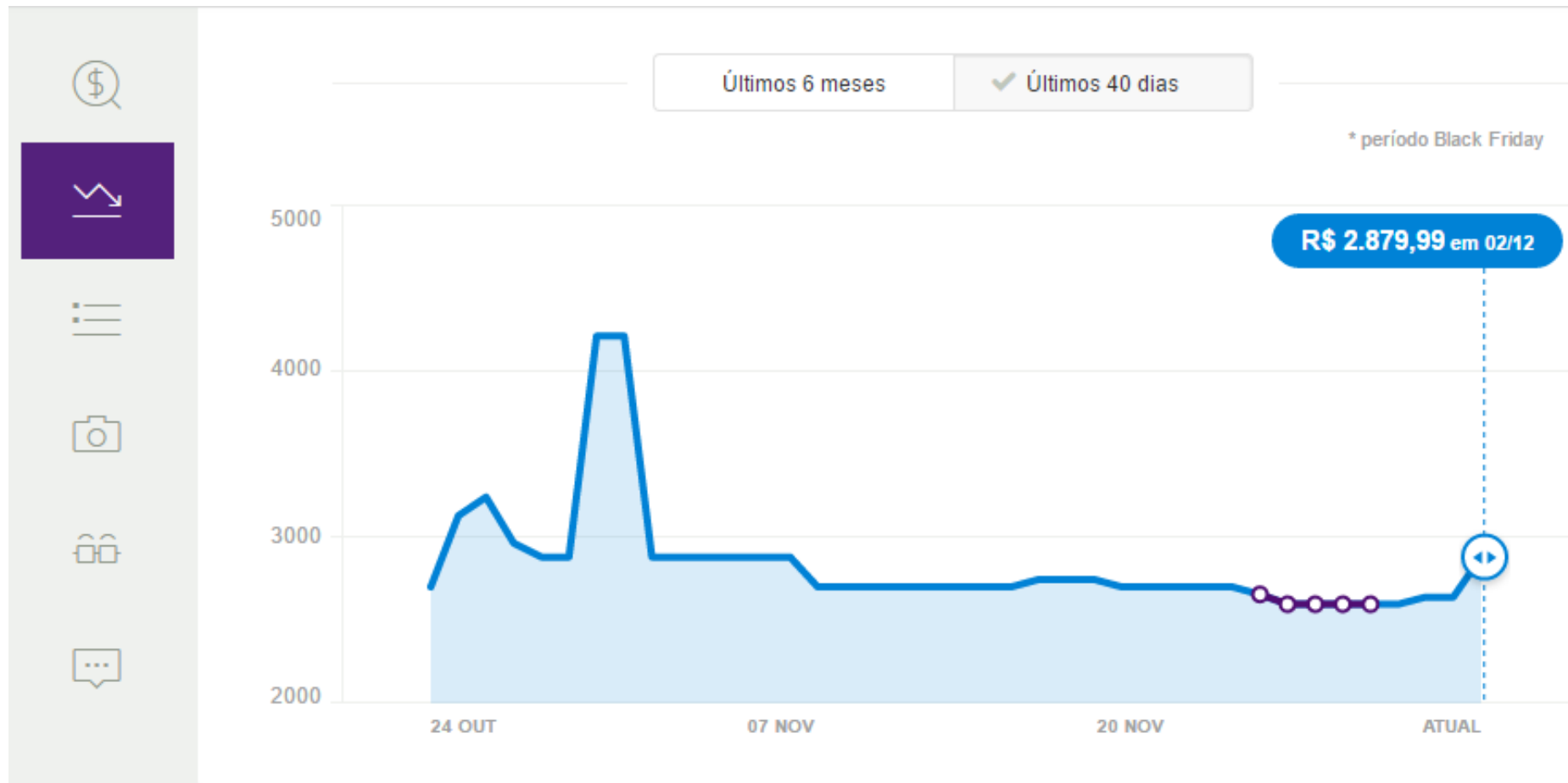


# Listas em Python

# Contextualização #1

Histórico de preços de um produto principal



# Histórico de Preços



Passos de uma possível solução:

- 1) Obter a sequência de preços a exibir
- 2) Encontrar o maior e o menor valor para desenhar os eixos X e Y
- 3) Desenhar os eixos
- 4) Desenhar as retas a partir dos pontos do histórico de preços

# Contextualização #2

## Corrigir prova de múltipla escolha:

### Questão 1

Os sobrinhos do personagem da Disney chamado de Pato Donald são:

- a) Huguinho, Zézinho e Paulinho.
- b) Joãozinho, Zézinho, Huguinho e Paulinho.
- c) Juninho, Zézinho e Huguinho.
- d) Luizinho, Huguinho e Zézinho.
- e) Patinho, Patola e Patinhozinho.

### Questão 2

A metade da metade de 12 é igual a:

- a) 6.
- b) 4.
- c) 2.
- d) 3.
- e) 12.

### Questão 3

Nome do rio onde fica a Cachoeira de Paulo Afonso.

- a) São Francisco.
- b) Paraná.
- c) Tietê.
- d) Paraíba do Sul.
- e) Amazonas.

### Questão 4

Uma dessas afirmações está ERRADA.

- a) A árvore símbolo que deu nome ao nosso país é o Pau Brasil.
- b) O Jacaré é um réptil.
- c) O Mamute era um anfíbio.
- d) A lua é o satélite natural da terra.
- e) A piranha é um peixe de água doce.

### Questão 5

Ao entrar numa sala, João contou 4 pessoas, incluindo ele. Todos estavam calçados. Sem contar com ele quantos sapatos havia na sala?

- a) 4.
- b) 6.
- c) 8.
- d) 16
- e) 10.

### Questão 6

Complete a frase a seguir: "De Grão em grão..."

- a) A galinha fica mais gorda.
- b) O galo fica maior.
- c) O pintinho vai ficando gordo.
- d) O Depósito fica cheio.
- e) A galinha enche o papo.

