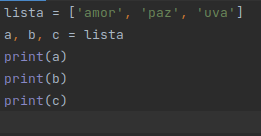
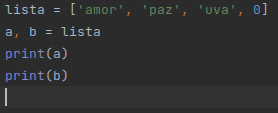
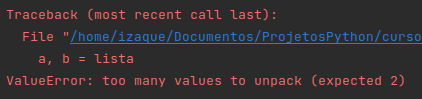
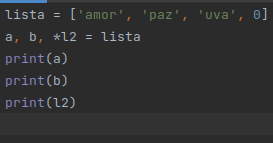
**43 – Desempacotamento**

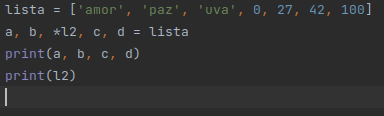
Daremos o exemplo de desempacotamento com Listas, mas pode ter o desempacotamento com qualquer iterable object.

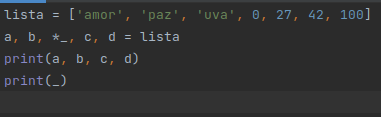
 

Para contornar isso, podemos empacotar os outros valores,colocando-os em uma lista, para isso use o asterisco \* antes do nome dela:

Se não formos usar a lista é convencionado nomeá-la como \_:

**Desempacotamento no for in**

Sabemos que a sintaxe do for é:

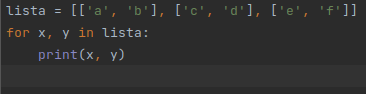
**for <var(s)>** **in <iterable>:**

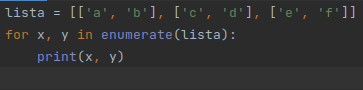
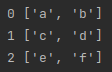
**<block>**

Colocamos um (s) no <var(s)> porque podemos fazer um desempacotamento no for.

Sabemos que o for in funciona da seguinte forma: Ele pega o primeiro item do iterable e atribui a <var(s)> e realiza o <block>, depois pega o segundo do item do iterable atribui a <var(s)> e realiza o <block> ... até o ultimo item do iterable.

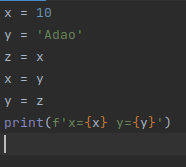
Repare, que se <var(s)> ter mais de uma variável e cada item do <iterable> ser um iterable com o numero de itens iguais ao numero de variaveis do <var(s)> então será realizado um desempacotamento.

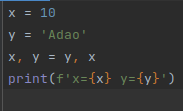
 

**Desempacotamento utilizando variaveis no lado direito:**

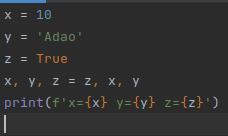
Na maioria das linguagens quando queremos trocar os valores de variaveis, devemos criar uma variável auxiliar, dessa forma:

Em python, há uma forma mais fácil, utilizando desempacotamento:

Analogamente:

Também podemos desempacotar objetos iteráveis nos parâmetros de uma função, para isso basta acrescentar o asterisco \* antes desse objeto:

