

Mastering Python: From Basics to Advanced Projects



runner-code.com



Presented By Runner Code



Table of Contents

Introduction to Python
 Variables, Data Types, and Operators
 Control Statements
 Functions and Recursion
 Lists, Tuples, and Dictionaries
 Object-Oriented Programming (OOP)
 File Handling
 Libraries and Modules
 Advanced Concepts
 Advanced Projects





Defines the title in the browser bar
Derive a name for the page when bookmarked
Displays the name of the page when bookmarked





Python is a high-level, interpreted, and general-purpose programming language that emphasizes readability, simplicity, and versatility. Created by Guido van Rossum and first released in 1991, Python has become one of the most popular and widely used languages in the world due to its easy-to-understand syntax and vast ecosystem.

Introduction LO RATON

Key Features of Python: 1. Readable and Clean Syntax: semicolons. 2. Interpreted Language: 3. Dynamically Typed: development. 5. Extensive Standard Library: 6. Cross-Platform: 8.Community Support: problems.

• Python's syntax closely resembles human language, which makes it easier for developers to read and write code. It emphasizes indentation, reducing the need for braces or • Python is an interpreted language, meaning code is executed line by line, which simplifies debugging and testing. • Python does not require explicit variable declarations, and types are inferred at runtime, which leads to faster 4. Object-Oriented and Functional Programming: • Python supports multiple programming paradigms, including object-oriented, imperative, and functional programming, allowing flexibility in how you structure your code. • Python comes with a vast standard library for tasks like file handling, data manipulation, web development, and scientific computation. • Python can run on virtually any platform, including Windows, macOS, Linux, and others, making it highly portable. 7. Third-Party Libraries and Frameworks: • Python has an extensive ecosystem of third-party libraries and frameworks for various applications, such as Django and Flask for web development, TensorFlow and PyTorch for machine learning, and Pandas and NumPy for data analysis.

• Python has a vast and active global community, making it easy to find resources, tutorials, and solutions to

Nny Leafn Python?

- Data Science & Machine Learning: Python is the go-to language for data scientists and machine learning engineers due to powerful libraries like Pandas, NumPy, Matplotlib, and scikitlearn.
- Web Development: Python is widely used for developing websites and web applications, thanks to frameworks like Django and Flask.
- Automation & Scripting: Python is excellent for automating repetitive tasks and building small scripts for system administration.
- Scientific Computing: With libraries like SciPy, Matplotlib, and SymPy, Python is heavily used in scientific and research-based applications.
- Versatility: Python's simplicity and power allow it to be used in diverse fields such as game development, network programming, desktop applications, and more.





Python is known for its ability to handle a wide range of applications, from small scripts to large-scale systems, and is an excellent choice for beginners while also being powerful enough for professional software development. Whether you're into web development, automation, or data science, Python's simplicity and rich ecosystem make it a top choice for developers.

Setting Up the Development Environment



 Download and install Python from the <u>official website</u>.
 Install an IDE or text editor like PyCharm, VS Code, or Jupyter Notebook.
 Write and run your first program.









Your First Python Program Code Example:



print("Welcome to Python Programming!")

Explanation:
• print():

• print(): Outputs text to the console.



Chapter 2: Variables, Data Types, and Operators

Variables and Data Types • Data Types: int, float, str, bool, list, tuple, dict.

• Example:

age = 25height = 5.9grade = 'A' is student = True name = "John"

Input and Output Operations Code Example:

usage = int(input("Enter your age: ")) print(f"You entered: {age}")



Operators



- Arithmetic Operators: +, -, *, /, %
- Relational Operators: ==, !=, <, >, <=, >=
- Logical Operators: and, or, not



a = 10 b = 20print(a + b) # Outputs 30





print(a > b) # Outputs False

Chapter 3: Control Statements





if age >= 18: print("You are an adult.") else: print("You are a minor.")