### Questão 7

O animal já extinto chamado DODÔ, era:

- a) Um réptil.
- b) Um dinossauro.
- c) Um pássaro.
- d) Um peixe que media até 3 metros de comprimento.
- e) Uma serpente marinha que se alimentava exclusivamente de algas.

### Questão 8

A palavra MARAJÁ quer dizer:

- a) Pessoa muito rica.
- b) Pessoa que vive sem fazer nada.
- c) Pessoa que ganha dinheiro sem trabalhar.
- d) Título de nobreza indiano.
- e) Espécie de gato silvestre selvagem.

### Questão 9

A palavra MARACUTAIA tem quantas vogais:

- a) 4.
- b) 5.
- c) 6.
- d) 7.
- e) 8.

### Questão 10

O Carnaval é:

- a) 38 dias antes da Páscoa.
- b) 39 dias antes da Páscoa
- c) 40 dias antes da Páscoa
- d) 42 dias antes da Páscoa.
- e) 45 dias antes da Páscoa.

# Correção automática de uma prova

Questão 1) Escreva ao lado o resultado da expressão:

1.  $3 \cdot 4 - 6 / (4 + 2)$  = .....

2.  $6 + 4 / 2 / 2$  = .....

3.  $(-5 + 4) \cdot (7 \cdot -3)$  = .....

Questão 2) Escreva os anagramas que podem ser formados a partir das letras da palavra AMO que iniciam com M.

.....

Questão 3: Dadas as funções reais  $f(x) = 2x - 6$  e  $g(x) = ax + b$ , se  $f[g(x)] = 12x + 8$ , o valor de  $a + b$  é.....

Questão 4) Considere as funções  $f(x) = 2x + 1$  e  $g(x) = x^2 - 1$ . As raízes da equação  $f(g(x)) = 0$  são:

a) inteiras      b) negativas      c) racionais      d) inversas      e) opostas

Questão 5) Dada a função  $f(x) = (x-1)^x / (-x^x)$ , calcule  $f(0)$ . .....

Questão 6) a função  $f(x): 2x^2 + 7x$  cruza o eixo dos y? .....

Questão 7) Os pontos  $(0, -60)$ ,  $(2, -42)$  e  $(7, 108)$  pertencem à parábola  $y = ax^2 + bx + c$  para quais valores de  $a$ ,  $b$  e  $c$ ?

$a =$  .....       $b =$  .....       $c =$  .....

Questão 8) Escreva o produto cartesiano dos conjuntos  $A = \{1, 2\}$  e  $B = \{4, 8, 9\}$

.....

Questão 9) Uma bola será retirada de uma sacola contendo 5 bolas verdes e 7 bolas amarelas. Qual a probabilidade desta bola ser verde? .....

Questão 10) Na fila do caixa de uma padaria estão três pessoas. De quantas maneiras elas podem estar posicionadas nesta fila? .....

# Corrigir provas - Principal

Questão 1  
Os sobrinhos do personagem da Disney chamado de Pato Donald são:  
a) Huguinho, Zézinho e Paulinho.  
b) Joãozinho, Zézinho, Huguinho e Paulinho.  
c) Juninho, Zézinho e Huguinho.  
d) Luizinho, Huguinho e Zézinho.  
e) Patinho, Patola e Patinhozinho.

Questão 2  
A metade da metade de 12 é igual a:  
a) 6.  
b) 4.  
c) 2.  
d) 3.  
e) 12.

Questão 3  
Nome do rio onde fica a Cachoeira de Paulo Afonso.  
a) São Francisco.  
b) Paraná.  
c) Tietê.  
d) Paraíba do Sul.  
e) Amazonas.

Questão 4  
Uma dessas afirmações está ERRADA.  
a) A árvore símbolo que deu nome ao nosso país é o Pau Brasil.  
b) O Jacaré é um réptil.  
c) O Mamute era um anfíbio.  
d) A lua é o satélite natural da terra.  
e) A piranha é um peixe de água doce.

Questão 5  
Ao entrar numa sala, João contou 4 pessoas, incluindo ele. Todos estavam calçados. Sem contar com ele quantos sapatos havia na sala?  
a) 4.  
b) 6.  
c) 8.  
d) 16  
e) 10.

Questão 6  
Complete a frase a seguir: "De Grão em grão..."  
a) A galinha fica mais gorda.  
b) O galo fica maior.  
c) O pintinho vai ficando gordo.  
d) O Depósito fica cheio.  
e) A galinha enche o papo.

Questão 7  
O animal já extinto chamado DODÔ, era:  
a) Um réptil.  
b) Um dinossauro.  
c) Um pássaro.  
d) Um peixe que media até 3 metros de comprimento.  
e) Uma serpente marinha que se alimentava exclusivamente de algas.

Questão 8  
A palavra MARAJÁ quer dizer:  
a) Pessoa muito rica.  
b) Pessoa que vive sem fazer nada.  
c) Pessoa que ganha dinheiro sem trabalhar.  
d) Título de nobreza indiano.  
e) Espécie de gato silvestre selvagem.

Questão 9  
A palavra MARACUTAIA tem quantas vogais:  
a) 4.  
b) 5.  
c) 6.  
d) 7.  
e) 8.

Questão 10  
O Carnaval é:  
a) 38 dias antes da Páscoa.  
b) 39 dias antes da Páscoa.  
c) 40 dias antes da Páscoa.  
d) 42 dias antes da Páscoa.  
e) 45 dias antes da Páscoa.

## Passos de uma possível solução:










1) Obter o gabarito ( sequência de respostas corretas)

2) Corrigir provas:

Obter respostas, contando acertos de acordo com o gabarito

# Contextualização #3

Selecione a cor:

								
---	---	---	---	---	---	---	---	---

Escolha o tamanho da sequência: 3 4 5 6

1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Verificar
2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Verificar
3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Verificar
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Verificar
5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Verificar
6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Verificar
7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Verificar
8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Verificar
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Verificar
10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Verificar

O objetivo do Mastermind é descobrir a sequência correta de cores.

