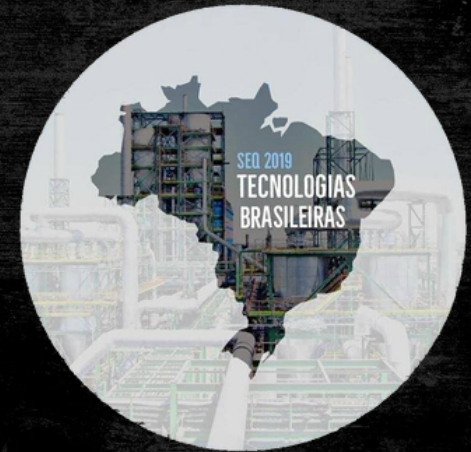




86ª EDIÇÃO

**SEQ
UFRJ**

26 a 30 de agosto



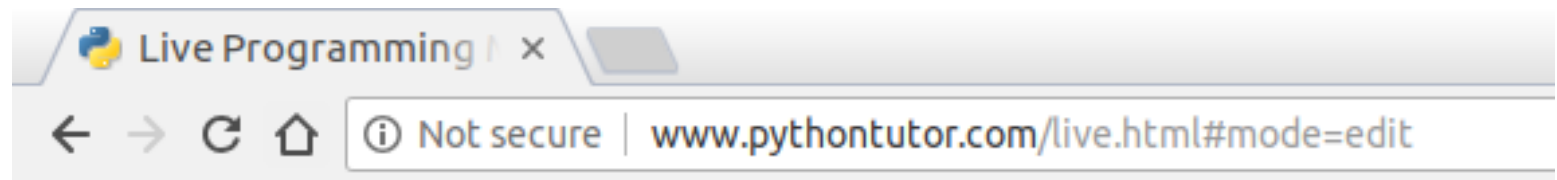
Iuri Segtovich

Parte 2: Lógica e Sintaxe

Funções (def, return, tuple, dict)

python tutor

[www.pythontutor.com/
live.html#mode=edit](http://www.pythontutor.com/live.html#mode=edit)



Write code in Python 3.6

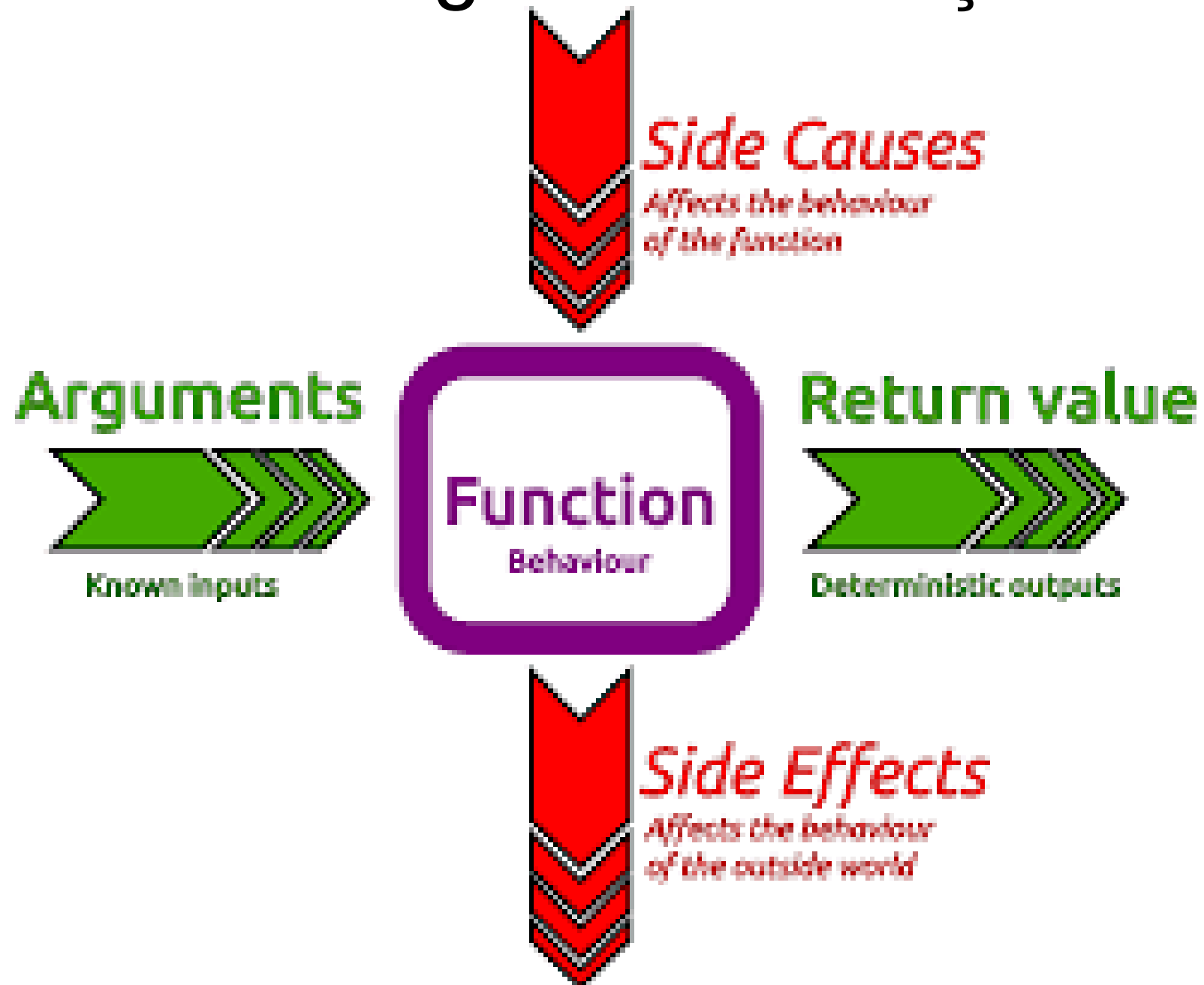
(drag lower right corner to resize code editor)

```
1 |
```

