**Curso em vídeo – Python**

**Curso Python#07 – Operadores Aritméticos**

**OBS:** **CTRL+/ = comenta varias coisas.**

Símbolos aritméticos de Python:

+ Adição | 5+2 == 7

- Subtração | 5-2 == 3

\* Multiplicação | 5\*2 == 10

/ Divisão | 5/2 == 2.5

\*\* Potência | 5\*\*2 == 25

// Divisão inteira | 5//2 == 2

% Resto da Divisão | 5%2 == 1

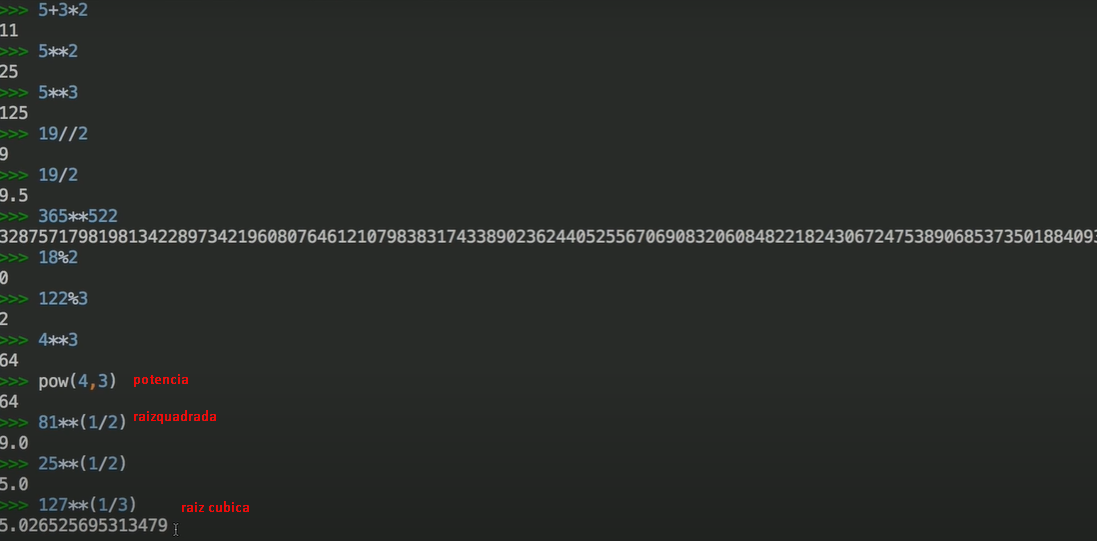
Ordem de Precedência:

1: ()

2: \*\*

3:\*, /, //, %

4:+, -

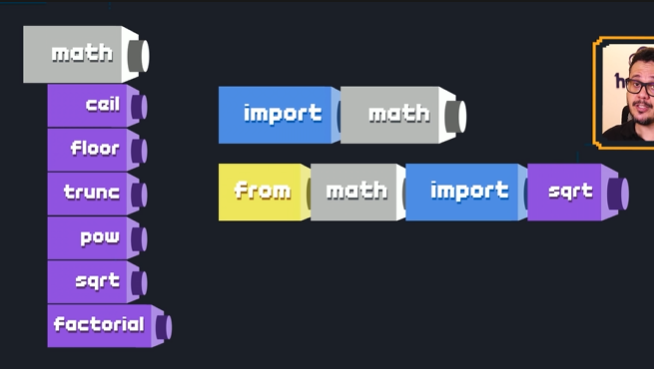


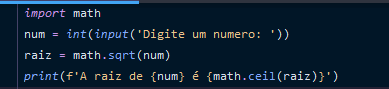
**Curso em Video#08 – Utilizando Módulos**

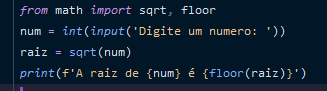
Como usar módulos? Exemplo:

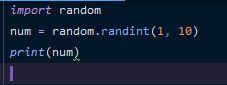
import math

from math import sqrt, ceil



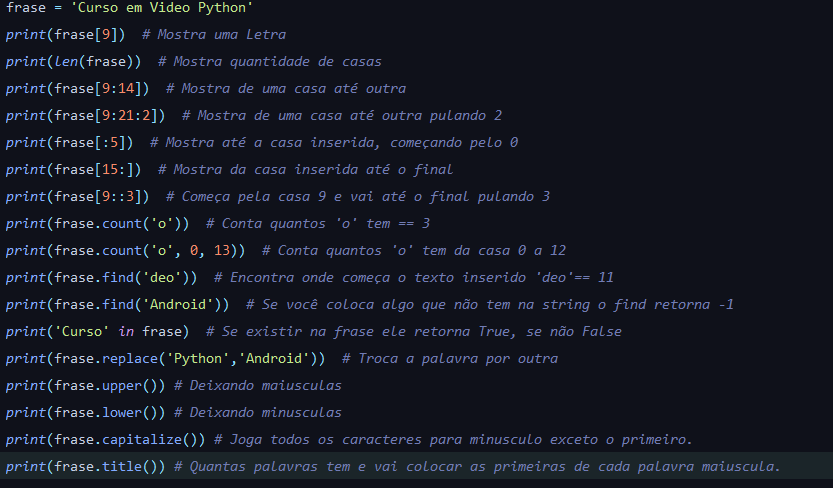


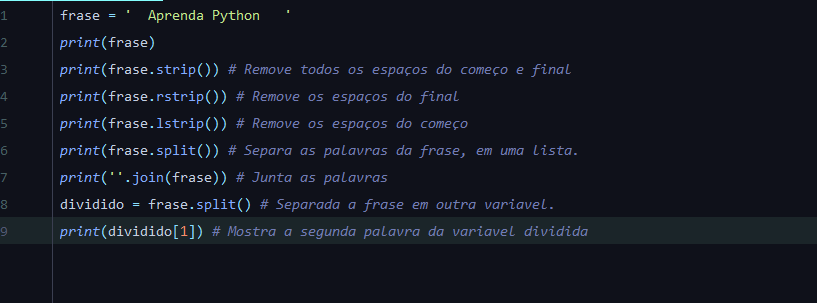






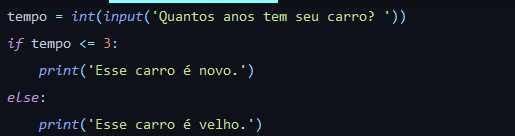
**Curso Python #09 – Manipulando Texto.**

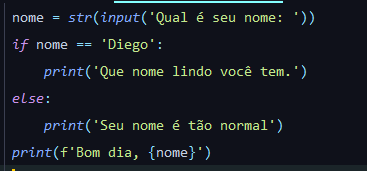


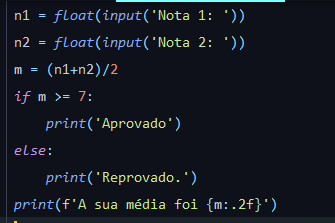


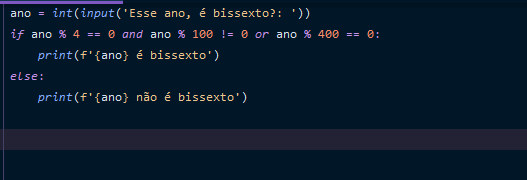
**Curso Python #10 - Condições (Parte 1)**

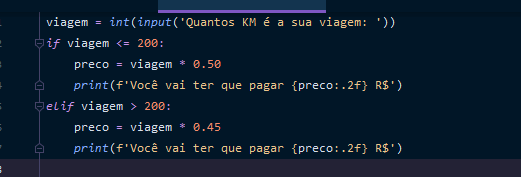
**Exemplo de IF e ELSE.**

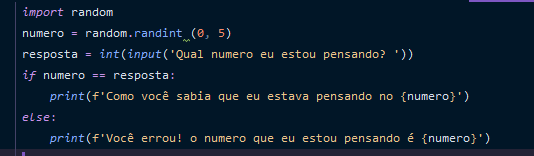


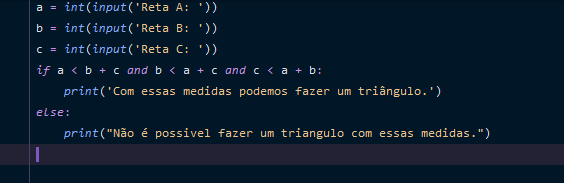












**Curso Python #012 - Condições Aninhadas**

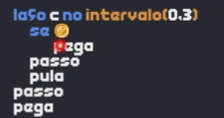
**Exemplos de if, elif e else**

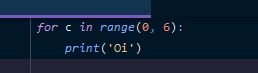


**Curso Python #013 - Estrutura de repetição for**

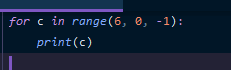


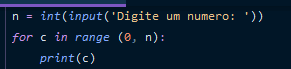


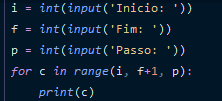


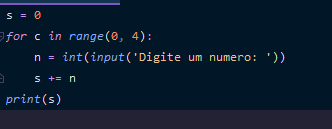








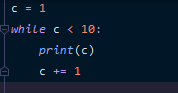




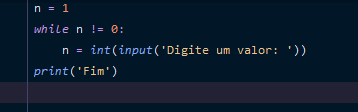
**Curso Python #014 - Estrutura de repetição while**