O tabuleiro do jogo: 10 linhas

Cada linha: uma chance

Para jogar: clicar sobre a cor e posição

Respostas em cada linha:

- Pino preto: acertou a cor e a posição de uma das cores da combinação.
- Pino branco: acertou a cor, mas não a posição de uma das cores da combinação.

# Jogo do *MasterMind* (ou Senha)

Escolha a quantidade de cores:				
<input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 <input checked="" type="radio"/> 8 <input type="radio"/> 9 <input type="radio"/> 10 <input type="radio"/> 11 <input type="radio"/> 12				
Selecione a cor:				
<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>				

Escolha o tamanho da sequência: <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6					
1				<div><div></div><div></div><div></div></div>	Verificar
2				<div><div>o</div><div>o</div><div></div></div>	Verificar
3				<div><div>•</div><div>•</div><div></div></div>	Verificar
4				<div><div>•</div><div></div><div></div></div>	Verificar
5				<div><div>•</div><div>•</div><div>•</div></div>	Verificar
6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<div><div></div><div></div><div></div></div>	Verificar
7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<div><div></div><div></div><div></div></div>	Verificar
8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<div><div></div><div></div><div></div></div>	Verificar
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<div><div></div><div></div><div></div></div>	Verificar
10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<div><div></div><div></div><div></div></div>	Verificar



# Contextualização 3: MasterMind - Principal

Selecione a cor:									
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Escolha o tamanho da sequência: 3 4 5 6									
1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Combinação Secreta									
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Passos de uma possível solução:

- 1) Criar um sequência de cores a adivinhar (não visível)
- 2) venceu= Jogo(sequência de cores a adivinhar , chances)
- 3) Mostrar sequência a adivinhar
- 4) Se venceu == TRUE

Exibir mensagem de Parabéns!!!

senão

Exibir mensagem de Consolação!!!

# Contextualização #4

## Sequências numéricas:

Complete os espaços com os números que faltam. Preste bastante atenção para descobrir a lógica de cada fase!



# Sequências numéricas



Passos de uma possível solução:

- 1) Criar um sequência de números com uma lei de formação
- 2) Esconder um (ou mais) número(s) da sequência, incluindo-os em outra sequência.
- 3) venceu= Jogo(sequência de números, sequência escondidos)
- 4) Se venceu

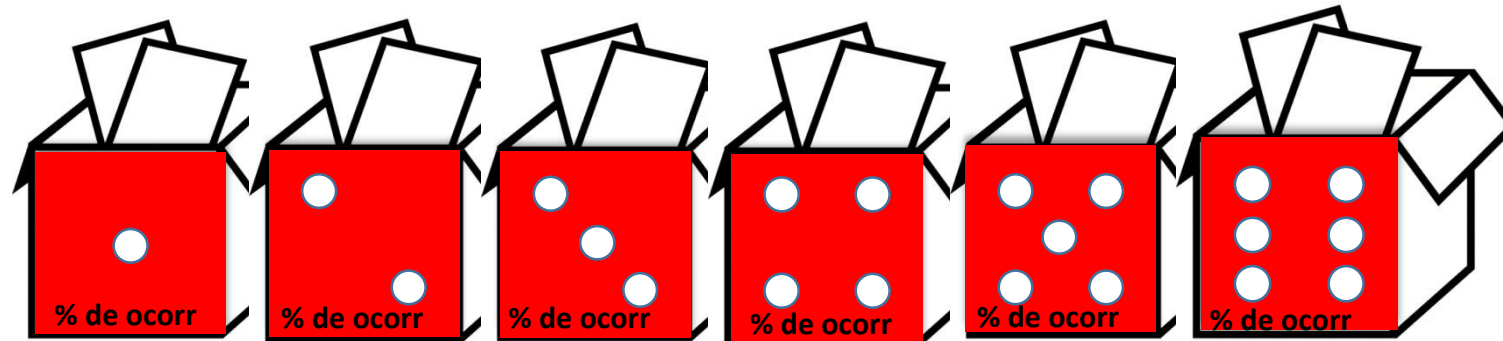
Exibir mensagem de Parabéns!!!

senão

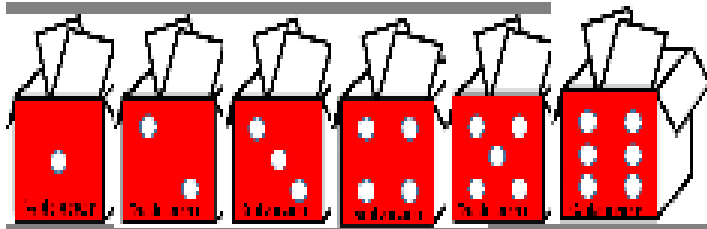
Exibir mensagem de Consolação!!!

# Contextualização #5

Verificar se um dado está viciado: campeão por modalidade esportiva na InterEng



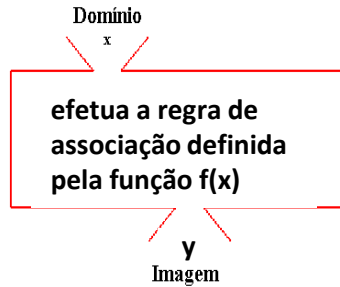
# Dados viciados?



