /, %)
- Comparison operators (==, !=, , =)
- Logical operators (and, or, not)
- Assignment operators (=, +=, -=, *=, /=, %=)
- You can use operators to perform a variety of tasks, such as:
  - Adding two numbers
  - Comparing two strings
  - Creating a new list

- Assigning a value to a variable
- Calculating the average of a list of numbers

Chapter 4: Functions

Functions are used to group code together and perform specific tasks. You can think of them as reusable blocks of code that you can call from anywhere in your program.

To create a function, use the `def` keyword. For example:

```
def my_function(): print("Hello, world!")
```

Once you've created a function, you can call it by using its name. For example:

```
my_function()
```

You can also pass arguments to functions. Arguments are values that are passed into a function when it is called. For example:

```
def my_function(name): print("Hello, " + name + "!") my_function("J
```

Chapter 5: Loops

Loops are used to iterate over sequences of data. Python has two types of loops: `for` loops and `while` loops.

`For` loops are used to iterate over a sequence of values. For example:

```
for i in range(10): print(i)
```

`While` loops are used to iterate over a sequence of values until a condition is met. For example:

```
while i Chapter 6: Classes <p>Classes are used to create new data ty
```

python class Person: def __init__(self, name, age): self.name = name self.age = age

Once you've created a class, you can create objects from it. For example:

python person = Person("John", 30)

Objects have their own set of attributes and methods. You can access the attributes of an object using the dot operator. For example:

python person.name

You can call the methods of an object using the dot operator followed by the method name. For example:

python person.get_age()

Chapter 7: Data Science Applications

Python is a powerful language for data science. It has a number of libraries that are specifically designed for data analysis and machine learning.

Some of the most popular Python libraries for data science include:

- NumPy
- Pandas
- Scikit-learn
- TensorFlow
- Keras

You can use these libraries to perform a variety of data science tasks, such as:

- Loading and cleaning data
- Exploratory data analysis
- Machine learning
- Deep learning

Python is a versatile and powerful language that is perfect for beginners. It is easy to learn and use, and it can be used for a wide variety of tasks, from web development to data science.

If you are interested in learning Python, I encourage you to check out the resources on the official Python website. There are also a number of online courses and tutorials that can help you get started.

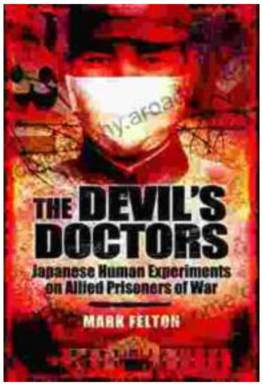
Once you have a basic understanding of Python, you can start using it to solve real-world problems. The possibilities are endless!

Getting Started with Python: Understand key data structures and use Python in object-oriented programming

★★★★★ 5 out of 5

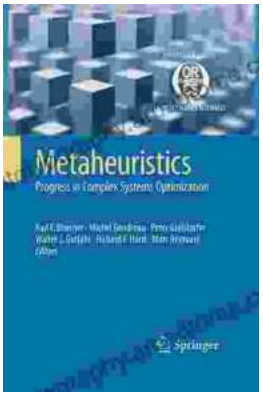
Language	: English
File size	: 12222 KB
Text-to-Speech	: Enabled
Screen Reader	: Supported
Enhanced typesetting	: Enabled
Print length	: 724 pages

FREE DOWNLOAD E-BOOK PDF



The Devil Doctors: A Heart-wrenching Tale of Betrayal and Resilience

The Devil Doctors is a gripping novel that explores the dark side of the medical profession. It follows the story of a young doctor who...



Progress In Complex Systems Optimization Operations Research Computer Science

This book presents recent research on complex systems optimization, operations research, and computer science. Complex systems are systems that...