

Python Code Samples: Calculations, Games and Utilities

This repository provides Python code that you can use to enhance your project with popular functionalities such as calculations, games, and utilities such as CPF generators and palindrome checkers. The code examples highlight Python's fundamental concepts by demonstrating easiest ways to apply it in the right way.



by **Fabrício Vieira**

Car Rental Calculator



Este programa calcula o preço do aluguel de um carro com base na quantidade de dias que o carro foi alugado e na quantidade de quilômetros percorridos. O usuário informa a quantidade de dias e a quantidade de quilômetros, e o programa calcula o preço total do aluguel. O preço por dia é fixado em R\$60 e o preço por quilômetro é fixado em R\$0,15. O programa realiza os cálculos necessários e exibe o resultado na tela, informando o total a ser pago.



This program calculates the price of renting a car based on the number of days the car was rented and the number of kilometers traveled. The user enters the number of days and the number of kilometers, and the program calculates the total price of the rental. The price per day is set at £60 and the price per kilometer is set at £0.15. The program performs the necessary calculations and displays the result on the screen, showing the total amount to be paid.



Calculate Discounted Price



Este programa calcula o preço com desconto de um produto com base no preço original e na porcentagem de desconto informada pelo usuário. O usuário informa o preço original do produto e a porcentagem de desconto, e o programa realiza os cálculos necessários para obter o preço com desconto. O programa exibe o preço com desconto na tela.



This program calculates the discounted price of a product based on the original price and the discount percentage entered by the user. The user enters the original price of the product and the discount percentage, and the program performs the necessary calculations to obtain the discounted price. The program displays the discounted price on the screen.



Made with Gamma

Finding the Hypotenuse of a Right Triangle



Este programa calcula a hipotenusa de um triângulo retângulo com base nos comprimentos dos catetos informados pelo usuário. O usuário informa o comprimento do cateto oposto e do cateto adjacente, e o programa realiza os cálculos necessários para obter a hipotenusa. O programa exibe o valor da hipotenusa na tela.



This program calculates the hypotenuse of a right triangle based on the lengths of the opposite side and the adjacent side entered by the user. The user enters the length of the opposite side and the length of the adjacent side, and the program performs the necessary calculations to obtain the hypotenuse. The program displays the value of the hypotenuse on the screen.



Made with Gamma

Concession Stand Order



Este código simula o processo de fazer pedidos em um estande de concessão em um jogo de beisebol. O programa pede que o usuário faça um pedido, escolhendo entre as opções disponíveis, como pipoca, Coca-Cola, pizza, cheeseburger, nachos e água. O programa armazena os pedidos em uma lista e continua a perguntar se há algo mais a ser pedido até que o usuário não queira mais. Em seguida, o programa calcula o custo total dos pedidos com base nos preços de cada item e adiciona a taxa de 7% sobre o valor total. Por fim, exibe o pedido final e o custo total a ser pago.



This code simulates the process of placing orders at a concession stand during a baseball game. The program prompts the user to place an order, choosing from available options such as popcorn, Coca-Cola, pizza, cheeseburger, nachos, and water. The program stores the orders in a list and continues to ask if there is anything else to be ordered until the user does not want to order anymore. Then, the program calculates the total cost of the orders based on the prices of each item and adds the 7% tax on the total amount. Finally, it displays the final order and the total cost to be paid.

Character Count



Este código conta a quantidade de caracteres em um texto fornecido pelo usuário. Primeiro, o programa solicita que o usuário digite um texto. Em seguida, ele percorre cada caractere do texto e os conta usando um dicionário chamado **character_count**. Se um caractere já estiver presente no dicionário, o valor correspondente é incrementado em 1. Caso contrário, o caractere é adicionado ao dicionário com o valor inicial de 1. Por fim, o programa exibe a contagem de caracteres, substituindo o caractere de espaço em branco por "espaço" para melhor visualização.



This code counts the number of characters in a text provided by the user. First, the program asks the user to enter a text. Then, it iterates through each character of the text and counts them using a dictionary called **character_count**. If a character is already present in the dictionary, its corresponding value is incremented by 1. Otherwise, the character is added to the dictionary with an initial value of 1. Finally, the program displays the character count, replacing the whitespace character with "space" for better readability.



Hangman Game



Este código implementa o jogo da Forca (Hangman Game). O objetivo do jogo é adivinhar uma palavra secreta escolhida aleatoriamente da lista de palavras fornecida.