Passos de uma possível solução:

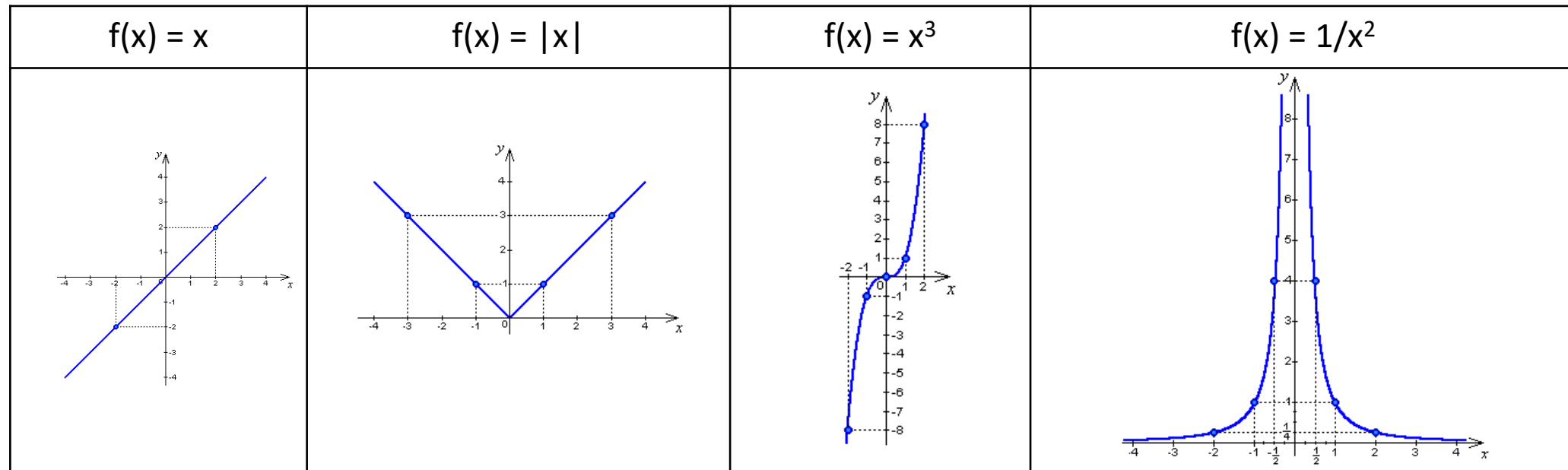
- 1) Zerar contadores de ocorrência (coleção de contadores)
- 2) Para cada lançamento registrado:  
Somar 1 ao contador do número ocorrido no lançamento
- 3) Calcular percentuais
- 4) Verificar se há discrepância

# Contextualização #6

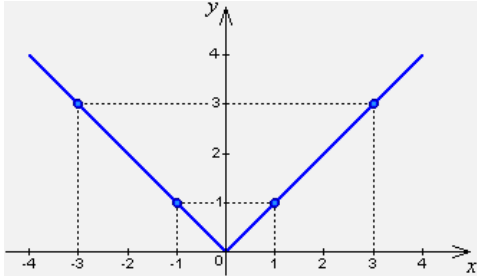


Traçar gráfico de uma função:

O gráfico cartesiano de uma função  $f$  é o conjunto de todos os pontos  $(x, f(x))$  do plano com  $x$  variando no domínio de  $f$



# Traçar gráfico de uma função



Passos de uma possível solução:

- 1) Selecionar uma sequência de valores para  $x$
- 2) Desenhar os eixos cartesianos  $x$  e  $y$ :
  - 1) Desenhar duas linhas ortogonais para os eixos cartesianos ( $x$  e  $y$ )
  - 2) Marcar estas linhas com pontos igualmente espaçados entre si
- 3) Para cada valor  $x$  desta sequência:
  - 1) Calcular  $f(x)$
  - 2) Marcar o ponto  $x$  no eixo  $X$  e o ponto  $f(x)$  no eixo  $Y$
  - 3) Desenhar o ponto no plano onde ocorre a interseção das retas paralelas aos eixos que passa nos pontos marcados

# Como resolver?

Todos estes exemplos manipulam (pelo menos) uma sequência de valores, não necessariamente do mesmo tipo.

Em alguns problemas é preciso acessar os elementos na ordem que aparecem na sequência, em outros, não .



- ✓ Como criar uma variável capaz de guardar uma sequência de valores e com este comportamento?
- ✓ Como saber quantos valores têm na sequência?
- ✓ Como exibir os valores da sequência?
- ✓ Como acessar/alterar um ou mais elementos da sequência?
- ✓ Como adicionar /eliminar um ou mais elementos à sequência?



LISTA: uma sequência ou coleção ordenada de valores de qualquer tipo.

Valores de uma lista (*elementos* ou *ítems*):

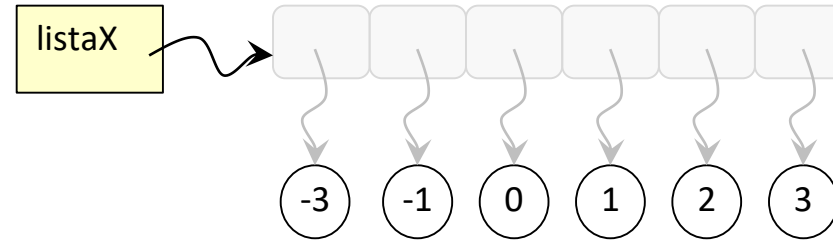
- podem ser de tipos diferentes e até mesmo outras listas
- são delimitados por colchetes ( `[]` ) e separados por vírgulas

**Exemplo: Lista com o gabarito da 2ª prova:**

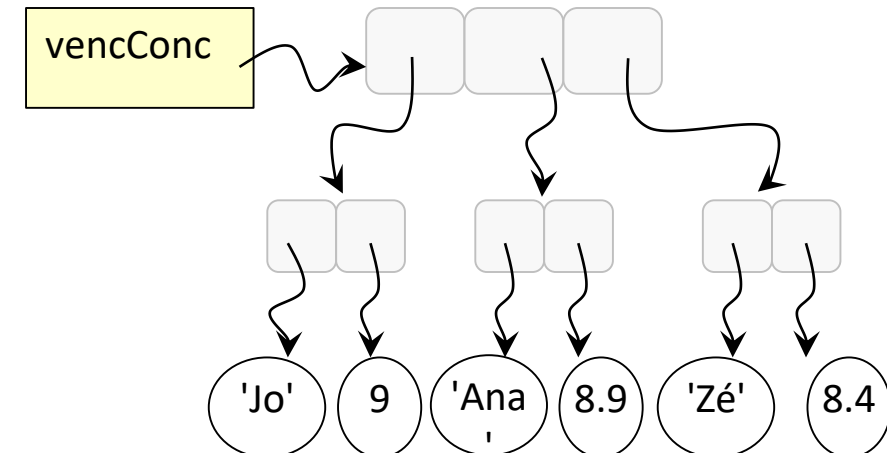
```
GabProva2 = [ [11.0,7.0,21] , ['MAO','MOA'] , 13 , 'c' , [1,1,0.25] , 'Não' , [3,3,-60] , [[1,4] , [1,8] , [1,9] ,  
[2,4] , [2,8] , [2,9]] , 5/12 , 6 ]
```