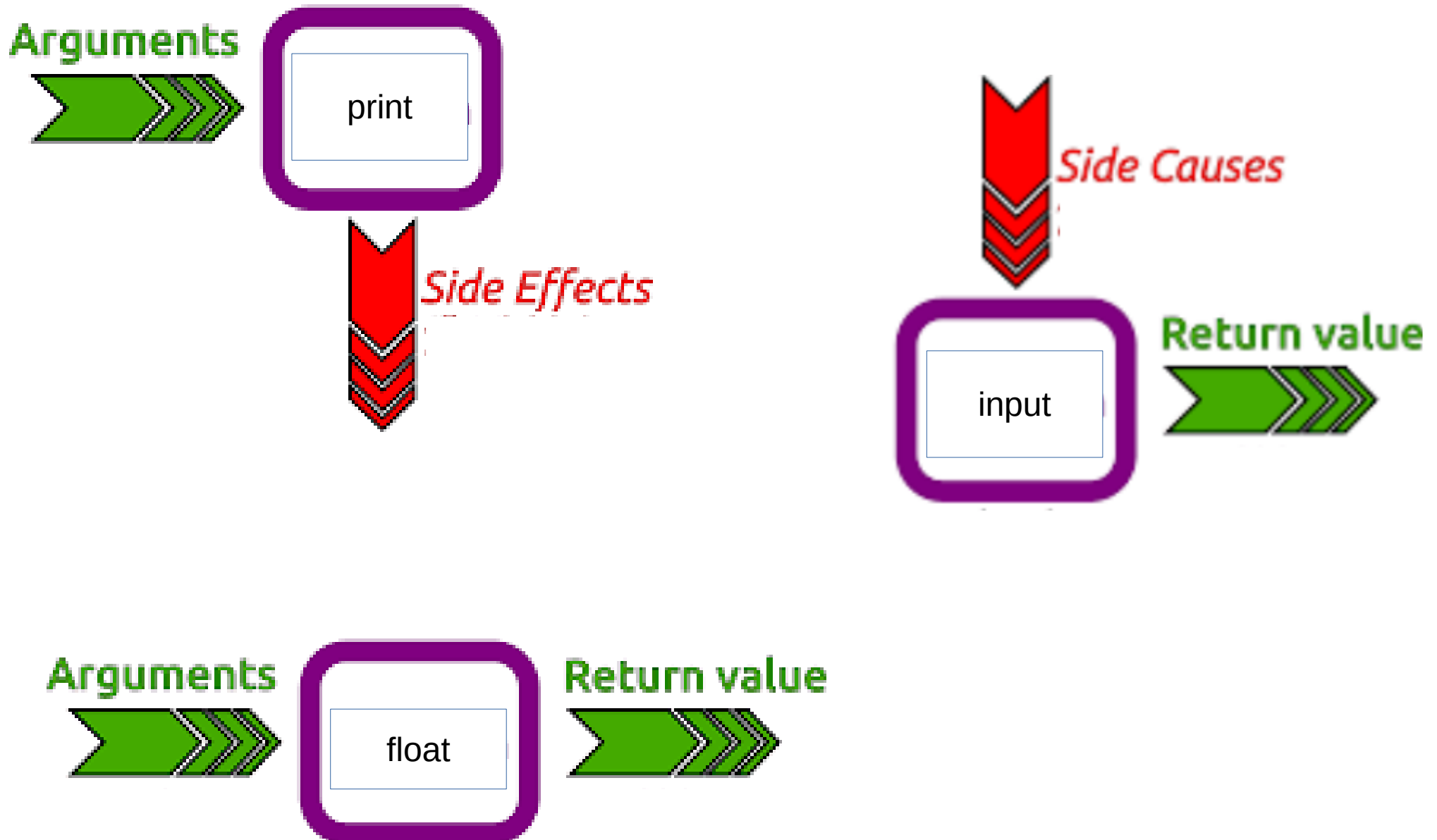
→ line that has just executed

→ next line to execute

O caso geral das funções

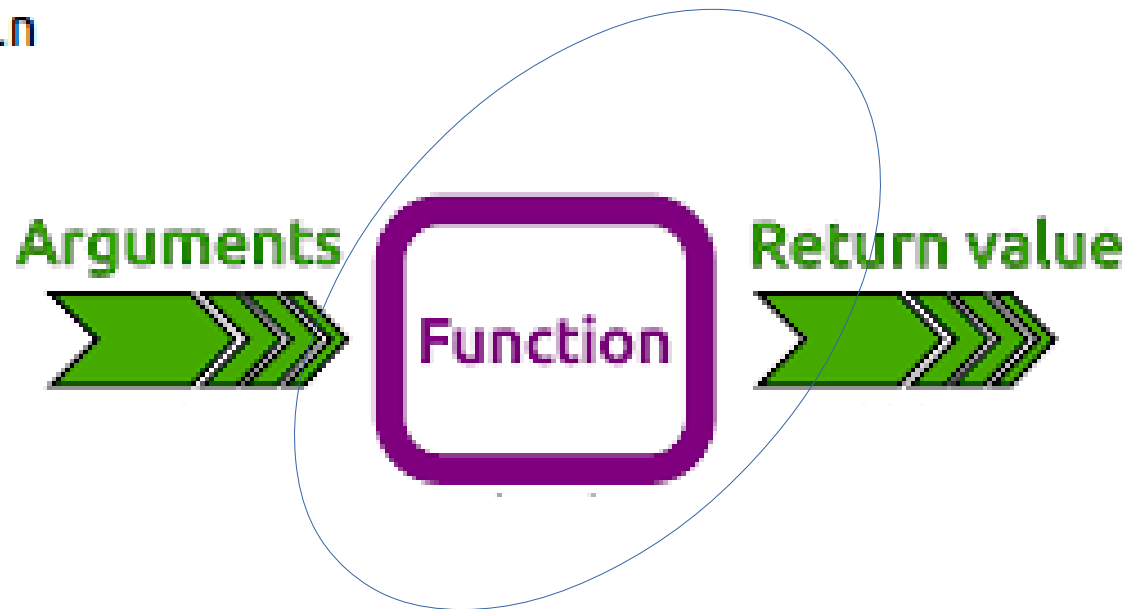


Exemplos



def função / return

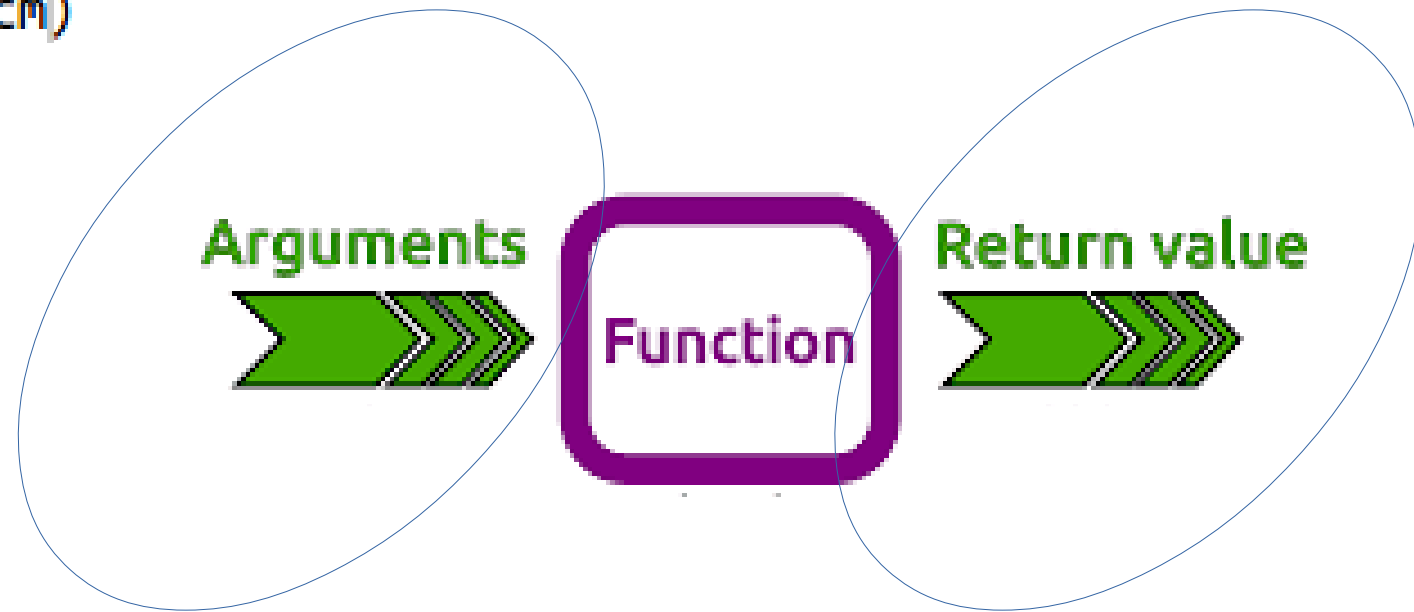
```
1 def converte_m_para_cm(x_in):  
2     x_out = 100.*x_in  
3     return x_out  
4
```



- Definição:
 - def, Nome, Parênteses, Argumentos DUMMY
 - dois pontos
 - indentação
 - return

chamada

```
5 medida_m=5.  
6 medida_cm = converte_m_para_cm(medida_m)  
→ 7 print(medida_cm)
```



- Chamada:
 - Variável resposta, sinal de igual (BINDING)
 - Nome da função, Parênteses, Argumentos ACTUAL

Exemplo: função de conversão de unidades de temperatura de F para C e vice-versa.

$$F = 1,8C + 32$$

Side causes



Side Causes

Arguments



Function
Behaviour

Return value



```
1 R = 8.314
```

```
2  
3 def fCtoK(T):  
4     return T+273.15
```

```
5  
6 def fVgi(TC,P):  
7     TK = fCtoK(TC)  
8     return R*TK/P
```

```
9  
10 T1 = 0
```

```
11 P1 = 1e5
```

```
12  
13 V1 = fVgi(T1,P1)
```

```
14
```


argumentos

- Chamada por argumentos posicionais

```
10 T1 = 0
11 P1 = 1e5
12
13 V1 = fVgi(T1,P1)
```

- Chamada por argumentos palavra-chave

```
14
15 T2 = 25
16 P2 = 1e6
```

→ 18 V2 = fVgi(TC=T2,P=P2)

A chamada passa o objeto de nome T1/T2 para ser usado dentro da função
O nome *dummy* TC faz referência a esse objeto enquanto estivermos dentro da função