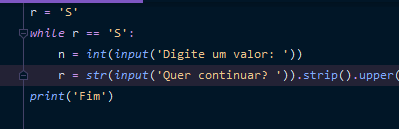


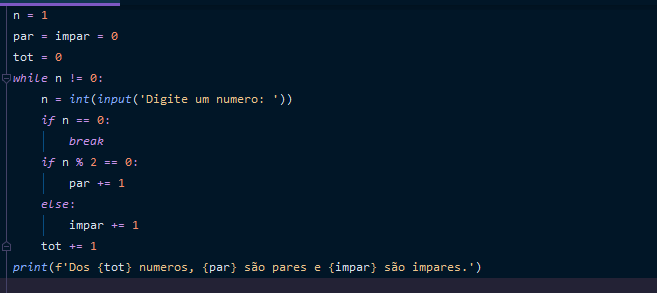


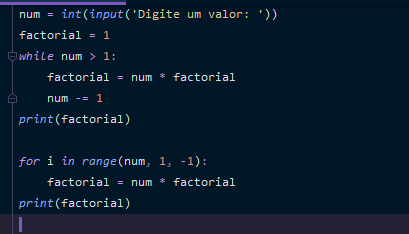


**PONTO DE PARADA (Flag)**

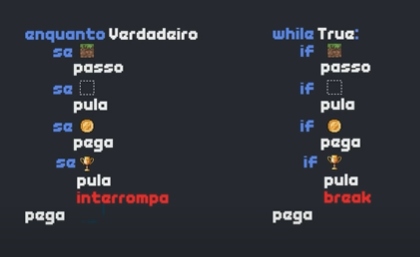


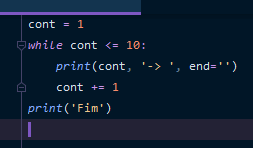


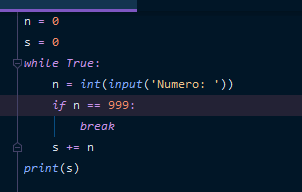




**Curso Python #15 - Interrompendo repetições while**



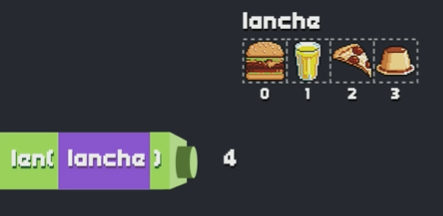


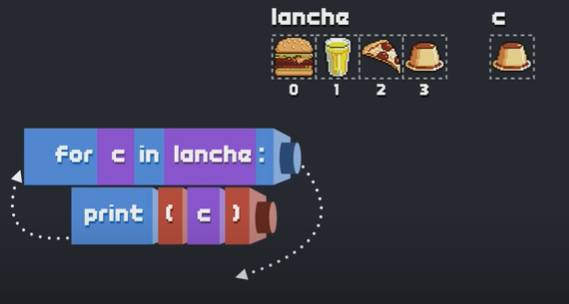


**Curso Python #16 – Tuplas**









**AS TUPLAS SÃO IMUTAVEIS**



**() Tuplas, [] Listas, {} Dicionario**



**Quero ver o que tem na casa 0>1>2 = Pizza**



**Agora quero ver de traz pra frente o que na casa 0>-1>-2 = Pizza**



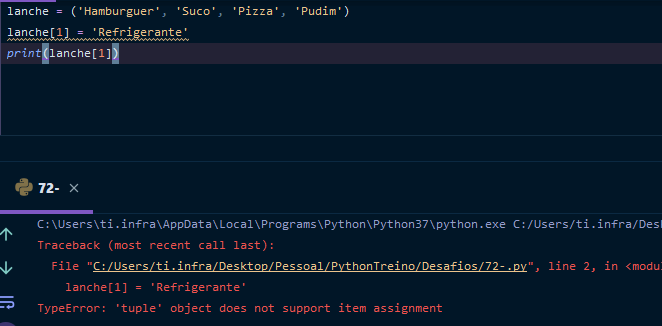
**O que tem da casa 1 a 4 = Suco, Pizza, Pudim, pq o Hambúrguer é o 0**



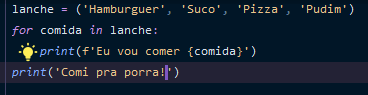
**O que tem a partir do 2? 0>1>2>3 = Pizza e Pudim**



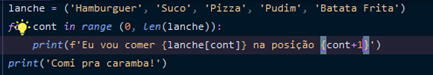
**Tentando alterar uma tuple.**



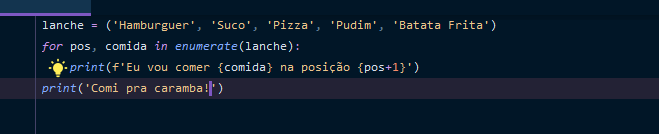
**Usando for pra mostrar todos os nomes.**



**Mesma coisa só que utilizando o len, assim podendo ler a posição.**



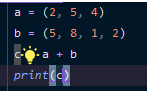
**Mesma coisa só que sem len, utilizando duas variáveis e o comando enumerate.**



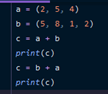
**Ordenando em ordem alfabética minha tupla.**



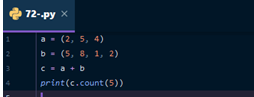
**Juntando tuplas**



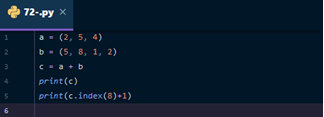
**Não são iguais...**



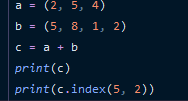
**Quantos 5 tem na tupla? = 2**



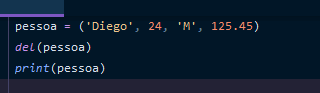
**Em que casa está aparecendo o 8 = 4 mas por que eu coloquei +1 então é 5.**



