TIPOS DE ESTRUTURAS

As estruturas básicas:

Estruturas sequenciais

Estruturas de decisão

Estruturas de repetição

ESTRUTURA DE REPETIÇÃO

Existem ocasiões em que é necessário repetir um determinado trecho do

programa um determinado número de vezes.

Se uma ação se repete em um algoritmo, ao invés de escrevê-la várias

vezes, em certos casos podemos resumir anotando uma vez só e

solicitando que ela se repita.

ESTRUTURA DE REPETIÇÃO

Para essas situações utilizaremos um laço de repetição, também

conhecido como looping, que efetuará o processamento de um trecho do

programa tantas vezes quantas forem necessárias.

ESTRUTURA DE REPETIÇÃO

Em Python, os loops são codificados por meio dos comandos for e

while.

O primeiro que veremos é o for. Ele nos permite percorrer os itens de

uma coleção e, para cada um deles, executar um bloco de código.

FOR (PARA)

Usada

para executar repetidamente um bloco

de código por um

número específico de vezes.

Essa estrutura tem o seu funcionamento controlado por uma variável

denominada contador.

FOR (PARA)

Sendo assim, todas as instruções que estiverem entre as instruções para ...

fim\_para serão executadas um determinado número de vez. Será necessário

indicar a variável a ser controlada com os seguintes valores: início, fim e

incremento.

Exemplo:

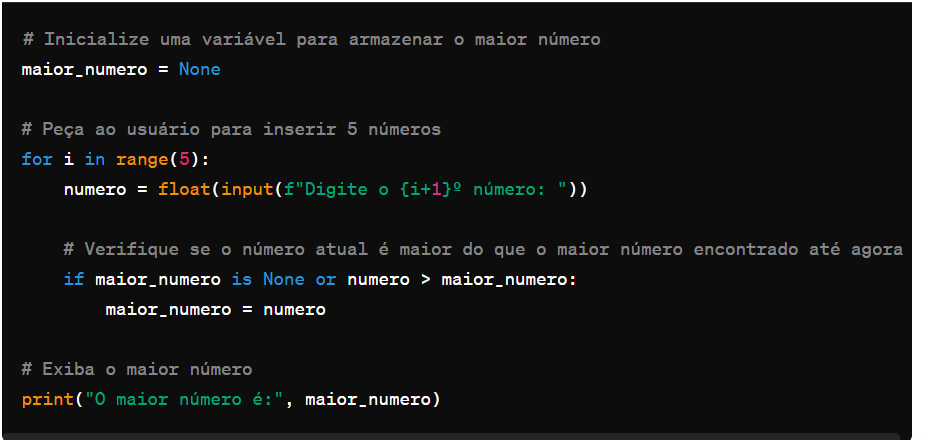
Para <contador> de <inicio> até <fim> passo <incremento> faça

