

Curso Gratuito  
Anúncio · xperium.com

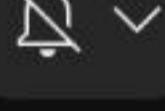
Saiba mais

## Como Mesclar PDFs com Python em Menos de 10 Linhas de Código

10 mil visualizações há 8 meses Python para o Mercado ...mais



Hashtag Programação 410 mil



544



Compartilhar

Remix

Val

### Comentários 17



Adicione um comentário...

Google Ads

Patrocinados

Sua Empresa

Próximo: Tratamento de Imagens com Python...  
Python • 1/59

A screenshot of a video player interface. At the top, there's a status bar with the time '21:50' and battery level '41%'. Below it is a dark-themed code editor window showing Python code for merging PDF files using the PyPDF2 library. The code imports PyPDF2, creates a PdfFileMerger object, appends a file named 'exemplo-de-pdf.pdf', and writes the result to 'PDF\_Final.pdf'. On the right side of the screen, there's a video frame showing a man with a beard and short hair, wearing a dark t-shirt, looking directly at the camera.



Curso Gratuito  
Anúncio · xperium.com

[Saiba mais](#)

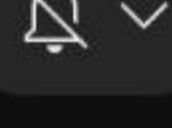


## Como Mesclar PDFs com Python em Menos de 10 Linhas de Código

10 mil visualizações há 8 meses Python para o Mercado ...mais



Hashtag Programação 410 mil



544

Compartilhar

Remix

Val

### Comentários 17



Adicione um comentário...

Google Ads

Patrocinados

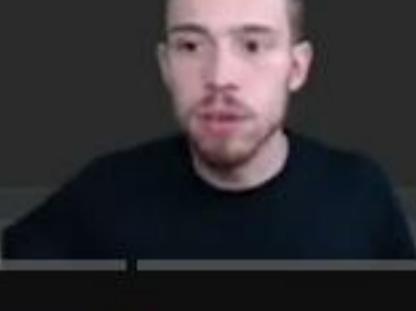
Sua Empresa

Próximo: Tratamento de Imagens com Python...  
Python • 1/59

```
import PyPDF2
import os

merger = PyPDF2.PdfFileMerger()

lista_arquivos = os.listdir("pdfs_nesca")
merger.write("PDF_Final.pdf")
```



Curso Gratuito  
Anúncio · xperium.com

[Saiba mais](#)

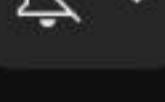


## Como Mesclar PDFs com Python em Menos de 10 Linhas de Código

10 mil visualizações há 8 meses Python para o Mercado ...mais



Hashtag Programação 410 mil



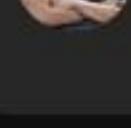
544

Compartilhar

Remix

Val

### Comentários 17



Adicione um comentário...

# Google Ads

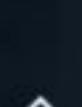
Patrocinados

Sua Empresa



Próximo: Tratamento de Imagens com Python...

Python • 1/59



21:53

40%

The screenshot shows a PyCharm interface with a code editor containing Python code for merging PDF files using the PyPDF2 library. The code imports PyPDF2 and os, initializes a PdfFileMerger object, lists all files in a directory named 'pdfs\_mesclar', appends them to the merger, and finally writes the merged file to 'PDF\_Final.pdf'. The terminal below shows the command run and its output.

```
import PyPDF2
import os

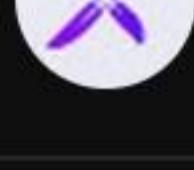
merger = PyPDF2.PdfFileMerger()

lista_arquivos = os.listdir("pdfs_mesclar")
for arquivo in lista_arquivos:
    merger.append(f"pdfs_mesclar/{arquivo}")

merger.write("PDF_Final.pdf")
```

```
C:\Users\joao1\PycharmProjects\GravacoesYoutube\venv\Scripts\python.exe C:\Users\joao1\PycharmProjects\Gravacoes
['Exemplo-de-PDF (1).pdf', 'exemplo-de-pdf.pdf', 'pdf_sample_2.pdf']

Process finished with exit code 0
```



Curso Gratuito  
Anúncio · xperium.com

Saiba mais



## Como Mesclar PDFs com Python em Menos de 10 Linhas de Código

10 mil visualizações há 8 meses Python para o Mercado ...mais



Hashtag Programação 410 mil



544

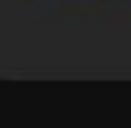


Compartilhar

Remix

Val

### Comentários 17



Adicione um comentário...

Google Ads

Patrocinados

Sua Empresa

Próximo: Tratamento de Imagens com Python...

Python • 1/59

21:54

40%

The screenshot shows a PyCharm interface with a code editor containing Python code for merging PDF files. The code uses the PyPDF2 library to merge multiple PDF files into one. The code editor has syntax highlighting and a status bar indicating the current file is 'merger.pdf'. Below the code editor is a terminal window showing the command run and its output, which includes the path to the merged PDF file.

```
import PyPDF2
import os

merger = PyPDF2.PdfFileMerger()

lista_arquivos = os.listdir("pdfs_mesclar")
for arquivo in lista_arquivos:
    if ".pdf" in arquivo:
        merger.append(f"pdfs_mesclar/{arquivo}")

merger.write("PDF_Final.pdf")
```

```
for arquivo in lista_arquivos: "pdf" in arquivo
C:\Users\joao1\PycharmProjects\GravacoesYoutube\venv\Scripts\python.exe C:\Users\joao1\PycharmProjects\Gravacoes
['Exemplo-de-PDF (1).pdf', 'exemplo-de-pdf.pdf', 'pdf_sample_2.pdf']

Process finished with exit code 0
```



Curso Gratuito  
Anúncio · xperium.com

Saiba mais



## Como Mesclar PDFs com Python em Menos de 10 Linhas de Código

10 mil visualizações há 8 meses Python para o Mercado ...mais



Hashtag Programação 410 mil



544

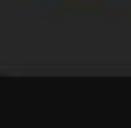


Compartilhar

Remix

Val

### Comentários 17



Adicione um comentário...