**Onde está o 5 depois da casa 2 = 2, 5, 4, 5 = 3**



**Deletando tupla**



**Curso Python #17 - Listas (Parte 1)**



**Trocando valor da Lista**



**Adicionando valores no final da Lista**



**Adicionando valor em uma casa especifica.**



**Como remover os dados de uma Lista**



**Del = deleta uma casa especifica**

**Pop = usado para deleta o ultimo elemento da Lista, mas podemos especificar a casa em questão**

**Remover para remover dizendo o valor da casa**

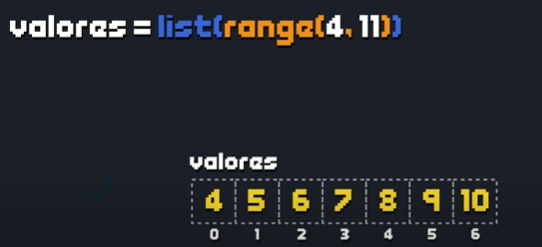
**O resultado dos 3 é a lista abaixo.**

 **Exclui o valor/casa e refaz a lista.**

**Verificando se tem o valor em listas.**



**Criando uma lista com valores, utilizando o range**



**Colocando em ordem os valores.**



**Caso precise fazer do menor pro maior...**



**Contando quantas casas tem com a função len**



**Listas diferente de Tupla são mutáveis...**



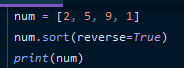
**Para inserir valores, utilizamos o append.**



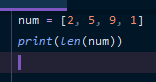
**Lista em ordem crescente.**



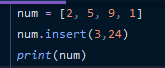
**Lista em ordem decrescente.**



**Verificando quantas casas tem com len.**



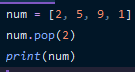
**Inserindo valores em casa especifica com o insert. (Inseri na casa 3 0>1>2>3 o valor 24 e empurrei o 9 para o 4 e o 1 para o 5).**



**Removendo o ultimo valor com o pop.**



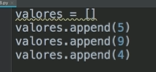
**Removendo o valor da casa predeterminada com o pop. (o 9 vai ser removido)**



**Removendo um valor com o remove.**



**Nesse exemplo podemos ver que tem 2 números 2, o remove vai apenas identificar a primeiro ocorrência e remover ela, ele não tira todos os 2.**



**Podemos também adicionar valores a uma lista vazia.**



**Adicionando valores em uma lista pelo for.**

**Agora vamos ver o que acontece se você tenta fazer uma cópia da lista de forma simples.**



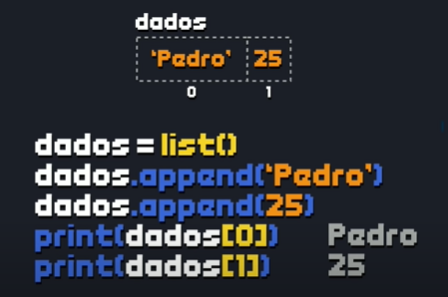
**Mesmo você informando que quer alterar no B, ele mudou em ambos...**

**Por que o Python utilizando b = a está fazendo uma ligação de listas, não uma cópia.**

**Agora se você utilizar b = a[:] (Todos os dados de A, ele vai fazer uma cópia não uma ligação.**

**Curso Python #18 - Listas (Parte 2)**

**Analisando dados de uma Lista**



**Inserindo uma lista em outra lista [:]**



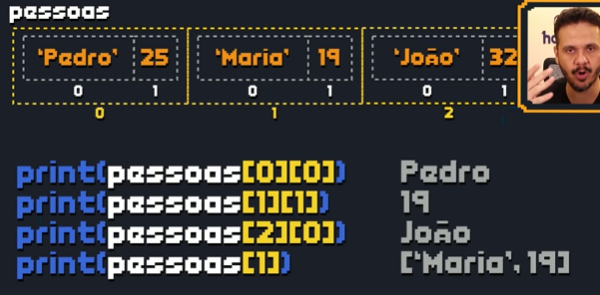
**Inserindo dados de uma lista (3 nomes e idades) de uma vez em outra Lista.**