<instruções a serem executadas>

fim\_para

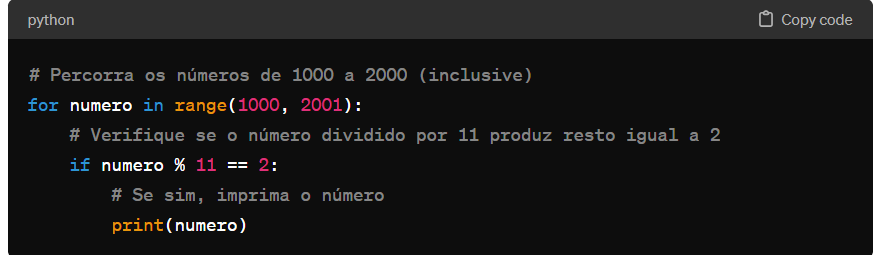
ATVIDADE

1. Faça um programa que leia 5 números e informe o maior número.



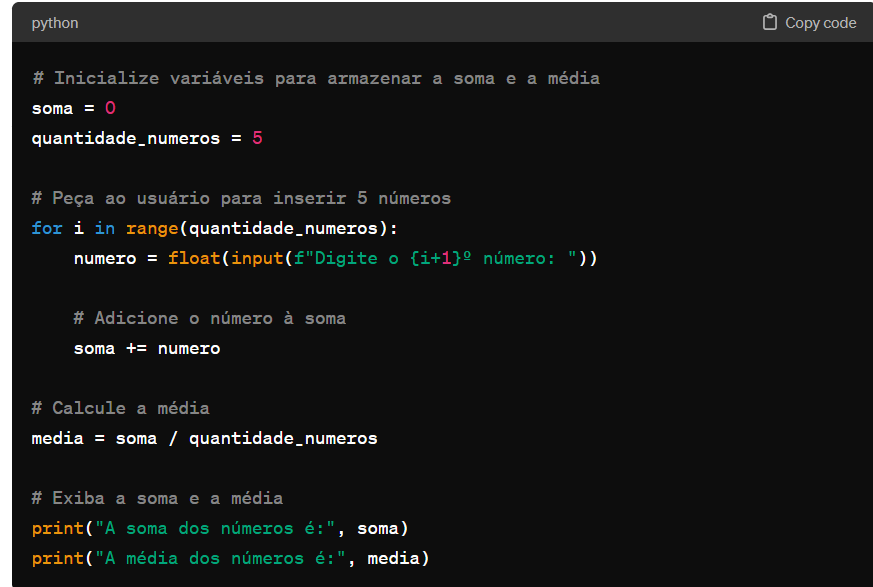
2. Faça um programa que verifique e mostre os números entre 1.000 e 2.000

(inclusive) que, quando divididos por 11 produzam resto igual a 2.



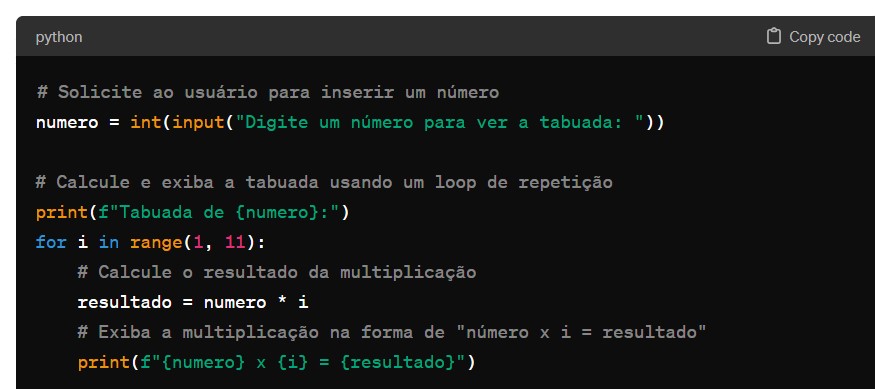
3. Faça um programa que leia 5 números e informe a soma e a média dos

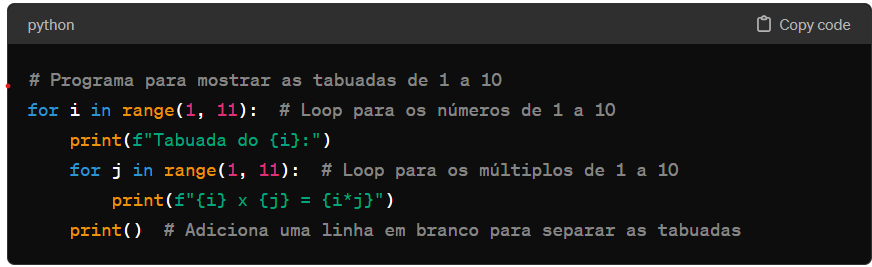
números.



4. Faça um programa que receba um número e usando laços de repetição

calcule e mostre a tabuada desse número.