# Lista em Python: representação

listaX = [ -3,-1,0,1,2,3 ]



vencConc = [ ['Jo',9] , ['Ana',8.9] , ['Zé',8.4] ]

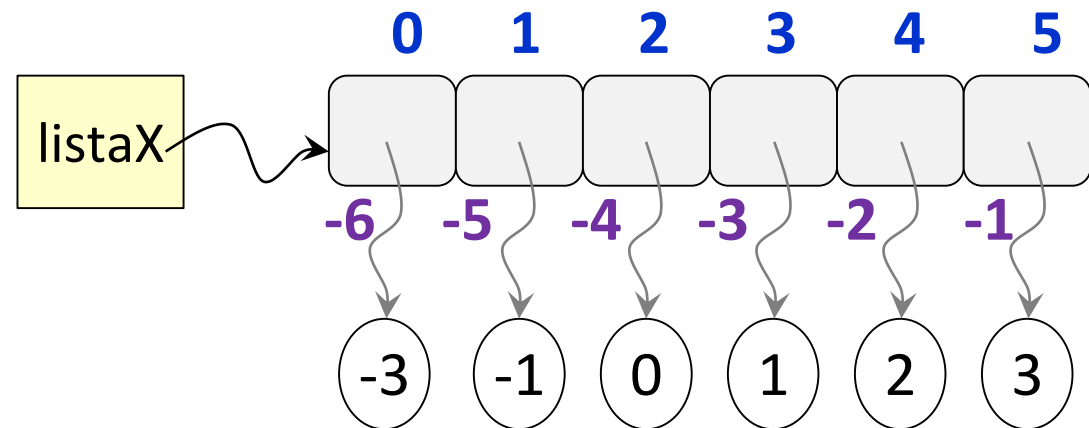


# Tipo Lista em Python - Índices

Os itens de uma lista são identificados por **índices**, que variam de:

- 0 até o comprimento da lista-1 (posição a partir do início)
- -1 até -comprimento da lista (posição a partir do final)

Exemplo: `listaX = [-3,-1,0,1,2,3]`



# Tipo Lista em Python - Dinâmica

As listas têm tamanho variável. Crescem ou diminuem quando elementos são inseridos ou retirados



```
arroz  
detergente  
abobrinha  
limão  
escova de dente  
farinha de trigo  
leite  
óleo  
queijo ralado  
rúcula  
alface
```



**São mutáveis**, ao contrário de strings. A qualquer momento, um item:

- ✓ pode ser incluído na lista
- ✓ pode ser removido
- ✓ pode ser alterado
- ✓ pode ser consultado

# Criando uma lista por enumeração

Lista Vazia:

```
nomeVariável Lista = [ ]  
ou  
nomeVariável Lista = list()
```

Lista com elementos enumerados :

```
nomeVariável Lista = [ el1,el2,...,eln ]  
ou  
nomeVariável Lista = list(el1,el2,...,eln)
```

# Criando uma lista por enumeração

## Exemplos:

```
amigos = [ 'João', 'Pedro', 'Ana' ]  
megaSena = [ 1,17,22,24,25,36 ]  
vazia = []  
mista= [ 'Pedro', 9.0, 15*2, [1,2, 'carlos'] ]  
listas = [ amigos, megaSena ]
```

**Lista  
aninhada  
ou sublista**

# Comprimento de uma lista

**`len( varLista )`**

Retorna o número de elementos de uma lista.

Uma lista aninhada (*sublista* ou *lista dentro da lista*) é contada como MAIS UM elemento da lista que a contém.

## Exemplos:

```
amigos = [ 'João', 'Pedro', 'Ana' ]  
listas = [ amigos, megaSena ]  
vazia = []  
megaSena = [ 1, 17, 22, 24, 25, 36 ]  
mista = [ 'Pedro', 9.0, 15*2, [1, 2, 'carlos'] ]
```

```
len(amigos)      → 3  
len(listas)      → 2  
len(vazia)       → 0  
len(megaSena)    → 6  
len(mista)       → 4
```

# Exibindo uma variável do tipo lista

```
print( varLista)
```

## Exemplos:

```
amigos = [ 'João', 'Pedro', 'Ana' ]  
megaSena = [ 1,17,22,24,25,36 ]  
vazia = []  
mista = [ 'Pedro', 9.0, 15*2, [1,2,'carlos'] ]  
listas = [ amigos,megaSena ]
```

```
print('Amigos: ', amigos)  
print('MegaSena: ', megaSena)  
print('Com sublista:', mista)  
print('Com listas:', listas)  
print('Vazia:', vazia)
```

```
Amigos: ['João', 'Pedro', 'Ana']  
MegaSena: [1, 17, 22, 24, 25, 36]  
Com sublista: ['Pedro', 9.0, 30, [1, 2, 'carlos']]  
Com listas: [['João', 'Pedro', 'Ana'], [1, 17, 22, 24, 25, 36]]  
Vazia: []
```