Google Ads

Patrocinados

Sua Empresa

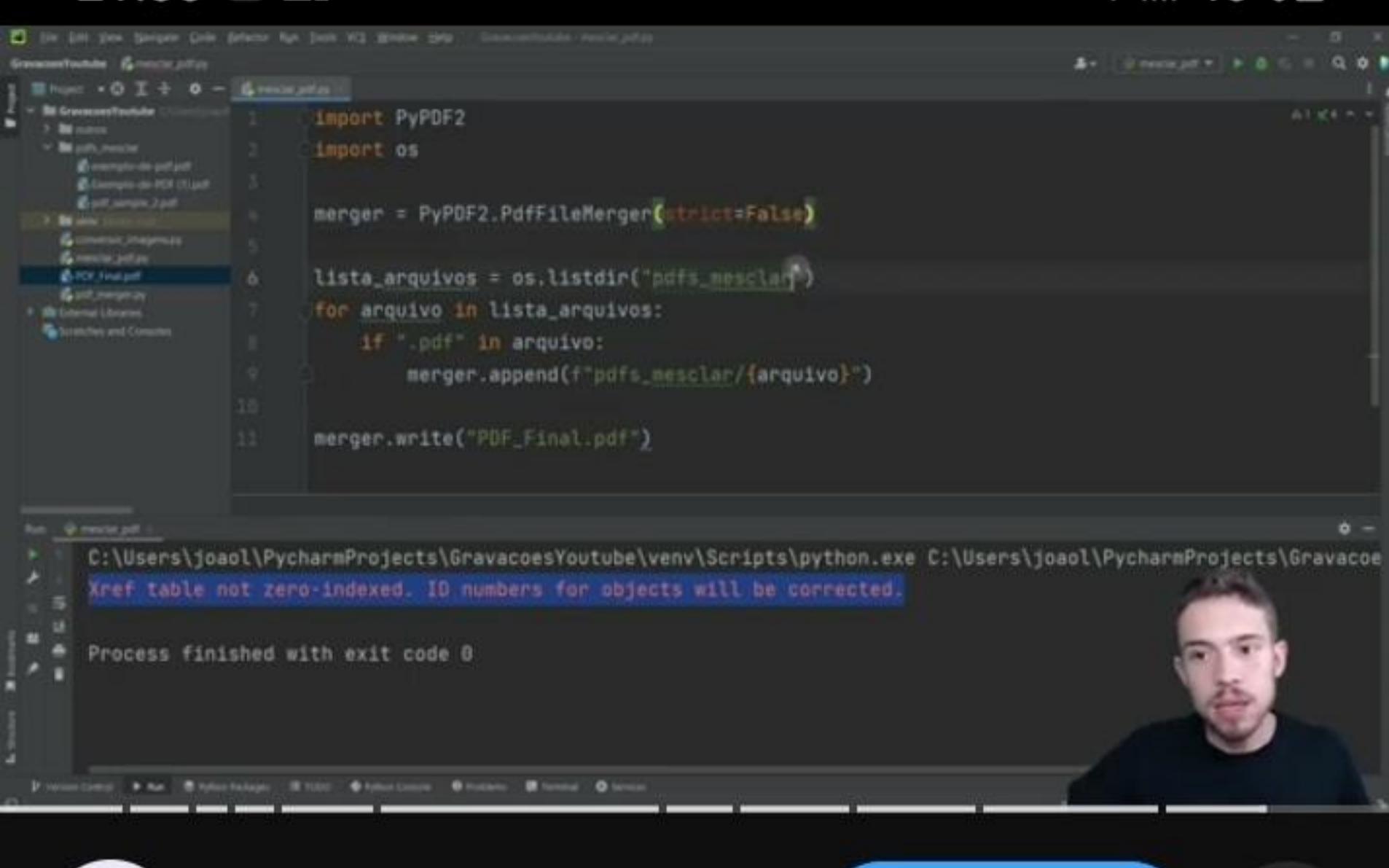
Próximo: Tratamento de Imagens com Python...

Python • 1/59



21:55

40%



PyCharm interface showing a Python script named `mesclar.py`. The code uses the `PyPDF2` library to merge multiple PDF files into one. The terminal output shows the command run and a warning about Xref table indexing.

```
import PyPDF2
import os

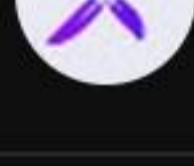
merger = PyPDF2.PdfFileMerger(strict=False)

lista_arquivos = os.listdir("pdfs_mesclar")
for arquivo in lista_arquivos:
    if ".pdf" in arquivo:
        merger.append(f"pdfs_mesclar/{arquivo}")

merger.write("PDF_Final.pdf")
```

```
C:\Users\joao1\PycharmProjects\GravacoesYoutube\venv\Scripts\python.exe C:\Users\joao1\PycharmProjects\GravacoesYoutube\mesclar.py
Xref table not zero-indexed. ID numbers for objects will be corrected.
```

Process finished with exit code 0



Curso Gratuito  
Anúncio · xperium.com

Saiba mais



## Como Mesclar PDFs com Python em Menos de 10 Linhas de Código

10 mil visualizações há 8 meses Python para o Mercado ...mais



Hashtag Programação 410 mil



544

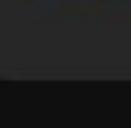


Compartilhar

Remix

Val

### Comentários 17



Adicione um comentário...