Argumentos OPCIONAIS

- Valor DEFAULT

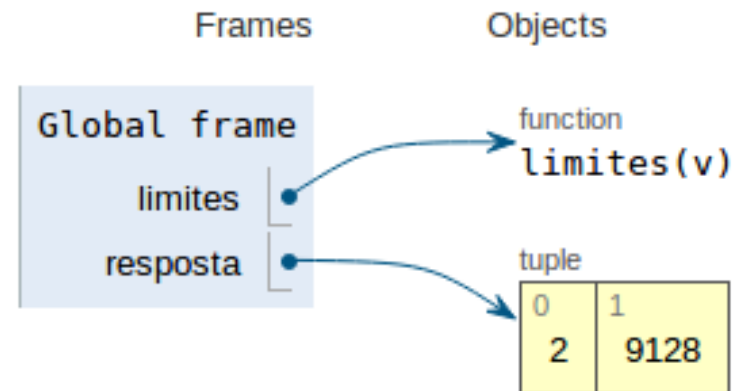
- Chamada por argumentos palavra-chave

```
R=8.314
import math as m
def Turbina(P1,T1,P2,ETA=1.):
    T2=T1*m.exp(R*m.log(P2/P1)/Cp)
    Wi=Cp*(T2-T1)
    Wr=Wi*ETA
    return T2, Wr

Cp=1000
print( Turbina(100,300,1) )
print( Turbina(100,300,1,ETA=0.8) )
```

Resultados estruturados (tuple)

```
1 def limites(v):
2     inf = min(v)
3     sup = max(v)
4     return inf, sup
5
6 → resposta = limites([9128,345,56,2,678])
```

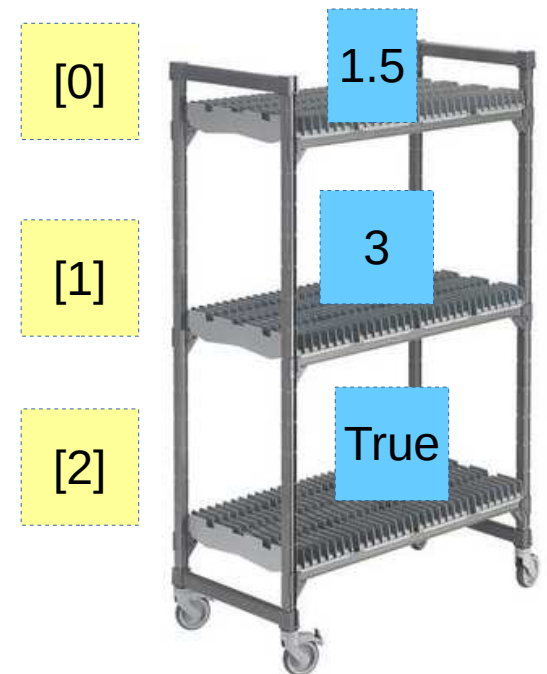


```
print( resposta[0] )
print( resposta[1] )
```

Indexação igual às listas

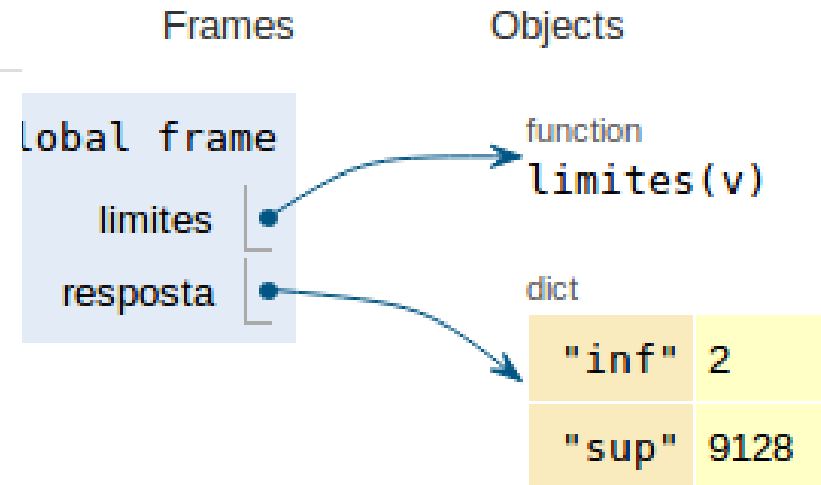
```
a,b = limites([9128,345,56,2,678])
```