# Acesso aos elementos: indexação

Pelo operador de indexação **[ ]**.

```
varLista[expressão inteira]
```

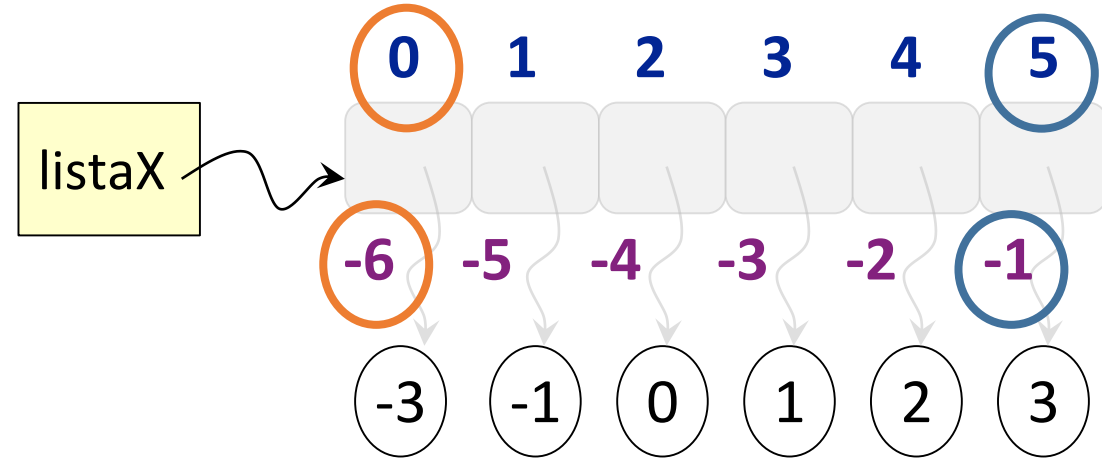
Expressão: resultando em um valor inteiro e determina o índice.

- ✓ Índices positivos ( 0 até comprimento da lista-1) indicam a posição do elemento na lista da esquerda para a direita.
- ✓ Índices negativos ( -1 até -comprimento da lista) indicam a posição do elemento da direita para a esquerda

Índices **fora do intervalo** provocam um **erro**

# Indexando elementos da lista (1/7)

listaX = [-3,-1,0,1,2,3]

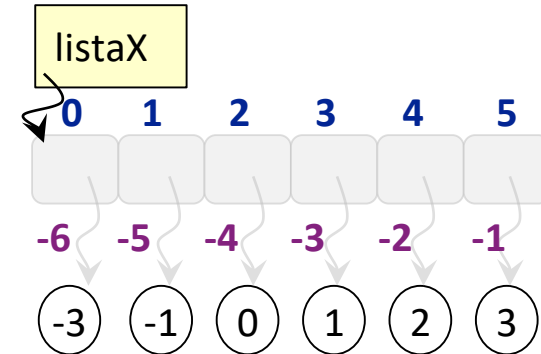


Qual o índice do primeiro elemento? 0

Qual o índice do último elemento? -1 ou  $\text{len}(\text{listaX})-1$

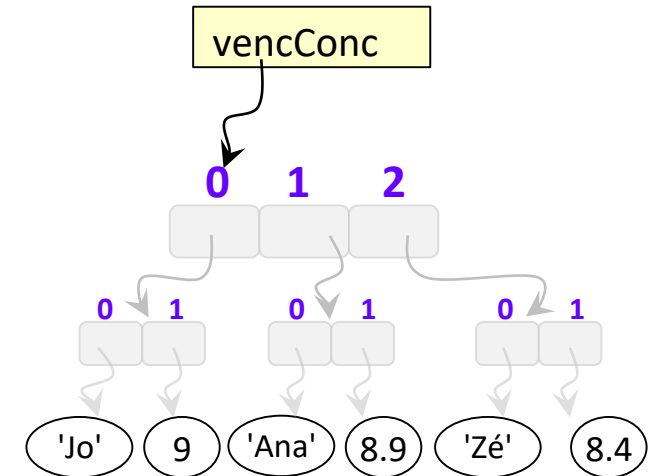
# Indexando elementos da lista (3/7)

**Exemplo 1:**      `listaX = [-3,-1,0,1,2,3]`



```
tam = len(listaX)
print(listaX[ 2 ])           → 0
print(listaX[ 9-8 ])        → -1
print(listaX[ -2 ])         → 2
print(listaX[ tam-1 ])      → 3
print(listaX[ -1 ])         → 3
print(listaX[ tam ])        → IndexError: list index out of range
```

**Exemplo 2:** `vencConc = [['Jo',9 ],['Ana',8.9],['Zé',8.4]]`



```
print (vencConc [ 0 ] )
```

→ ['Jo', 9]

```
print (vencConc [ -len (vencConc) ] )
```

→ ['Jo', 9]