# Google Ads

Patrocinados

Sua Empresa

Próximo: Tratamento de Imagens com Python...

Python • 1/59



# Python



Code Editor

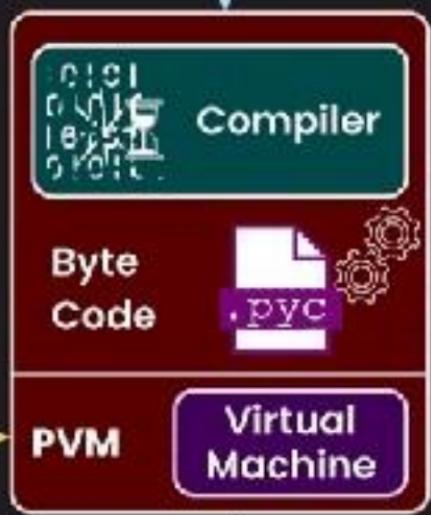


Source File

Python Interpreters



Libraries/  
Modules



Binary/Machine  
Code



Running Program



2



Other useful methods include the following:

- `contains()`: Returns true if the list contains the specified element
- `get(int index)`: Returns the element at the specified position in the list
- `size()`: Returns the number of elements in the list
- `clear()`: Removes all of the elements from the list



Note: As with arrays, the indexing starts with 0.

Code Coach

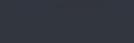
## ArrayList



Write a code to add input numbers to ArrayList while the size of ArrayList isn't equal to 3.

XP +5

PRACTICE



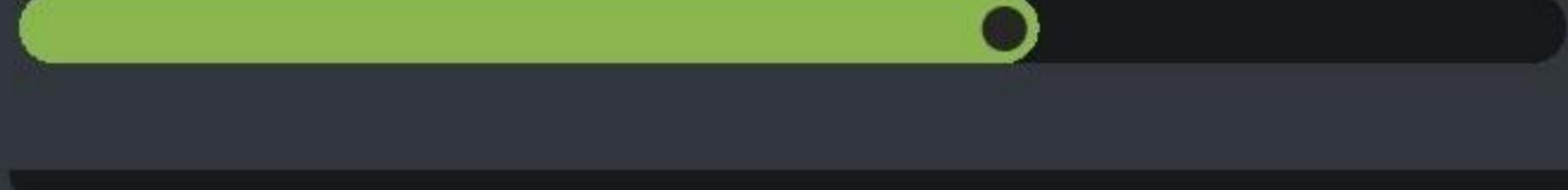
COMMENTS

CONTINUE





3



## Other useful methods in the `Collections` class:



`max(Collection c)`: Returns the maximum element in c as determined by natural ordering.

`min(Collection c)`: Returns the minimum element in c as determined by natural ordering.

`reverse(List list)`: Reverses the sequence in list.

`shuffle(List list)`: Shuffles (i.e., randomizes) the elements in list.

## Code Coach

### Sorting Lists



Complete the code to output ArrayList's maximum and minimum values.

XP +5

PRACTICE

—

COMMENTS

CONTINUE





3

## Reading a file

The `Scanner` class inherits from the `Iterator`, so it behaves like one.

We can use the `Scanner` object's `next()` method to read the file's contents.

JAVA

```
try {
    File x = new File("C:\\\\solearn\\\\test.txt");
    Scanner sc = new Scanner(x);
    while(sc.hasNext()) {
        System.out.println(sc.next());
    }
    sc.close();
} catch (FileNotFoundException e) {
    System.out.println("Error");
}
```

The file's contents are output word by word, because the `next()` method returns each word separately.

—  
10 COMMENTS

CONTINUE



3

```
try {  
  
    File f = new File("a.txt");  
  
    Scanner sc = new Scanner(f);  
  
    while (sc.hasNext()) {  
  
        String a = sc.next();  
  
        String b = sc.next();  
  
        System.out.println(a + " " + b);  
  
    }  
  
    sc.close();  
  
}  
  
catch (Exception e) {  
  
    System.out.println("Error");  
  
}
```



## Creating & Writing Files

:

PROBLEM

CODE

RESULT

```
1 import java.io.File;
2 import java.util.Scanner;
3 import java.util.Formatter;
4
5 public class Main {
6     public static void main(String[]
7 args) {
8         Scanner input = new
9 Scanner(System.in);
10        try {
11            Formatter f = new
12 Formatter("tasks.txt");
13            int count = 0;
14            while (count < 3) {
15                String task =
16                input.nextLine();
17                f.format("%s%n",
18 task);
19                count++;
20            }
21            f.close();
22
23            File x = new
24 File("tasks.txt");
25            Scanner sc = new
26 Scanner(x);
27            while (sc.hasNext()) {
28                System.out.println(sc.nextLine());
29            }
30            sc.close();
31        } catch (Exception e) {
32
33            System.out.println("Error");
34        }
35    }
36}
```



TAB

&lt;

&gt;

!

/

.

RUN





3

Rearrange the code to write "Hi there" to the file.

```
Formatter f = new  
Formatter("a.txt");
```

```
f.format("%s", "Hi ");
```

```
f.format("%s", "there");
```

```
f.close();
```

4 COMMENTS

Learn a little every day

CONTINUE



3

```
••  
def rect(d1, d2):  
    area = d1 * d2  
    perimeter = 2 * d1 + 2 * d2  
    return area, perimeter
```

```
x, y = rect(50, 100)
```

When a function returns multiple values, they can be stored in **multiple variables** (on 1 line).