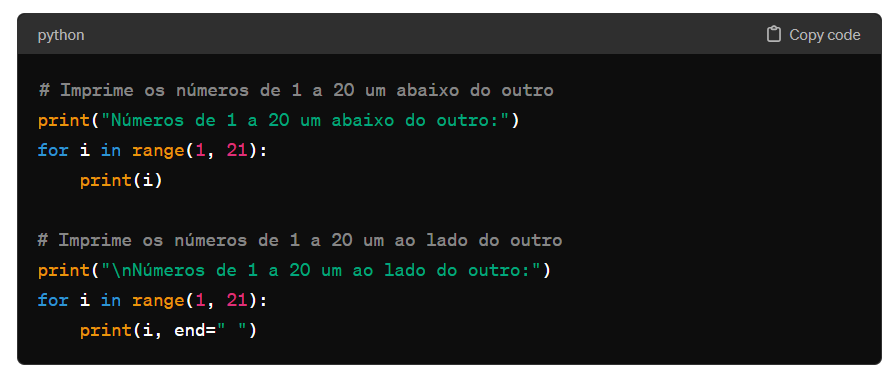


5. Faça um programa que mostre as tabuadas dos números de 1 a 10 usando laços de repetição 

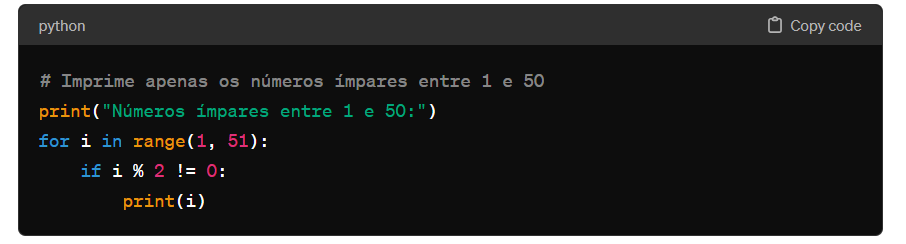
6. Faça um programa que imprima na tela os números de 1 a 20, um abaixo

do outro. Depois modifique o programa para que ele mostre os números

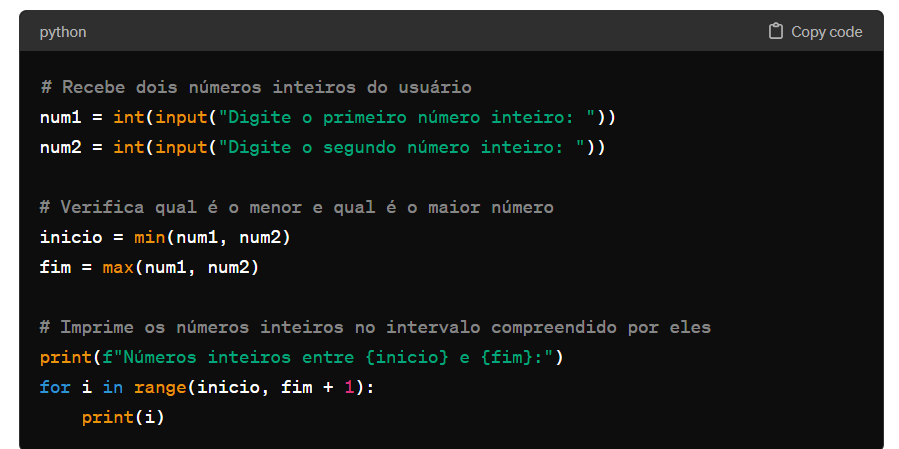
um ao lado do outro.



7. Faça um programa que imprima na tela apenas os números ímpares entre

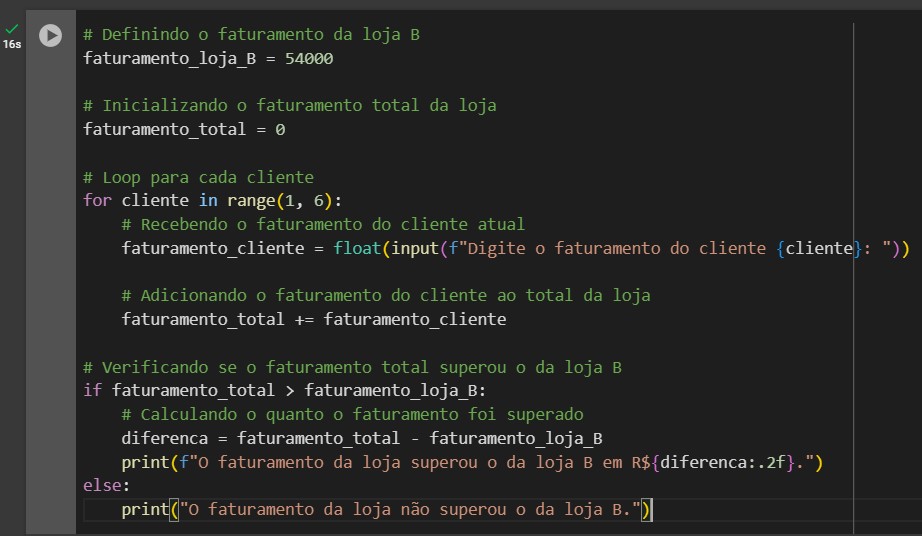
1 e 50. 