Elif Example:

```
grade = 'A'
if grade == 'A':
    print("Excellent!")
elif grade == 'B':
    print("Good!")
else:
    print("Try harder!")
```











For Loop Example:



for i in range(5):
 print(i)





While Loop Example:

i = 0
while i < 5:
 print(i)
 i += 1</pre>



Chapter 4: Functions and Recutsion





Example:

def add(a, b): return a + b





Defining Functions

- print(add(10, 20)) # Outputs 30





Example:

Recursive Functions

def factorial(n): if n == 0: return 1 return n * factorial(n - 1)

\$(function(){cards();});
\$(function(){cards
\$(window).on('resize', function(){cards
\$(window).width();
function cards(){
function cards(){
function cards(){
function cards(){
function cards(){
function cards();
function car }else{ cardsbigscreen();

42

43

44 7

45 46 1

tolower(

define('PSI_INTERNAL_XML', false); f (version_compare("5.2", PHP_VERSION, ">")) {
 die("PHP 5.2 or greater is required!!!"); (!extension_loaded("pcre")) {
 die("phpSysInfo requires the pcre extension to php in order to work
 properly."); require_once APP_ROOT.'/includes/autoloader.inc.php';

Chapter 5: Lists, Tuples, and Dictionaries





Lists Example:

arr = [1, 2, 3, 4, 5]
for num in arr:
 print(num)









TUDIES Example:



tup = (1, 2, 3)
print(tup[0])







Dictionaries Example:

dict = {"name": "John", "age": 25}
print(dict["name"])







Chapter 6: Object-Offented Programming (OOP)





class Car:

my_car.honk()

Classes and Objects Example:

```
def __init__(self, brand):
    self.brand = brand
def honk(self):
    print("Beep! Beep!")
```

```
my_car = Car("Toyota")
```



Chapter 7: File Handling

File Operations Example:



with open("example.txt", "w") as file:
 file.write("Hello, File!")

with open("example.txt", "r") as file: content = file.read() print(content)







Chapter 8: Libraries and NOCULES





Using Libraries Example:

import math

print(math.sqrt(16))









Chapter 9: Advanced Concepts





Asynchronous Programming **Example:**

import asyncio

async def fetch_data(): await asyncio.sleep(1) print("Data fetched.")

asyncio.run(fetch_data())









```
class Book:
         def __init__(self, id, title, author):
             self.id = id
             self.title = title
 4
             self.author = author
 5
         def str (self):
             return f"ID: {self.id}, Title: {self.title}, Author: {self.author}"
 8
     class Library:
10
         def __init__(self):
11
             self.books = []
12
13
         def add book(self, book):
14
             self.books.append(book)
15
16
             print(f"Book '{book.title}' added successfully.")
17
         def remove_book(self, id):
18
             book_to_remove = next((book for book in self.books if book.id == id), None)
19
20
             if book to remove:
                 self.books.remove(book to remove)
21
                 print(f"Book '{book_to_remove.title}' removed successfully.")
22
23
             else:
                 print("Book not found.")
24
25
                                                             Code
         def display_books(self):
26
27
             if not self.books:
                 print("No books in the library.")
28
29
             else:
                 print("Library Catalog:")
30
31
                 for book in self.books:
32
                     print(book)
```



```
library = Library()
while True:
    print("\nLibrary Management System")
    print("1. Add Book")
    print("2. Remove Book")
    print("3. Display Books")
    print("4. Exit")
    choice = input("Choose an option: ")
    if choice == "1":
        id = int(input("Enter Book ID: "))
        title = input("Enter Book Title: ")
        author = input("Enter Book Author: ")
        library.add_book(Book(id, title, author))
    elif choice == "2":
        id = int(input("Enter Book ID to Remove: "))
        library.remove_book(id)
    elif choice == "3":
        library.display_books()
    elif choice == "4":
        print("Exiting the Library Management System.")
        break
    else:
        print("Invalid option. Please try again.")
```

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56











C:\WINDOWS\py.exe

Library Management System 1. Add Book 2. Remove Book 3. Display Books 4. Exit Choose an option: 1 Enter Book ID: 100 Enter Book Title: Introducation to Python Enter Book Author: Runner Code

×

 $+ \cdot$

Book 'Introducation to Python' added successfully.

Library Management System

- 1. Add Book
- 2. Remove Book
- 3. Display Books
- Exit

Choose an option:







Thank You

<u>infoctunnetfoccesco</u>

runner-code.com