Match the return value to its storing variable

x: area

y: perimeter

ANSWER

CHECK





2

The default value is used only if no other value has been passed as an argument when the function is called.

PY

```
def greet(name="Guest"):  
    print("Welcome", name)  
greet("Anna")
```

What will this code output?

Welcome Anna

Welcome Guest

18 COMMENTS

Give yourself a pat on the back!

CONTINUE



3

What is the output of this code?

```
def test(func, arg):  
    return func(func(arg))
```

```
def mult(x):  
    return x * x
```

```
print(test(mult, 2))
```

16

19 COMMENTS

You're getting good at this!

CONTINUE

## 19 COMMENTS

Most Popular ▾



Elvin Alshanov

8/17/2023



test(mult, 2) is called.  
func is mult and arg is 2, so func(arg) is mult(2)  
which returns  $2 * 2$  (i.e., 4).  
Now, func is still mult, and arg is 4, so func(arg) is  
mult(4) which returns  $4 * 4$  (i.e., 16).

^ +21

3 REPLIES REPLY



Zayne Pilgrim

8/18/2023



Actually this would be an error since there is no  
indent

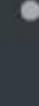
^ +13

2 REPLIES REPLY



Farhad Rahmani

8/8/2023



Why? can anyone help

^ +5

1 REPLY REPLY



Jonathan

8/11/2023



Questions too easy now

^ +4

0 REPLIES REPLY



Павел П

8/7/2023



Нет отступов. Должен выдать ошибку, но выдаст  
16

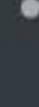
^ +4

1 REPLY REPLY



Joaquin Vaudeville

9/27/2023



Seems to be not well explained... :(

^ +2

0 REPLIES REPLY



Acer Quinn

8/24/2023



Bless that guy that explained in the previous  
comment section. Answering this was a breeze

^ +1

1 REPLY REPLY



## ← Generators

PROBLEM

CODE

RESULT

# Generators

Finding **prime numbers** is a common coding interview task.

The given code defines a function **isPrime(x)**, which returns True if x is prime.

You need to create a generator function **primeGenerator()**, that will take two numbers as arguments, and use the **isPrime()** function to output the prime numbers in the given range (between the two arguments).

### Sample Input

10

20

### Sample Output

[11, 13, 17, 19]



The given code takes the two arguments as input and passes them to the generator function, outputting the result as a list.

START SOLVING



## ← Recursion

[PROBLEM](#)[CODE](#)[RESULT](#)

# Recursion

The given code defines a recursive function **convert()**, which needs to convert its argument from decimal to binary.

However, the code has an error.

Fix the code by adding the **base case** for the recursion, then take a number from user input and call the `convert()` function, to output the result.

### Sample Input

8

### Sample Output

1000



The binary representation of 8 is 1000.

[START SOLVING](#)



1

What is the output of this code?

```
def func(**kwargs):  
    print(kwargs["zero"])
```

```
func(a = 0, zero = 8)
```

ANSWER

Type

ANSWER

CHECK





1

Which keyword is used to return a value from a generator?

Type

ANSWER

CHECK





1

What is `A() ^ B()` evaluated as, if A doesn't implement any magic methods?

`A().__xor__(B())`

`B().__xor__(A())`

`B().__rxor__(A())`

ANSWER

CHECK





# Magic Methods

Python also provides magic methods for comparisons.

**lt** for <

**le** for <=

**eq** for ==

**ne** for !=

**gt** for >

**ge** for >=

If **ne** is not implemented, it returns the opposite of **eq**.

There are no other relationships between the other operators.

**Example:**

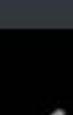
CODE PLAYGROUND

PY

```
class SpecialString:  
    def __init__(self, cont):  
        self.cont = cont
```

—  
6 COMMENTS

CONTINUE





1

# Magic Methods

There are several magic methods for making classes act like containers.

**len** for `len()`

**getitem** for indexing

**setitem** for assigning to indexed values

**delitem** for deleting indexed values

**iter** for iteration over objects (e.g., in for loops)

**contains** for in

There are many other magic methods that we won't cover here, such as **call** for calling objects as functions, and **int**, **str**, and the like, for converting objects to built-in types.

Example:

CODE PLAYGROUND

PY

```
import random
```

```
class VagueList:
```

6 COMMENTS

CONTINUE





1

```
class VagueList:  
    def __init__(self, cont):  
        self.cont = cont  
  
    def __getitem__(self, index):  
        return self.cont[index +  
random.randint(-1, 1)]  
  
    def __len__(self):  
        return random.randint(0,  
len(self.cont)*2)  
  
vague_list = VagueList(["A", "B", "C",  
"D", "E"])  
print(len(vague_list))  
print(len(vague_list))  
print(vague_list[2])  
print(vague_list[2])
```

Tap to try |



We have overridden the `len()` function for the class `VagueList` to return a random number. The indexing function also returns a random item in a range from the list, based on the expression.

—

**6 COMMENTS**

**CONTINUE**



## ← Operator Overloading

PROBLEM

CODE

RESULT

# Operator Overloading

We are improving our drawing application. Our application needs to support adding and comparing two **Shape** objects.

Add the corresponding methods to enable addition + and comparison using the greater than > operator for the Shape class.

The addition should return a new object with the sum of the widths and heights of the operands, while the comparison should return the result of comparing the areas of the objects.