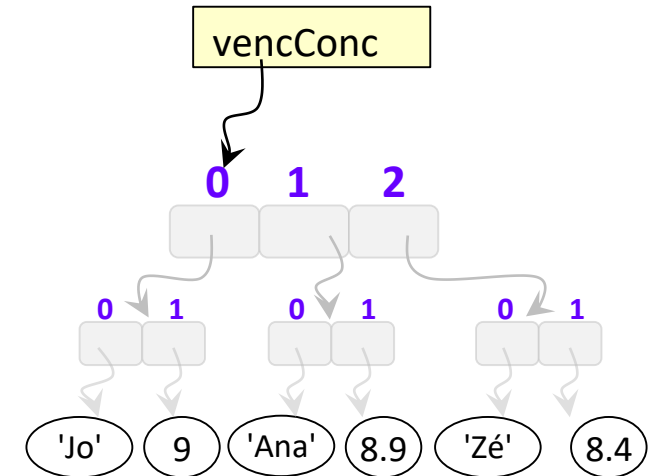
```
print (vencConc [ -4 ] )
```

→

**IndexError:**

**list index out of range**

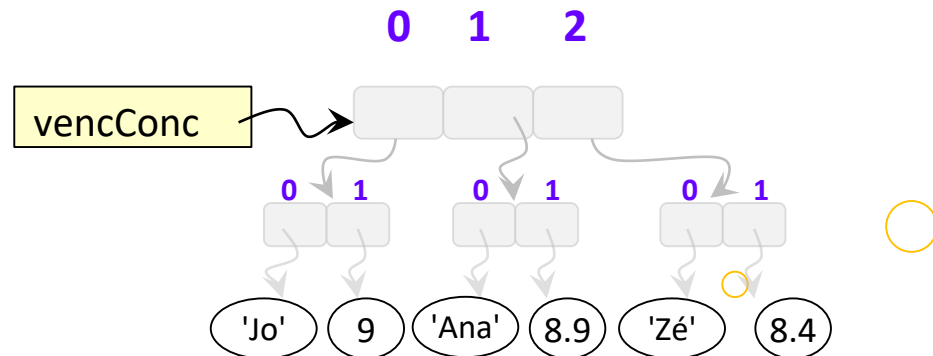
**Exemplo 2:** `vencConc = [['Jo',9 ],['Ana',8.9],['Zé',8.4]]`



```
print(vencConc[len(vencConc)-1]) → ['Zé', 8.4]
print(vencConc[-1]) → ['Zé', 8.4]
print(vencConc[9-10]) → ['Zé', 8.4]
```

# Indexando elementos da lista (6/7)

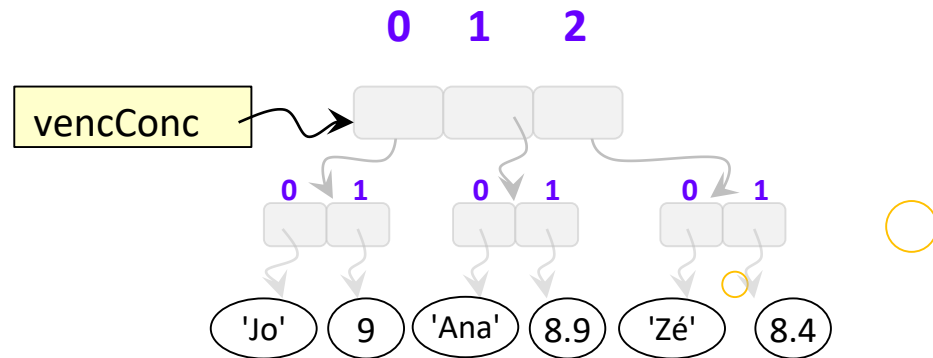
```
vencConc = [['Jo',9 ],['Ana',8.9],['Zé',8.4]]
```



Como acessar  
para exibir o  
nome do terceiro  
colocado?

# Indexando elementos da lista (6/7)

```
vencConc = [['Jo',9 ],['Ana',8.9],['Zé',8.4]]
```



Como acessar  
para exibir o  
nome do terceiro  
colocado?

```
print(vencConc[-1][0])
```

```
varLista[expressão inteira] = valor
```

Substitui o valor do elemento indexado pelo resultado da expressão

**Exemplos:**

```
listaX = [-3,-1,0,1,2,3]
```

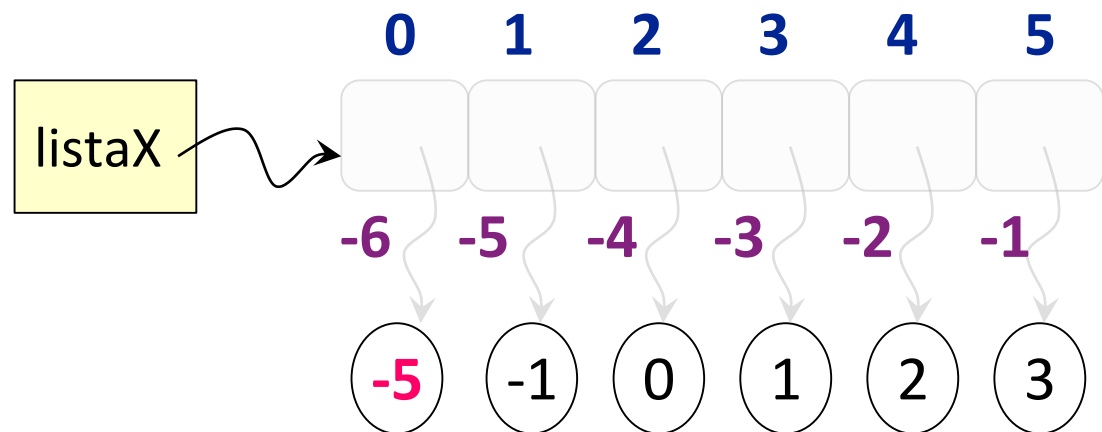
```
listaX[ 0 ] = -5
```

```
listaX[ 5 ] = [4,7]
```