O jogo começa com a opção de iniciar uma nova partida. Em seguida, uma palavra secreta é selecionada aleatoriamente da lista de palavras. O jogador pode escolher o número de tentativas que deseja ter para adivinhar a palavra.

Durante o jogo, o jogador pode optar por adivinhar uma letra ou a palavra inteira. Se o jogador optar por adivinhar uma letra, o programa verifica se a letra está presente na palavra secreta. Caso esteja, a letra é adicionada às letras corretas. Se não estiver, a letra é adicionada às letras erradas e uma tentativa é contabilizada. O programa exibe a quantidade de tentativas utilizadas.

Se o jogador optar por adivinhar a palavra inteira, ele deve digitar a palavra completa. Se a palavra digitada estiver correta, o jogador vence o jogo. Caso contrário, a palavra é adicionada às palavras erradas e uma tentativa é contabilizada. O programa exibe a quantidade de tentativas utilizadas.

A palavra secreta é exibida com letras ocultas, representadas por underscores. As letras corretas são reveladas enquanto as incorretas são exibidas em uma lista separada.

O jogo continua até que o jogador adivinhe corretamente a palavra secreta, atinja o número máximo de tentativas ou decida sair do jogo.

Resumo: O código implementa o jogo da Forca, onde o jogador deve adivinhar uma palavra secreta selecionada aleatoriamente. O programa permite que o jogador adivinhe uma letra ou a palavra inteira e exibe informações sobre as letras corretas, letras erradas, tentativas utilizadas e a palavra secreta com letras ocultas.



This code implements the Hangman Game. The objective of the game is to guess a secret word randomly chosen from a given word list.

The game starts with the option to start a new game. Then, a secret word is randomly selected from the word list. The player can choose the number of attempts they want to have to guess the word.

During the game, the player can choose to guess a letter or the entire word. If the player chooses to guess a letter, the program checks if the letter is present in the secret word. If it is, the letter is added to the correct letters. If it is not, the letter is added to the wrong letters, and an attempt is counted. The program displays the number of attempts used.

If the player chooses to guess the entire word, they must enter the complete word. If the entered word is correct, the player wins the game. Otherwise, the word is added to the wrong words, and an attempt is counted. The program displays the number of attempts used.

The secret word is displayed with hidden letters, represented by underscores. The correct letters are revealed while the incorrect letters are shown in a separate list.

The game continues until the player guesses the secret word correctly, reaches the maximum number of attempts, or decides to quit the game.

Summary: The code implements the Hangman Game, where the player must guess a secret word randomly selected. The program allows the player to guess a letter or the entire word and displays information about the correct letters, wrong letters, attempts used, and the secret word with hidden letters.

Finding the Largest and Smallest Numbers



Este código permite encontrar o maior e o menor valor entre três números digitados pelo usuário.

Primeiro, são coletados três valores digitados pelo usuário. Em seguida, os valores são comparados para encontrar o maior e o menor valor.

O maior valor é inicializado com o primeiro valor digitado, e o menor valor também é inicializado com o primeiro valor digitado.

Em seguida, os valores são comparados com o maior valor e o menor valor já existentes. Se um valor for maior que o maior valor atual, ele se torna o novo maior valor. Se um valor for menor que o menor valor atual, ele se torna o novo menor valor.

Por fim, os valores são exibidos na tela. O maior valor é exibido juntamente com a informação sobre qual valor foi digitado (primeiro, segundo ou terceiro). O menor valor é exibido na mesma linha, indicando qual valor foi digitado (primeiro, segundo ou terceiro).

Resumo: O código permite encontrar o maior e o menor valor entre três números digitados pelo usuário. Os valores são comparados e exibidos na tela, indicando qual valor é o maior e qual valor é o menor.



This code allows you to find the largest and smallest value among three numbers entered by the user.

First, three values are collected from the user. Then, the values are compared to find the largest and smallest value.

The largest value is initially set to the first entered value, and the smallest value is also initially set to the first entered value.

Next, the values are compared to the current largest value and smallest value. If a value is greater than the current largest value, it becomes the new largest value. If a value is smaller than the current smallest value, it becomes the new smallest value.

Finally, the values are displayed on the screen. The largest value is displayed along with information about which value was entered (first, second, or third). The smallest value is displayed on the same line, indicating which value was entered (first, second, or third).

Summary: The code allows you to find the largest and smallest value among three numbers entered by the user. The values are compared and displayed on the screen, indicating which value is the largest and which value is the smallest.