The given code creates two **Shape** objects from user input, outputs the area() of their addition and compares them.

START SOLVING





2

How would the attribute `_a` of the class `b` be accessed from outside the class?

Type

ANSWER

CHECK





2

How would the attribute `_a` of the class `b` be accessed from outside the class?

`_b_a`

4 COMMENTS



3 in a row!

CONTINUE



2

## CODE PLAYGROUND

PY

```
class Pizza:  
    def __init__(self, toppings):  
        self.toppings = toppings  
  
    @staticmethod  
    def validate_topping(topping):  
        if topping == "pineapple":  
            raise ValueError("No  
pineapples!")  
        else:  
            return True  
  
ingredients = ["cheese", "onions",  
"spam"]  
if all(Pizza.validate_topping(i) for i  
in ingredients):  
    pizza = Pizza(ingredients)
```

Tap to try |



Static methods behave like plain functions, except for the fact that you can call them from an instance of the class.

—

14 COMMENTS

CONTINUE





3

Fill in the blanks to make the egg attribute strongly private and access it from outside of the class.

```
class Test:
```

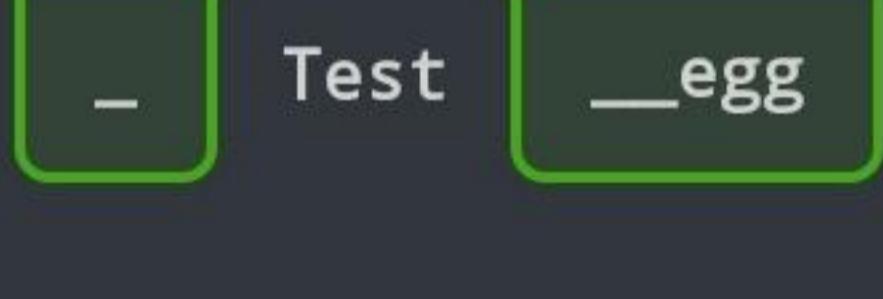


```
    egg = 7
```

```
t = Test()
```



```
print(t.
```



```
    __egg
```

13 COMMENTS



5 in a row!

CONTINUE





3

Fill in the blanks to make a setter for the property name.

```
class Person:  
  
    def __init__(self, name):  
  
        self._name = name  
  
    @property  
  
    def name(self):  
  
        return self._name
```

@name.setter

```
def name(self, value):
```

```
    self._name = value
```

3 COMMENTS

Keep on learning!

CONTINUE



## ← Exception Handling

PROBLEM

CODE

RESULT

# Exception Handling

An ATM machine takes the amount to be withdrawn as input and calls the corresponding withdrawal method.

In case the input is not a number, the machine should output "Please enter a number".

Use exception handling to take a number as input, call the `withdraw()` method with the input as its argument, and output "Please enter a number", in case the input is not a number.



A `ValueError` is raised when you try to convert a non-integer to an integer using `int()`.

START SOLVING

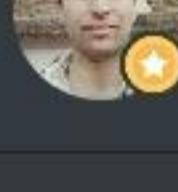
## Q&A Discussions



try:

```
#your code goes here  
if len(tweet) >= 42:  
    raise ValueError  
except:  
    print("Error")  
else:  
    print("Posted")
```

Nazarii Zavada



1 year

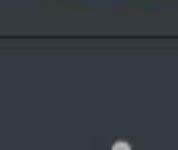
### My solution:



0 tweet = input()

```
try:  
    #your code goes here  
    if len(tweet) > 42:  
        raise Exception()  
  
except:  
    print("Error")  
else:  
    print("Posted")
```

Pooja Patel



11 months

0 tweet = input()



```
try:  
    #your code goes here  
    if len(tweet)>42:  
        raise Exception()  
except:  
    print("Error")  
else:  
    print("Posted")
```

المهندس أحمد



6 months



Write your post here





Fill in the blanks to output "Wrong Input" if the input is not an integer.

```
num = input()
```

```
try :
```

```
    x = int(num)
```

```
except :
```

```
    print ("Wrong Input")
```

2 COMMENTS

What could you do with this skill?

CONTINUE

## 8 COMMENTS

Most Popular ▾



Георгий Ким

8/13/2023



1.  $x = 0$
2. Program try to  $0+1$
3.  $x = 1$
4. raise get ValueError
5. Except:  $1+1$
6.  $x = 2$
7. finally: program  $2+1$
8.  $x = 3$

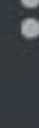
^ +8

0 REPLIES REPLY



Илья Мандрыкин

8/3/2023



Why 3?

^ +6

3 REPLIES REPLY



TeaWhyEleeeAre

10/16/2023



Start with Zero:  $x = 0$  means we start with a value of 0 for  $x$ .

Try to Add 1:  $x += 1$  means add 1 to the current value of  $x$ . So,  $x$  becomes 1.

Raise an Error: raise ValueError is like saying, "Wait, something went wrong!" It's intentionally causing an error.

Handle the Error: Since there's an error (a ValueError), we jump to the except ValueError: block. We add 1 to  $x$  again. So,  $x$  becomes 2.

Finally Block: The finally: block always runs, no matter what. We add 1 to  $x$  in the finally block. So,  $x$  becomes 3.

So, the final value of  $x$  is 3 after running this code. We added 1, encountered an error, handled the error, and then added 1 more in the finally block.

^ +1

0 REPLIES REPLY



VoltKid

10/16/2023

 $x = 0$ 

## 8 COMMENTS

Most Popular ▾

Raise an Error: raise ValueError is like saying, "Wait, something went wrong!" It's intentionally causing an error.