**Resultado abaixo.**



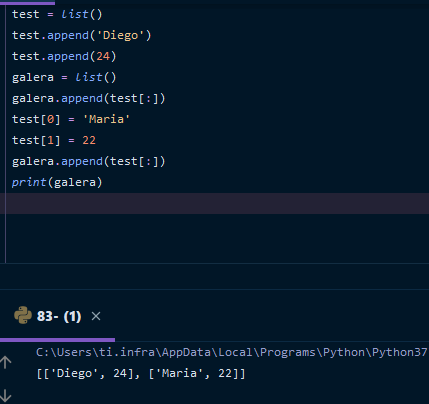
**Fatiando listas dentro de listas.**



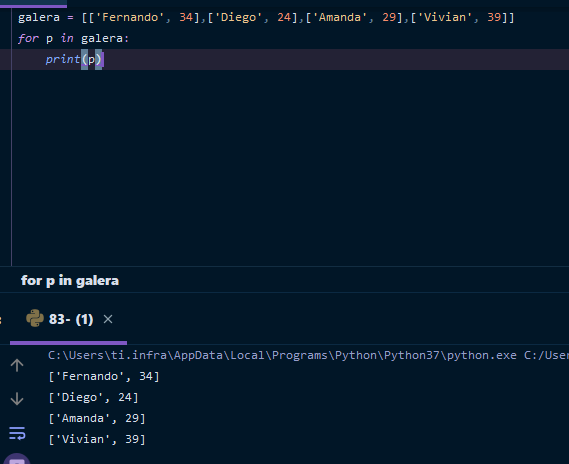
**Se você mudar algum dado de uma lista e inserir ela sem [:] ira mudar em ambas pois não são cópias e sim listas relacionadas uma na outra.**



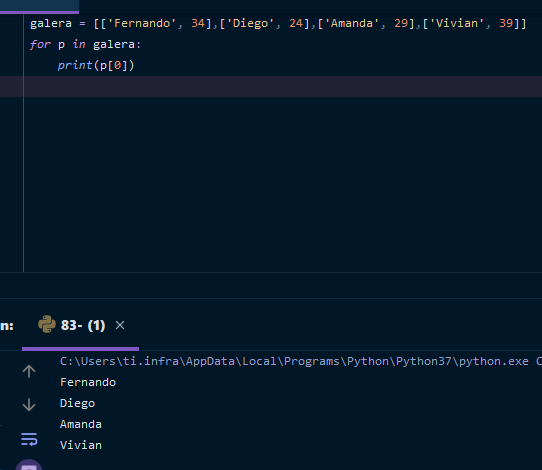
**Abaixo vemos, que primeiramente inserimos Diego,24 fizemos uma copia dessa lista em outra lista (galera) depois alteramos os 2 primeiros dados da primeira lista e depois copiamos os novos valores de (test) agora foi inserido mais dados na (galera)**



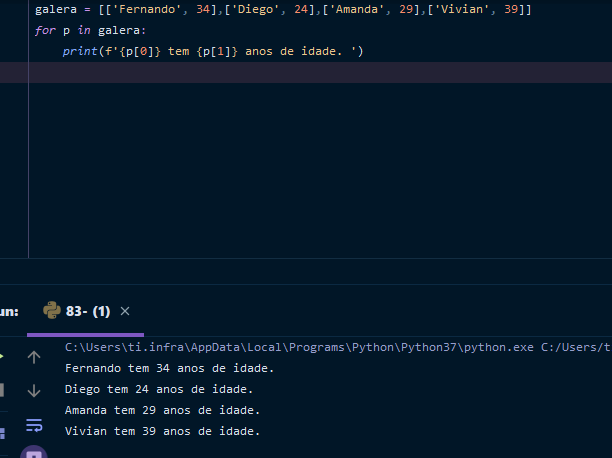
**Se você quiser ver listas dentro de outra lista basta utilizar o for sem fatiação.**



**Agora se você quiser ver uma casa especifica de todas as listas dentro da lista, utilize a fatiação exemplo p[0]**



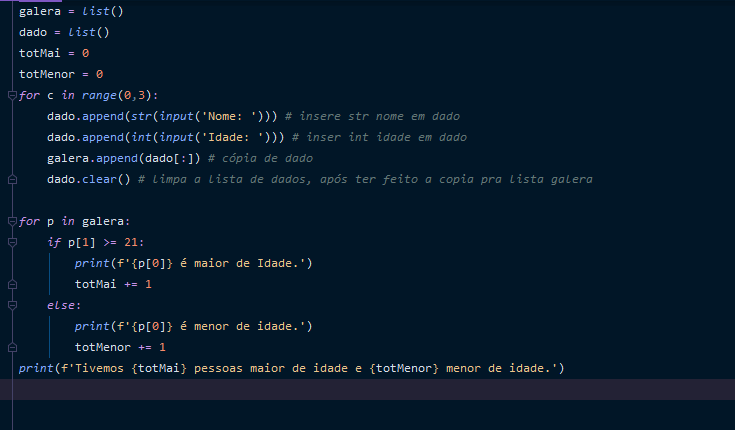
**Também podemos visualizar os 2 campos ou quantos tiver...**



**Abaixo tivemos que utilizar o dado.clear que toda vez no fim do laço for ele apaga os dados da lista dado para adicionar novos dados dentro da galera.**

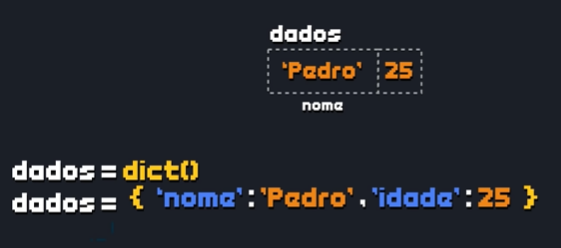


**Com um upgrade adicionando pra cada p na lista galera, verifica se o campo “idade” p[1] era maior ou menor que 21.**



**Curso Python #19 – Dicionários**

**Criando listas, temos o dict() ou inserindo manualmente dados.**



**Mostrando as Keys fora do fstring**



**Inserindo keeys e valores.**



**Deletando uma key e seus valores**



**Nesse dicionário temos 3 keys e 3 valores e 6 items.**

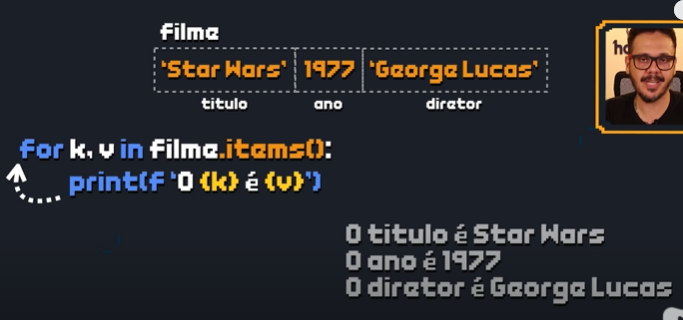








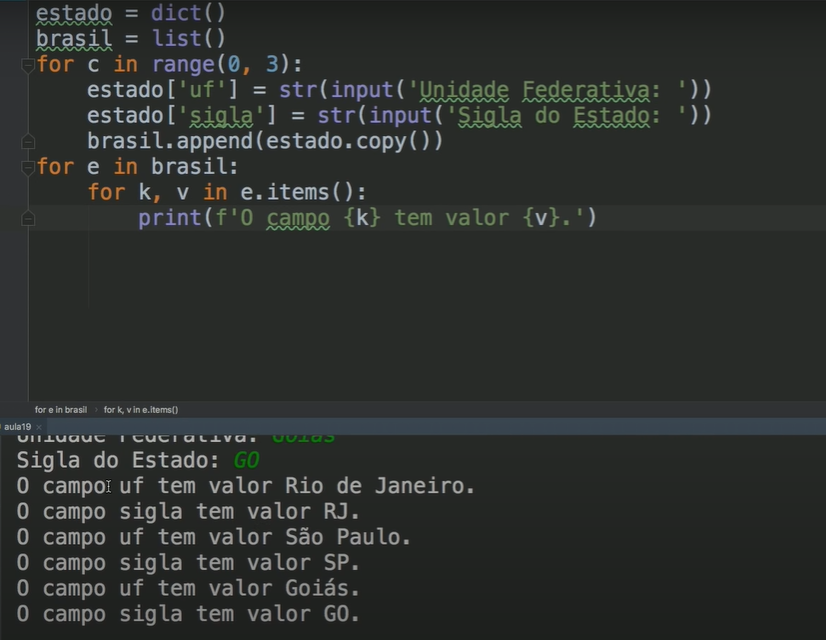
**Como ver as keys e values de um dicionário.**



**Como ver uma lista que contem dicionarios**

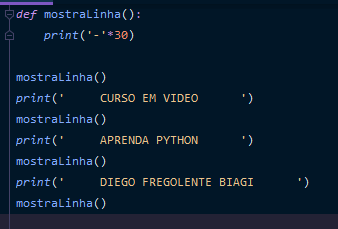


**Como adicionar dicionários em 1 lista e depois mostrar as keys e values dessa lista.**

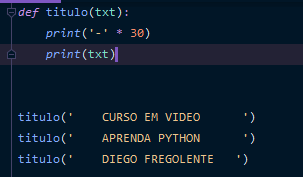


**Curso Python #20 - Funções (Parte 1)**

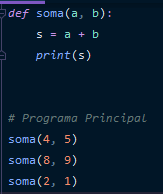
**Definindo uma função que crie uma linha com traços para separar títulos.**



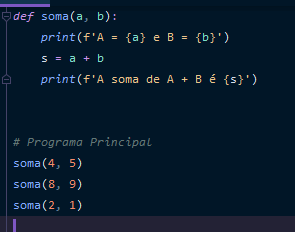
**Definindo uma função que receba uma texto e dentro dessa função execute a linha de títulos e o texto digitado na função.**



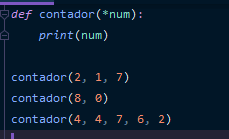
**Definindo a função soma() onde recebemos dois valores (a, b) dentro da função fazemos a soma dos 2 valores inseridos durante o código.**



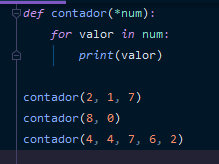
**Definindo a função soma(a, b) mostrando na tela quanto vale cada letra e mostrando o valor final da soma na tela.**



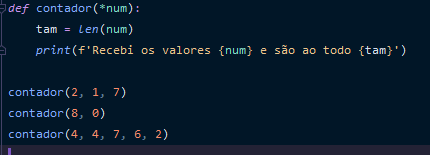
**Definindo a função contador (\*num) e mandando mostrar num, o \* tem a função de mostrar todos os nums.**



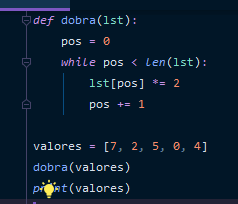
**Definindo a função contador(\*num) só que agora vamos separar os valores utilizando for e mostrar 1 valor de cada vez.**



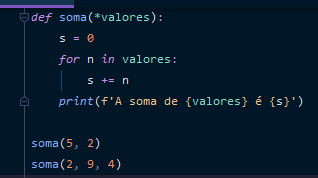
**Definindo contador(\*num), porém neste exemplo nele queremos saber o tamanho (casas) que tem em cada tupla (num) e mostrar na tela quais os valores (num) e quantas casa tem tam(len(num))**



**Definindo uma função dobra(lista) onde pegamos uma lista (valores) colocando em dobra e dentro dessa função ele vai definir a pos = 0, enquanto pos for menor que o tamanho de casas da lista ele vai fazer lista[pos] \*= 2, depois disso adicione 1 em pos para informar que essa casa já foi dobrada.**



**Definindo a soma de vários valores, temos o soma = 0 e um for n para cada casa em valores, dentro some os n em s, no final de todos os N mostre na tela o valor da sola de cada tupla, sem precisar determinar quantas casas são.**



**Curso Python #21 - Funções (Parte 2)**

**Função help() – inserindo a função desejada dentro dos parênteses o python mostra a documentação da função**



**Descrição do programa:**

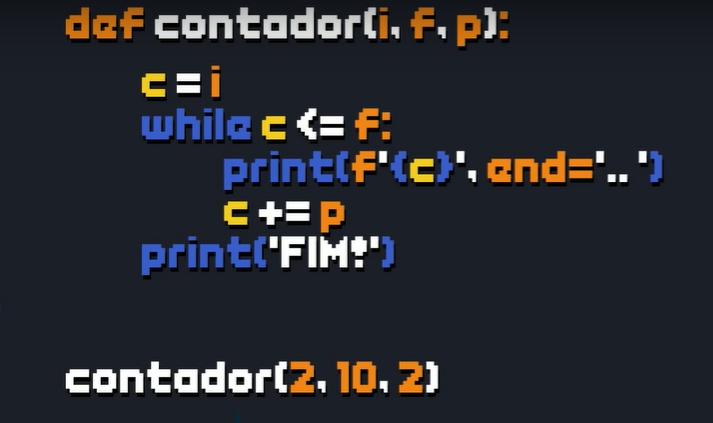
**Definimos a função contador, com os parâmetros inicio fim.**

**C vai receber o valor do inicio, enquanto o C for menor ou igual ao FIM.**

**Mostre o C, sem pular linha.**

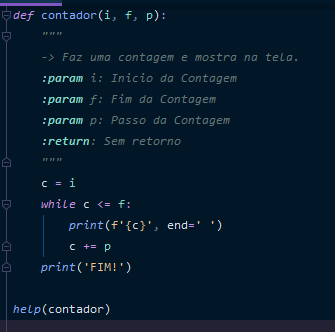
**E o C vai somar o valor de PASSO.**

**Quando o c chega no valor maior que o Fim, ele encerra.**



**Abaixo criamos o programa anterior, documentando a função com 3 aspas duplas.**

**Depois utilizamos o help() para ver a documentação da função criada por nós.**



**Abaixo mostramos como criar uma função que recebe 3 valores, e colocamos que caso não tenha o valor informado na hora de chamar a função ela recebe 0.**



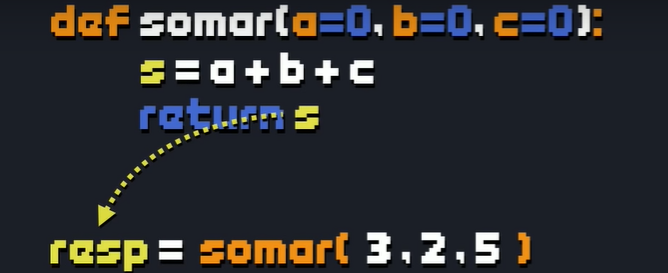
**Abaixo, mostramos como funciona uma variável local e global, temos o A dentro da função teste e o A fora da teste.**

