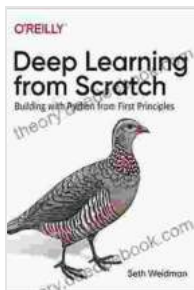


Building with Python From First Principles: A Comprehensive Guide for Beginners

Python is a powerful, versatile programming language that has become increasingly popular for a wide range of applications, from web development and data science to machine learning and artificial intelligence. Despite its versatility, Python is also known for its simplicity and ease of learning, making it an ideal choice for beginners who are just starting out in the world of programming. In this comprehensive guide, we will explore the basics of building with Python from first principles, providing step-by-step instructions and practical examples to help you get started and build your own Python projects.

Installing Python

Before you can start building with Python, you need to install it on your computer. Python is available for all major operating systems, including Windows, macOS, and Linux. You can download the latest version of Python from the official Python website (<https://www.python.org/downloads/>).



Deep Learning from Scratch: Building with Python from First Principles

by Seth Weidman

★★★★☆ 4.3 out of 5

Language : English
File size : 11637 KB
Text-to-Speech : Enabled
Screen Reader : Supported
Enhanced typesetting : Enabled
Print length : 253 pages



Once you have downloaded the Python installer, follow the instructions to install Python on your computer. Once the installation is complete, you can open a terminal or command window and type `python` to start the Python interpreter.

Writing Your First Python Program

Now that you have Python installed, you can start writing your first Python program. A Python program is simply a text file that contains Python code. You can use any text editor to write a Python program, but there are also several specialized Python editors available, such as PyCharm and IDLE.

To write your first Python program, open a text editor and type the following code:

```
python print("Hello, world!")
```

This program simply prints the message "Hello, world!" to the console. To run the program, save the file with a `.py` extension (e.g., `hello_world.py`) and then type the following command in a terminal or command window:

```
python hello_world.py
```

You should see the following output:

```
Hello, world!
```

Variables and Data Types

Variables are used to store data in Python. You can create a variable by assigning it a value, such as:

```
python name ="John Doe" age = 30
```

In Python, variables do not have a specific data type. The data type of a variable is determined by the value that is assigned to it. In the above example, the variable `name` is assigned a string value, so it is a string variable. The variable `age` is assigned an integer value, so it is an integer variable.

Python supports a variety of data types, including:

- Integers (e.g., 1, 2, 3)
- Floats (e.g., 1.23, 4.56, 7.89)
- Strings (e.g., "hello", "world", "Python")
- Lists (e.g., [1, 2, 3], ["hello", "world", "Python"], [1.23, 4.56, 7.89])
- Tuples (e.g., (1, 2, 3), ("hello", "world", "Python"), (1.23, 4.56, 7.89))
- Dictionaries (e.g., {"name": "John Doe", "age": 30}, {"hello": "world", "Python": "programming language"}, {1: 2, 3: 4, 5: 6})

Operators

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

- Arithmetic operators (e.g., +, -, *, /, %)

- Comparison operators (e.g., ==, !=, >, =,
- Logical operators (e.g., and, or, not)
- Assignment operators (e.g., =, +=, -=, *=, /=, %=)

For example, the following code uses the addition operator to add two numbers:

```
python num1 = 10 num2 = 20 result = num1 + num2 print(result)
```

This code will output the following:

30

Flow Control

Flow control statements are used to control the flow of execution in a Python program. Python supports a variety of flow control statements, including:

- If statements
- Elif statements
- Else statements
- For loops
- While loops
- Break statements
- Continue statements

For example, the following code uses an if statement to check if a number is greater than 0:

```
python num = 10 if num > 0: print("The number is greater than 0.")
```

This code will output the following:

The number is greater than 0.

Functions

Functions are used to group code together and perform specific tasks. You can define a function by using the `def` keyword, followed by the function name and parentheses. For example, the following code defines a function called `greet` that takes a name as an argument and prints a greeting:

```
python def greet(name): print("Hello, " + name + "!")
```

You can call a function by using its name followed by parentheses. For example, the following code calls the `greet` function and passes the name "John Doe" as an argument:

```
python greet("John Doe")
```

This code will output the following:

Hello, John Doe!

Classes and Objects

Classes and objects are used to create and organize data in Python. A class is a blueprint for creating objects. An object is an instance of a class.

To define a class, you use the **class** keyword, followed by the class name and a colon. For example, the following code defines a class called **Person** :

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

    def greet(self):
        print("Hello, my name is " + self.name + " and I am " +
              str(self.age) + " years old.")
```

To create an object, you use the **class** name followed by parentheses. For example, the following code creates an object of the **Person** class:

```
python person = Person("John Doe", 30)
```

You can access the attributes of an object using the dot operator. For example, the following code accesses the **name** attribute of the **person** object:

```
python print(person.name)
```

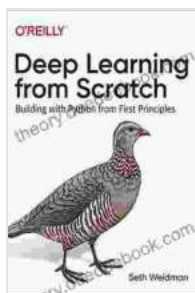
This code will output the following:

John Doe

This guide has provided a comprehensive overview of the basics of building with Python from first principles. By following the instructions and examples in this guide, you can start writing your own Python programs and building your own Python projects. As you gain more experience, you

can explore more advanced topics, such as object-oriented programming, data structures, and algorithms.

Python is a powerful and versatile programming language that can be used to build a wide range of applications. With its simplicity and ease of learning, Python is an ideal choice for beginners who are just starting out in the world of programming.

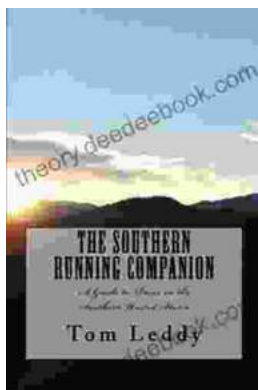


Deep Learning from Scratch: Building with Python from First Principles

by Seth Weidman

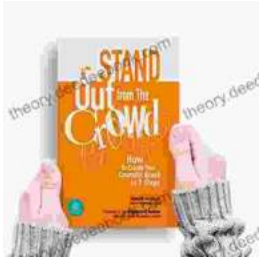
★★★★☆ 4.3 out of 5

Language : English
File size : 11637 KB
Text-to-Speech : Enabled
Screen Reader : Supported
Enhanced typesetting : Enabled
Print length : 253 pages



An Extensive Guide to Road Races in the Southern United States: Discover the Scenic Routes, Elevation Challenges, and Post-Race Festivities

Welcome to the vibrant world of Southern road racing! The Southern United States is a treasure trove of captivating races that offer a unique blend...



How to Create Your Cosmetic Brand in 7 Steps: A Comprehensive Guide

The cosmetic industry is booming, with an estimated global market size of over \$532 billion. If you're passionate about beauty and have a knack for entrepreneurship,...