Handle the Error: Since there's an error (a ValueError), we jump to the except ValueError: block. We add 1 to x again. So, x becomes 2.

Finally Block: The finally: block always runs, no matter what. We add 1 to x in the finally block. So, x becomes 3.

So, the final value of x is 3 after running this code. We added 1, encountered an error, handled the error, and then added 1 more in the finally block.

▲ +1

0 REPLIES REPLY

VoltKid

10/18/2023

⋮

x = 0

Try

x += 1 -&gt; 1

raise ValueError

Except ValueError

x += 1 -&gt; 2

finally

x += 1 -&gt; 3

▲ 0

0 REPLIES REPLY

⋮

Trey

10/7/2023

Could this lead to a massive chain reaction? Post:  
<https://sololearn.com/discuss/3246222/?ref=app>



Multi Exception

Trey

▲ 0

0 REPLIES REPLY

⋮





## Opening a file

You can use Python to read and write the contents of **files**.

This is particularly useful when you need to work with a lot of data that is saved in a file.

For example, in data science and analytics, the data is commonly in **CSV** (comma-separated values) files.

Let's start by working with text files, as they are the easiest to manipulate.

Before a file can be edited, it must be opened, using the **open** function.

PY

```
myfile = open("filename.txt")
```



The argument of the **open** function is the **path** to the file. If the file is in the current working directory of the program, you can specify only its name.



6 COMMENTS

CONTINUE





You can specify the **mode** used to open a file by applying a second argument to the **open** function. Sending "r" means open in read mode, which is the default.

Sending "w" means write mode, for rewriting the contents of a file.

Sending "a" means append mode, for adding new content to the end of the file.

Adding "b" to a mode opens it in **binary mode**, which is used for non-text files (such as image and sound files).

For example:

```
PY  
# write mode  
open("filename.txt", "w")  
  
# read mode  
open("filename.txt", "r")  
open("filename.txt")  
  
# binary write mode  
open("filename.txt", "wb")
```

—

2 COMMENTS

CONTINUE





Drag and drop from the options below to open a file called "test.bin" in binary read mode.

file = open( "test.bin" , "rb" )

"r"

"b"

"test.txt"

"w"

9 COMMENTS

You'll reach your goals in no time

CONTINUE



## Lesson Completed!

You learned Opening Files. You're one step closer to reaching your goal!

**Your reward:**

+10 XP

+5

**CONTINUE**



Rearrange the code to open a file, read its contents, print them, and then close the file.

```
file = open("test.txt")
```

```
cont = file.read()
```

```
print(cont)
```

```
file.close()
```

2 COMMENTS

You're a champion

CONTINUE

## ← Python Developer



```
1 file = open("/usercode/files/books.txt")
2 print(file.read(5))
3 print(file.read(7))
4 print(file.read())
5 file.close()
```



### OUTPUT

Harry  
Potter

The Hunger Games  
Pride and Prejudice  
Gone with the Wind



## 10 COMMENTS

Most Popular ▾

too... Although the answer is 2, it should ignore the char/byte assumption and the answer should be 4 characters. This needs to be fixed by solelearn!

▲ +10

1 REPLY REPLY



Dane Krizmanic

8/9/2023

⋮

Solution = 2

▲ +3

1 REPLY REPLY



Acer Quinn

9/7/2023

⋮

I did not understand. I actually answered 12.

▲ +1

0 REPLIES REPLY



Hezkiel Bram Setiawan

8/10/2023

⋮

characters each line, not total characters

▲ +1

1 REPLY REPLY



Christian Barber

10/16/2023

⋮

I answered 42 bc I thought  
 $(4/2) \times 21 = 42$ 

▲ 0

0 REPLIES REPLY



Alex V Sagalovych

10/16/2023

⋮

2

▲ 0

0 REPLIES REPLY



Elham Khanlari

10/13/2023

⋮

(4) bytes =&gt; 2 characters

▲ 0

0 REPLIES REPLY



## 10 COMMENTS

Most Popular ▾

T

Walid

9/8/2023

⋮

the question is wrong because to start with , firstly a character is 1 byte only and not 2 bytes, secondly the file is opened as ASCLi and not binary to read bytes. it should read characters. Therefore, the assumption is false and also the reading mechanism is false too... Although the answer is 2, it should ignore the char/byte assumption and the answer should be 4 characters. This needs to be fixed by solelearn!

^ +10

1 REPLY REPLY



Dane Krizmanić

8/9/2023

⋮

Solution = 2

^ +3

1 REPLY REPLY



Acer Quinn

9/7/2023

⋮

I did not understand. I actually answered 12.

^ +1

0 REPLIES REPLY



Hezkiel Bram Setiawan

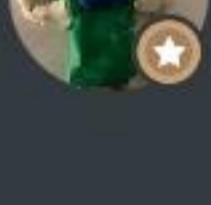
8/10/2023

⋮

characters each line, not total characters

^ +1

1 REPLY REPLY



Christian Barber

10/16/2023

⋮

I answered 42 bc I thought  
 $(4/2) \times 21 = 42$ 

^ 0

0 REPLIES REPLY



Alex V Sagalovych

10/16/2023

⋮

2

^ 0

0 REPLIES REPLY



Elham Khanlari

10/17/2023

⋮

(4) bytes =&gt; 2 characters

^ 0

0 REPLIES REPLY

+



Fill in the blanks to open a file, read its content and print its 2nd line.

```
file = open("filename.txt", "r")
```

```
cont = file.readlines()
```

```
print(cont[1])
```

```
file.close()
```

4 COMMENTS

You make this look easy!