Desempacota automaticamente



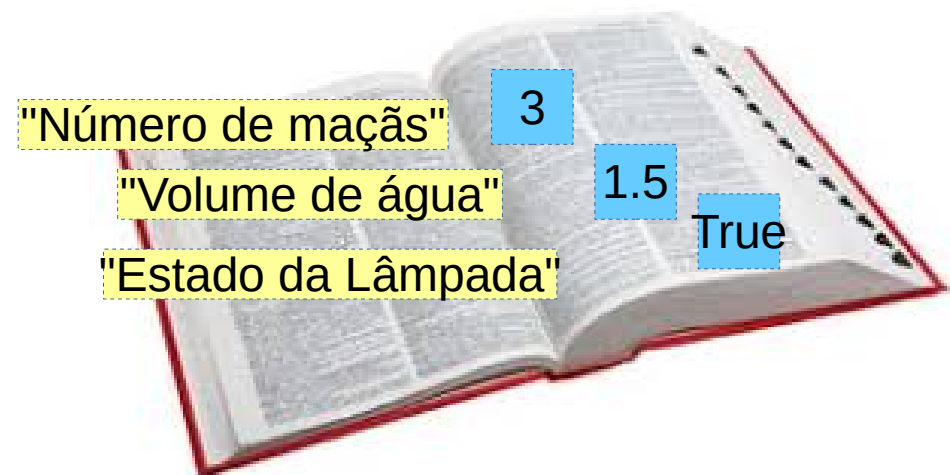
Resultados estruturados (dict)

```
1 def limites(v):
2     inf = min(v)
3     sup = max(v)
4     return {"inf":inf, "sup":sup}
5
6 → resposta = limites([9128,345,56,2,678])
```

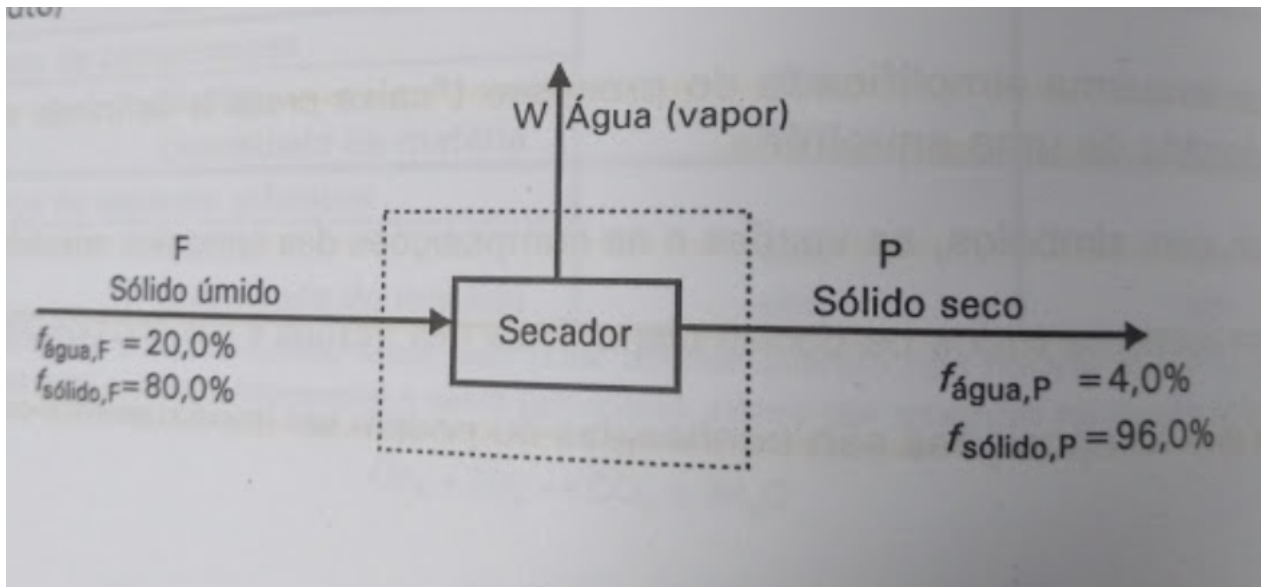


```
print( resposta["inf"] )
print( resposta["sup"] )
```

Indexação por palavra-chave



Exercício



$$P = F \frac{x_s^f}{x_s^p}$$

$$W = Fx_w^f - Px_w^p$$

$$R\% = 100 \frac{W}{Fx_w^f}$$

Referências principais

[https://www.tutorialspoint.com/
python3/
python_basic_syntax.htm](https://www.tutorialspoint.com/python3/python_basic_syntax.htm)

[https://stackoverflow.com/
search](https://stackoverflow.com/search)

perguntas