8. Faça um programa que receba dois números inteiros e gere os números

inteiros que estão no intervalo compreendido por eles. 

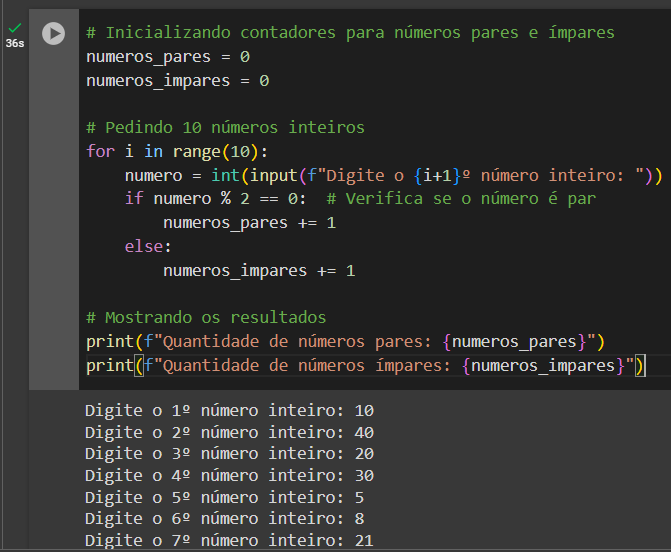
9. Uma loja deseja cadastrar 5 clientes e verificar se o faturamento da loja foi

superior a loja B (faturamento = 54000). Se o faturamento atingir esse valor

mostre na tela uma mensagem contendo em quanto foi superado o faturamento. 

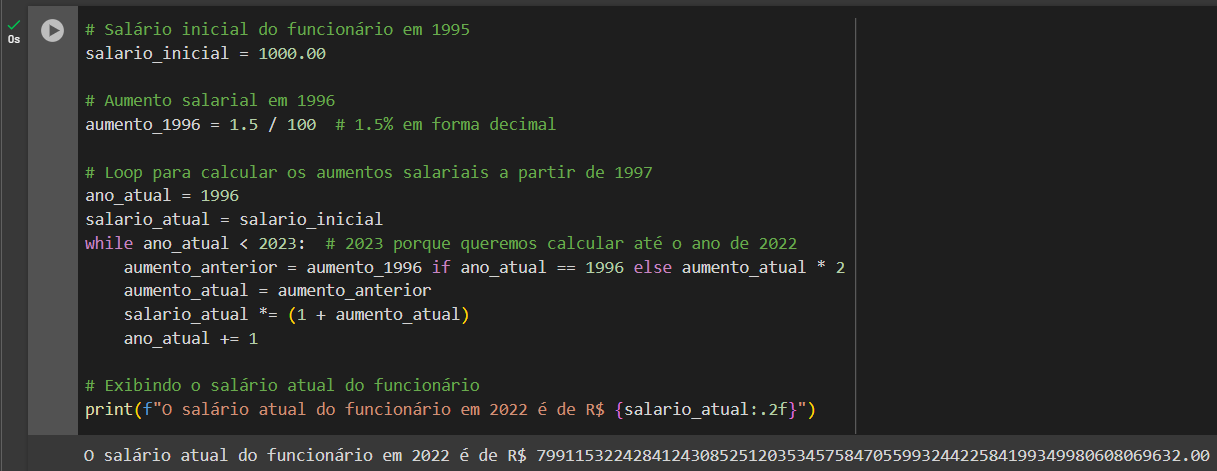
10. Faça um programa que peça 10 números inteiros, calcule e mostre a

quantidade de números pares e a quantidade de números impares.



11. Um funcionário de uma empresa recebe aumento salarial anualmente:

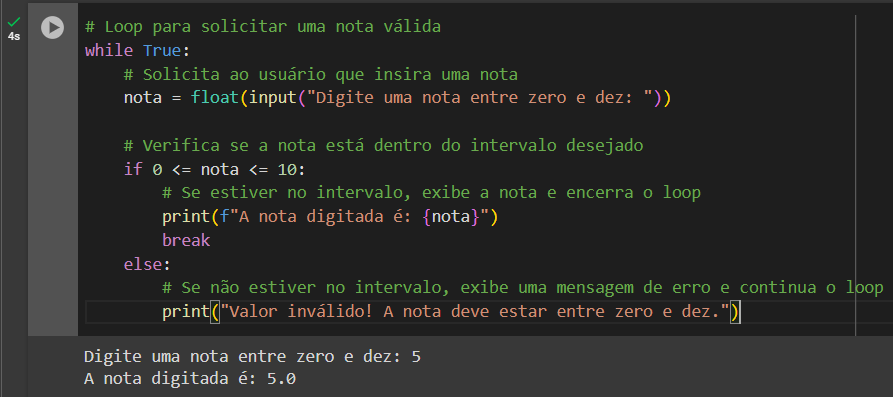
Sabe-se que:



12. Faça um programa que peça uma nota, entre zero e dez. Mostre uma

mensagem caso o valor seja inválido e continue pedindo até que o usuário

informe um valor válido.



13. Uma loja tem uma política de descontos de acordo com o valor da

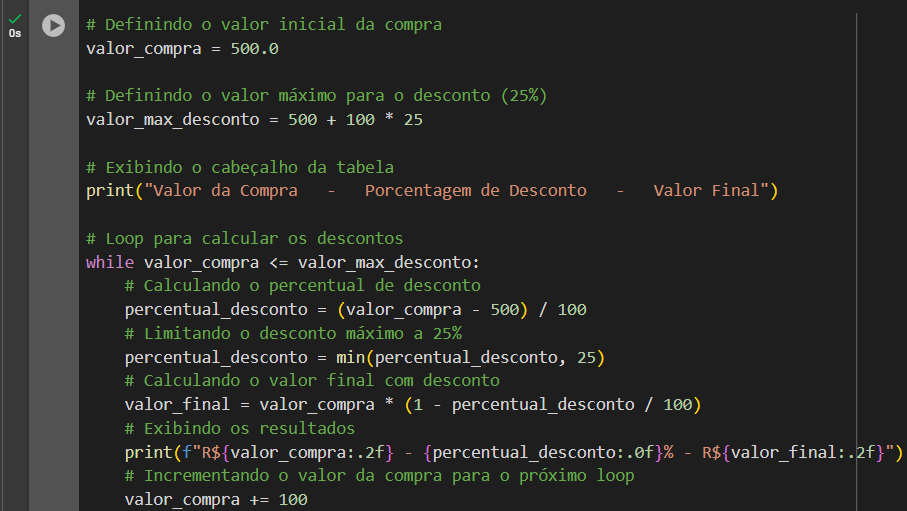
compra do cliente. Os descontos começam acima dos R$500. A cada 100

reais acima dos R$500,00 o cliente ganha 1% de desconto cumulativo até

25%.

Por exemplo: R$500 = 1% || R$600,00 = 2% ... etc...