CONTINUE

# ← Reading Files



PROBLEM

CODE

RESULT

```
1 file = open("/usercode/files/  
books.txt")  
2 #your code goes here  
3  
4 n= int(input())  
5
```



TAB

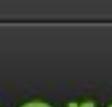
( )

"

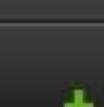
'

{ }

RUN ▶



あ



...



perc

active\_minutes

t

1 2 3 4 5 6 7 8 9 0

% ^ ~ | [ ] &lt; &gt; { }

q w e r t y u i o p

@ # &amp; \* - + = ( )

a s d f g h j k l

↑ - \$ " ' : ; / m ×

z x c v b n ← , !?

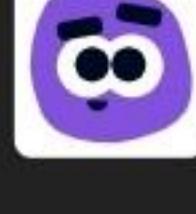
123 ☺ en/es/pt @ . ←



## ← Reading Files

**PROBLEM****CODE****RESULT**

```
1 file = open("/usercode/files/  
books.txt")  
2 #your code goes here  
3  
4 n= int(input())  
5  
6 print(file.read(n))  
7  
8 file.close()  
9
```



TAB

( )

"

'

{

RUN





To write to files you use the **write** method.

For example:

### CODE PLAYGROUND

PY

```
file = open("newfile.txt", "w")
file.write("This has been written to a
file")
file.close()
```

Tap to try |



In case the file already exists, its entire content will be replaced when you open it in write mode using "w".

—  
3 COMMENTS

CONTINUE





Which line of code writes "Hello world!" to a file?

`file.write("Hello world!")`

`write(file, "Hello world!")`

`write("Hello world!", file)`

1 COMMENT

What could you do with this skill?

CONTINUE



# Writing Files

If you want to add content to an existing file, you can open it using the "a" mode, which stand for "append":

## CODE PLAYGROUND

PY

```
file = open("/usercode/files/  
books.txt", "a")
```

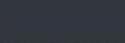
```
file.write("\nThe Da Vinci Code")  
file.close()
```

Tap to try |

This will add a new line with a new book title to the file.



Remember \n stands for a new line



4 COMMENTS

CONTINUE





# Writing Files

If you want to add content to an existing file, you can open it using the "a" mode, which stand for "append":

## CODE PLAYGROUND

PY

```
file = open("/usercode/files/  
books.txt", "a")
```

```
file.write("\nThe Da Vinci Code")  
file.close()
```

Tap to try |

This will add a new line with a new book title to the file.



Remember \n stands for a new line



4 COMMENTS

CONTINUE





If you want to add content to an existing file, you can open it using the "a" mode, which stand for "append":

### CODE PLAYGROUND

PY

```
file = open("/usercode/files/  
books.txt", "a")  
  
file.write("\nThe Da Vinci Code")  
file.close()
```

Tap to try |



Remember, \n stands for a new line.

4 COMMENTS

CONTINUE



19:44



VoLTE 16%



Fill in the blank to open the file in append mode.

a

9 COMMENTS

A little learning goes a long way

CONTINUE



# Writing Files

The **write** method returns the number of **bytes** written to a file, if successful.

## CODE PLAYGROUND

PY

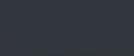
```
msg = "Hello world!"  
file = open("newfile.txt", "w")  
amount_written = file.write(msg)  
print(amount_written)  
file.close()
```

Tap to try |

The code above will write to the file and print the number of bytes written.



To write something other than a string, it needs to be converted to a string first.



6 COMMENTS

CONTINUE





If a file write operation is successful, which one of these statements will be true?

`file.write(msg) == True`

`file.write(msg) == msg`

`file.write(msg) == len(msg)`

1 COMMENT



3 in a row!

CONTINUE

## Q&A Discussions



You are provided a books.txt file, which includes the book titles, each one written on a separate line.

Read the title one by one and output the code for each book on a separate line.

For example, if the books.txt file contains:

Some book  
Another book

Your program should output:

S9  
A12

Recall the `readlines()` method, which returns a list containing the lines of the file.

Also, remember that all lines, except the last one, contain a `\n` at the end, which should not be included in the character count.

### MY SOLUTION:

```
file = open("/usercode/files/books.txt", "r")
```

```
#your code goes here
fileLines = []
for lines in file:
    fileLines.append(lines)
length = 1
for n in fileLines:
    if length == len(fileLines):
        print(n[0] + str(len(n)))
    else:
        print(n[0] + str(len(n)-1))
    length += 1
file.close()
```

I am using `len(n) - 1` because there is an extra space (" ") on all of the lines except for the last line. I opted not to use the `readlines()` method, but maybe I should have?

Thank you for the help!



Richard Villarreal

19.1K views • 2 years



## Q&A Discussions



A short way to do it is like this:



+45

```
file = open("/usercode/files/books.txt", "r")
for line in file.readlines():
    print(f'{line[0]}'{len(line.strip())})
file.close()
```

Lothar

2 years



Yup I did this way ,just used readlines() similar to what your fileLines list holds after looping over lines in file, and then checked if last character in line contains "\n" or not ,

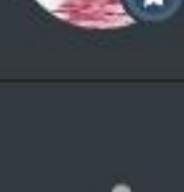


+21

```
for lines in file.readlines():
if lines[-1]=="\n":
    print(lines[0]+str(len(lines)-1))
else:
    print(lines[0]+str(len(lines)))
```