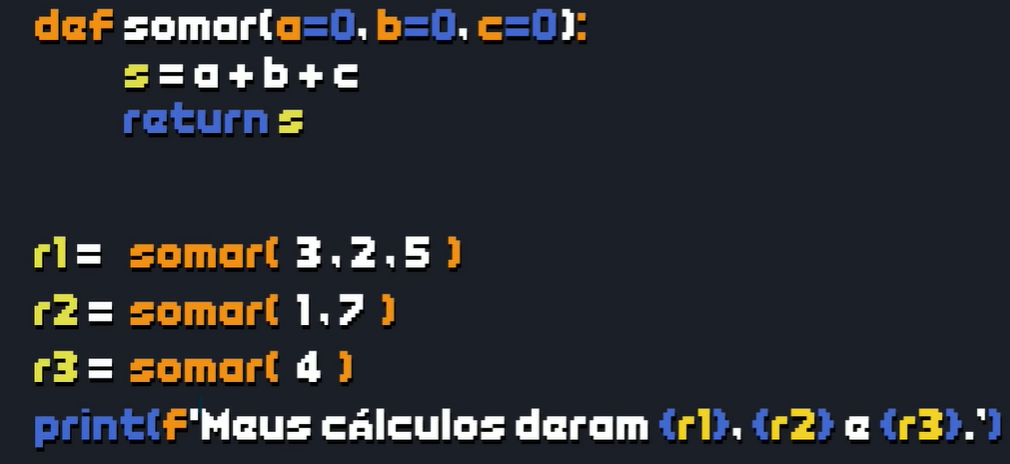


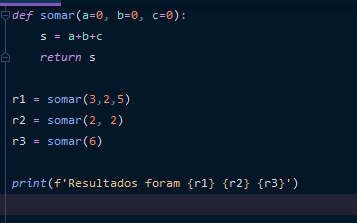
**Se quisermos informar que dentro de tal função iremos utilizar a variável global, basta informar como exemplo “global a”**



**Variavel que chama função e recebe o return “soma” da função somar.**

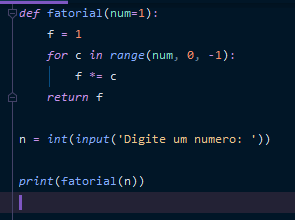






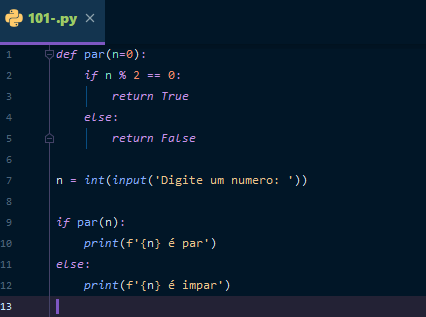
**Para calcularmos um fatorial com funções, precisamos saber de qual números queremos o fatorial.**

**Depois de definido o num, o fatorial inicial é 1, vamos fazer um for para que o c comece do numero definido e va até o 0, reduzindo de -1, toda vez que o for executar multiplicar o f por c, no final retornar o valor a f que será entre no print(fatorial(n))**



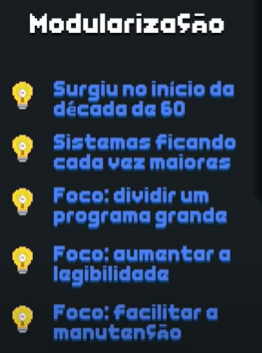
**Função que verifica se é par ou não.**

**Defina o numero, depois verifique se a função par(n) é True or False.**



**Curso Python #22 - Módulos e Pacotes**

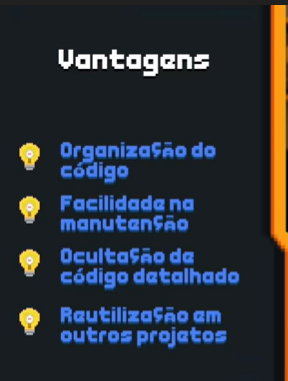
**Introdução a modularização.**



**Módulos do python x Módulos criados pelo usuário (uteis)**



**Vantagens de Módulos**



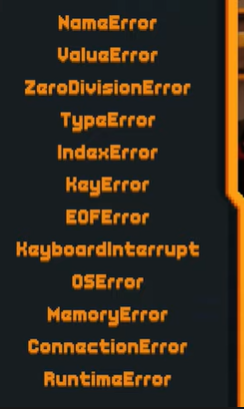
**Pacotes.**



**Curso Python #23 - Tratamento de Erros e Exceções**

**Os tipos de exceções que você pode receber do Python:**

[**https://docs.python.org/pt-br/3/library/exceptions.html**](https://docs.python.org/pt-br/3/library/exceptions.html)



**Tratando erro de forma simples, tente algo (try) se der errado (except)**

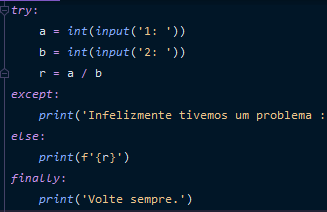


**Tratando erro de forma simples, tente algo (try) se der errado (except), se der certo (else), independente do que tenha acontecido (finally) also encerramento do (try).**

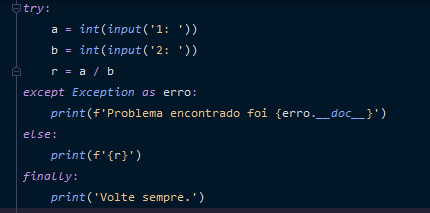
**OBS:else/finally opcional.**



**Prática no Python de identificação de erro.**



**Prática no Python de identificação de erro, agora adicionando a Exceção a uma variável e mostrando a documentação dessa Exception**



**Agora identificando que tipo de exception é, e mostrando uma mensagem conforme sua classe,**

**ValueErro/TypeError problema com o valor inserido ou exemplo letra ou caracteres especiais no lugar de numero,.**

**ZeroDivisionError problema com valores dividos por 0**

**KeyboardInterrupt se o usuário apertar CTRL+C ou DELETE encerra a aplicação do Try**