Faça um programa que exiba essa tabela de descontos no seguinte formato:

Valor da compra porcentagem de desconto – valor final

14. Faça um programa que receba a idade de 15 pessoas e que calcule e

mostre:

a) A quantidade de pessoas em cada faixa etária;

b) A percentagem de pessoas na primeira e na última faixa etária, com relação

ao total de pessoas:

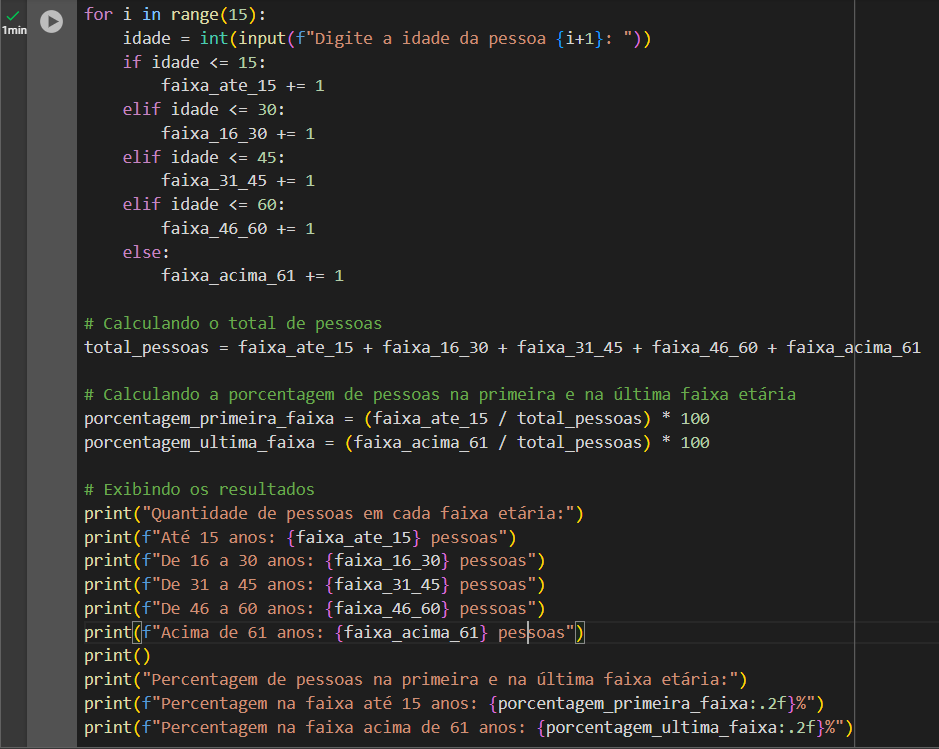
 Até 15 anos

 De 16 a 30 anos

 De 31 a 45 anos

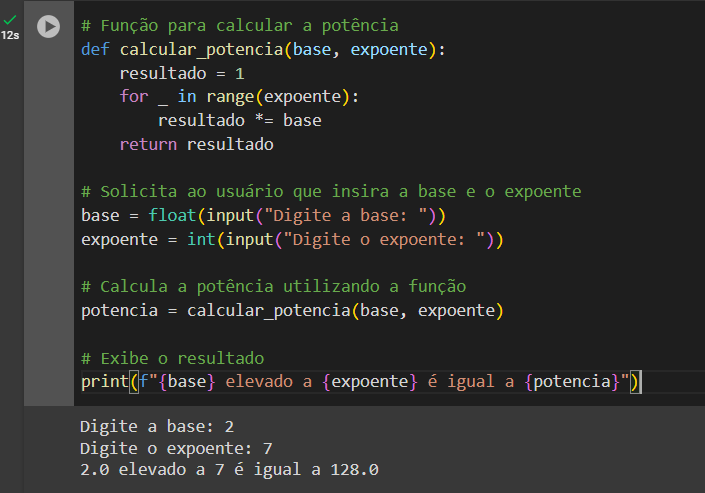
 De 46 a 60 anos

 Acima de 61 anos



15. Faça um programa que peça dois números, base e expoente, calcule e

mostre o primeiro número elevado ao segundo número. Não utilize a

função de potência da linguagem. 

16. Faça um programa que mostre todos os primos entre 1 e N sendo N um

número inteiro fornecido pelo usuário. O programa deverá mostrar também

o número de divisões que ele executou para encontrar os números primos.

Serão avaliados o funcionamento, o estilo e o número de testes (divisões)

executados. 