Abhay

2 years



+21

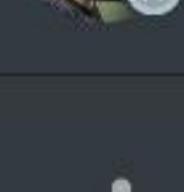
```
file = open("/usercode/files/books.txt", "r")
for x in file.readlines():
    x = x.strip('\n')
```



```
a = len(x)
b = x[0]
print(str(b) + str(a))
file.close()
```

Chris Wiseman

2 years



+10

Instead of dealing with \n we can simply remove it with the help of strip()



```
file = open("/usercode/files/books.txt", "r")
```

```
#your code goes here
books = file.readlines()
books = [x.strip() for x in books]
```



## Working with Files



PROBLEM

CODE

RESULT

```
1 with open("/usercode/files/  
books.txt") as f:  
2     #your code goes here  
3  
4 file = open("/usercode/files/  
books.txt", "r")  
5 for line in file.readlines():  
6     print(f'{line[0]}  
{len(line.strip())}')  
7 file.close()
```



TAB

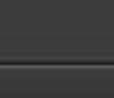
( )

"

'

{ }

RUN ▶



あ



...

int

Scanner

static

1 2 3 4 5 6 7 8 9 0

% ^ ~ | [ ] &lt; &gt; { }

q w e r t y u i o p

@ # &amp; \* - + = ( )

a s d f g h j k l

↑ - \$ " ' : ; / ×

z x c v b n m ←

123 ☺ , en/es/pt ... !? @ . ←

☰ ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋

## Working with Files



### PROBLEM

### CODE

### RESULT

```
1 with open("/usercode/files/  
books.txt") as f:  
2     #your code goes here  
3  
4     file = open("/usercode/files/  
books.txt", "r")  
5     for line in file.readlines():  
6         print(f'{len(line[0])}  
{len(line.strip())}')  
7     file.close()
```



TAB

()

"

'

{

RUN ▶



aあ



...



e

é

não

1

2

3

4

5

6

7

8

9

0

%

^

~

|

[

]

&lt;

&gt;

{

}

q

w

e

r

t

y

u

i

o

p

@

#

&amp;

\*

-

+

=

()

)



-

\$

"

'

:

;

/



123



,



en/es/pt



@

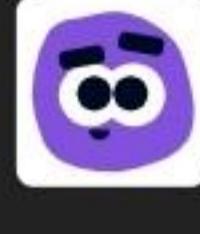
,!?



## Working with Files

**PROBLEM****CODE****RESULT**

```
1 with open("/usercode/files/  
books.txt") as f:  
2     #your code goes here  
3     books = f.readlines()  
4  
5     for x in range(len(books)):  
6         words = books[x].split()  
7         print("Line " + str(x+1) +  
": " + str(len(words)) + " words")  
8  
9  
10  
11
```



TAB

( )

"

'

{

RUN



20:08

12%



Open the file in binary write mode.

wb

19 COMMENTS

That's it!

CONTINUE





Fill in the blanks to try to open and read from a file. Print an error message in case of an exception.

try:

    with open("test.txt") as f

:

    print(f.read())

except

    print("Error")

ANSWER

CHECK





Fill in the blanks to read the content of a file called "records.txt" and output the number of characters it has.

f = open ("records.txt", "r")

cont = f .read()

print(len(cont))

f.close()

3 COMMENTS

Nice one!

CONTINUE



What does the `readlines()` method return?

a filename

a list

a string

nothing

3 COMMENTS



3 in a row!

CONTINUE



## ← Python Developer

# Congratulations!

COURSE CERTIFICATE

This is to certify that

**Jefferson Firmino**

has successfully completed the course by demonstrating  
theoretical and practical understanding of

**Python Developer**



 sololearn

CertID: CC-V2HXMCG43  
Issued 21 October, 2023

Yeva Hyusyan  
Chief Executive Officer

You have successfully completed  
the course!

[https://api2.sololearn.com/...](https://api2.sololearn.com/)

**COPY**

**SHARE**

**SAVE**



Learn



Community



Leaderboard



Create



Profile



# Leaderboard

**SCORES****EARN XP**

## Saturn League

Ends in 6d, 20h

1

**Jefferson Firmino****PRO**

300 XP

2

**ElMige xd**

165 XP

3

**Adam Idrissu**

100 XP

4

**Bruno Boal**

65 XP

5

**Dyari Elkamel**

45 XP

6

**Alena**

42 XP

7

**Nitr0Skay**

40 XP



Learn



Community



Leaderboard



Create



Profile



## ← Leaderboard

[FOLLOWING](#)[LOCAL](#)[GLOBAL](#)

Michelle

+819 XP

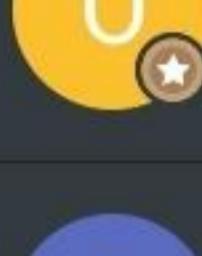
7851



Mohamed Sharik

+819 XP

7851



Ogorogile

+819 XP

7851



Da Programmer

+819 XP

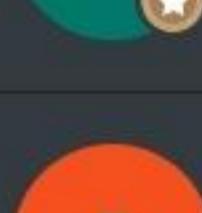
7851



Abdelmenam Khaled Elsobky

+819 XP

7851



saicharan teja

+819 XP

7851



admyn s

+818 XP

7871



Kaelyn Revan

+818 XP

7871



XYZ

+818 XP

7871



George Ismailov

PRO

+818 XP

7871



Jefferson Firmino

PRO

+818 XP

7871





The **SELECT** command is used to extract field data from a table.

Create a query to extract the **title** field from the movies table

```
SELECT title
```

```
FROM movies
```

```
retrieve
```

```
get
```

19 COMMENTS

Well done

CONTINUE