Calculator with While Loop



Este código implementa uma calculadora que pode realizar operações aritméticas básicas usando um loop while. O programa solicita que o usuário insira dois números e um operador. Em seguida, valida as entradas do usuário, verificando se os números são válidos e se o operador é válido. Em seguida, realiza os cálculos de acordo com o operador inserido e exibe o resultado na tela. O programa continua a pedir ao usuário por novos cálculos até que o usuário escolha sair.



This code implements a calculator that can perform basic arithmetic operations using a while loop. The program asks the user to enter two numbers and an operator. It then validates the user inputs, checking if the numbers are valid and if the operator is valid. It performs the calculations based on the entered operator and displays the result on the screen. The program continues to prompt the user for new calculations until the user chooses to exit.



Checking for Palindrome



Este código verifica se um texto digitado pelo usuário é um palíndromo.

Primeiro, é coletado o texto digitado pelo usuário. Em seguida, é feita a verificação para determinar se o texto é um palíndromo.

O código remove todos os espaços do texto usando o método `replace` e compara se o texto original, sem espaços, é igual à versão reversa do texto original, também sem espaços. Se a comparação for verdadeira, o texto é um palíndromo.

Se o texto for um palíndromo, uma mensagem é exibida indicando que o texto é um palíndromo, juntamente com a representação reversa do texto. Caso contrário, uma mensagem é exibida mostrando que o texto não é um palíndromo e apresentando a versão reversa do texto para comparação.

Resumo: O código verifica se um texto digitado pelo usuário é um palíndromo, comparando o texto original com sua versão reversa, ambos sem espaços.



This code checks if a text entered by the user is a palindrome.

First, the text is collected from the user. Then, a check is performed to determine if the text is a palindrome.

The code removes all spaces from the text using the `replace` method and compares if the original text, without spaces, is equal to the reversed version of the original text, also without spaces. If the comparison is true, the text is a palindrome.

If the text is a palindrome, a message is displayed indicating that the text is a palindrome, along with the reversed representation of the text. Otherwise, a message is displayed showing that the text is not a palindrome and presenting the reversed version of the text for comparison.

Summary: The code checks if a text entered by the user is a palindrome by comparing the original text with its reversed version, both without spaces.

Vowel Count



Este código conta o número de vogais em um texto digitado pelo usuário.

Primeiro, o texto é coletado do usuário. Em seguida, é feita a verificação para determinar se o texto contém vogais.

Se o texto contiver apenas dígitos, ou seja, não contiver letras, uma mensagem é exibida informando que nenhuma vogal foi encontrada no texto.

Caso contrário, o código percorre cada caractere do texto e verifica se é uma vogal. As vogais encontradas são armazenadas em uma lista chamada "vowels".

Ao final do código, são exibidos os resultados: o número de vogais identificadas e a lista de vogais encontradas. Além disso, o texto digitado pelo usuário também é mostrado.

Resumo: O código conta o número de vogais em um texto digitado pelo usuário, exibindo a quantidade de vogais encontradas e a lista de vogais, bem como o texto original digitado.



This code counts the number of vowels in a text entered by the user.

First, the text is collected from the user. Then, a check is performed to determine if the text contains vowels.

If the text contains only digits, meaning it does not contain any letters, a message is displayed indicating that no vowels were found in the text.

Otherwise, the code iterates through each character in the text and checks if it is a vowel. The found vowels are stored in a list called "vowels".

At the end of the code, the results are displayed: the number of identified vowels and the list of found vowels. Additionally, the original text entered by the user is also shown.

Summary: The code counts the number of vowels in a text entered by the user, displaying the count of vowels and the list of vowels found, as well as the original text entered.

Calculating Wall Area and Paint Quantity



Este código calcula a área de uma parede retangular e a quantidade de tinta necessária para pintá-la. O usuário fornece as dimensões da parede (largura e altura) em metros. Em seguida, a área da parede é calculada multiplicando a largura pela altura. A quantidade de tinta necessária é calculada dividindo a área por 2. Os resultados são exibidos na tela, mostrando a área da parede em metros quadrados e a quantidade de tinta em litros.



This code calculates the area of a rectangular wall and the quantity of paint required to paint it. The user provides the dimensions of the wall (width and height) in meters. Then, the wall area is calculated by multiplying the width by the height. The paint quantity is calculated by dividing the area by 2. The results are displayed on the screen, showing the wall area in square meters and the paint quantity in liters.