Exemplo 1: `listaX[ 0 ] = -5`

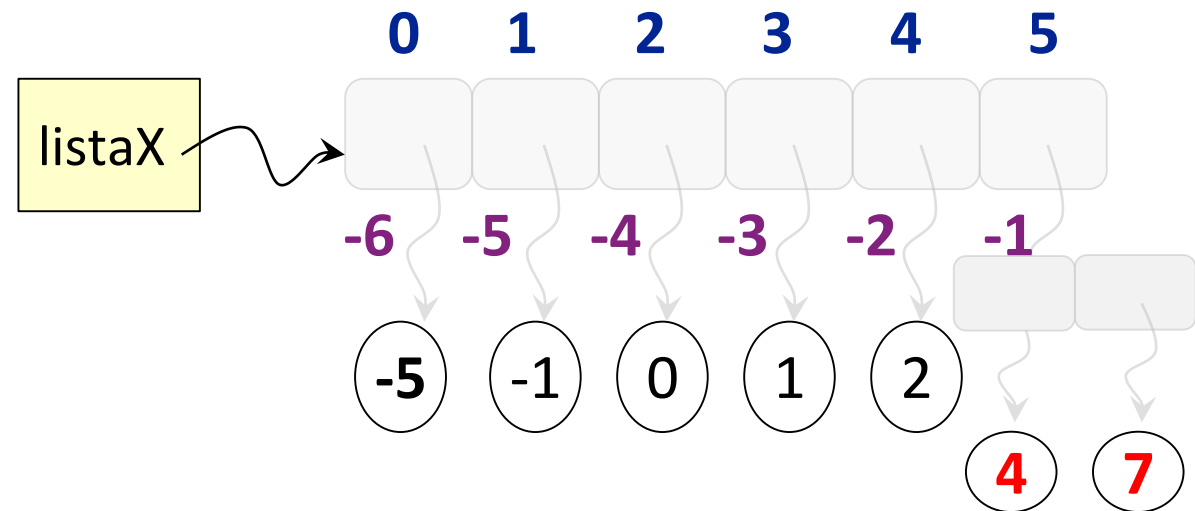
`[ -5, -1, 0, 1, 2, 3 ]`



Exemplo 2:

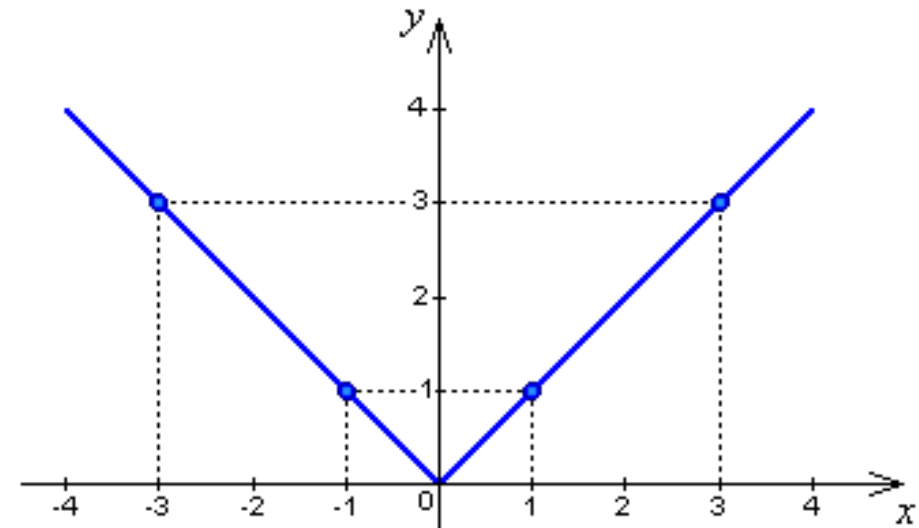
`listaX[ 5 ] = [4,7]`

`[ -5,-1, 0, 1, 2, [4,7] ]`



# Desenvolvendo o contexto #6

Traçar o gráfico de uma função:



***Idéia da solução:***

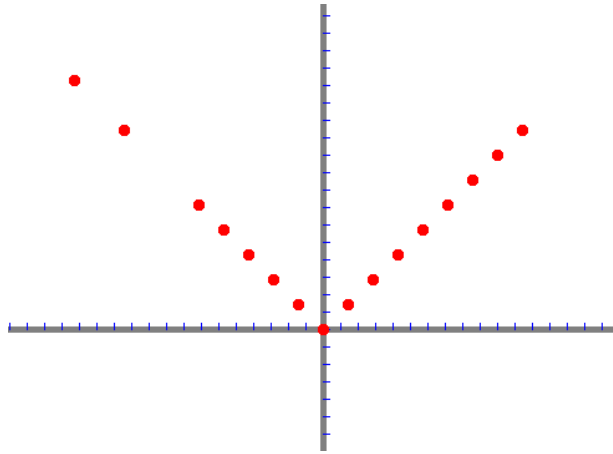
*Selecionar os valores de  $x$*

*Desenhar o eixo cartesiano*

*Para cada valor de  $x$ :*

*traçar ponto  $(x, y = f(x))$*

# Como plotar todos os pontos da lista?



Percorrer a lista, elemento a elemento , desenhando o ponto  $(x,y)$



Para cada  $x \in \text{lista}$   
 $y = f(x)$   
desenhar ponto  $(x,y)$

**Estrutura de Repetição!!!**

# Iterando sobre os elementos da lista

```
for elemento in lista:  
    CORPO
```

NOVO COMANDO *FOR*:

Acessa sequencialmente os elementos da lista, do 1º ao último, aplicando as instruções do corpo do bloco de comandos para cada elemento.

Exemplo:

```
lista= [1, 2, 3, 4]
```

```
for el in lista:  
    print(el)
```

# Comando FOR e Função range

```
for indice in range(len(lista)):
    CORPO
```

Gera os índices da lista sequencialmente, de 0 a n<sup>o</sup> de elementos-1

Exemplo:

```
lista= [1,2,3,4]
```

```
for i in range(len(lista)):
    print(lista[i])
```

# Comando FOR e Função enumerate

**for (indice,valor) in enumerate(lista):**  
**CORPO**

Acessa sequencialmente os elementos da lista, do 1º ao último, gerando os pares:  
índice da posição, valor da posição

Exemplo:

```
lista= [1,2,3,4]

for (i,val) in enumerate(lista):
    print("índice: %d - valor: %d",i,val)
```

# Exemplo

1) Considere as seguintes listas com as questões de uma prova e respectivo gabarito:

`IQuestoes=['3+5=....', '9*3=....', '2+3*4=....', '7-5+9=....', '4*6/3+2=....']`

`IGabarito=[8,27,14,11,10]`

- a) **Exibir o texto das questões**
- b) **Exibir o número da questão e o gabarito da questão**
- c) **Exibir cada questão seguida do gabarito**



```
lQuestoes=['3+5=....', '9*3=....', '2+3*4=....', '7-5+9=....', '4*6/3+2=....']  
lGabarito=[8,27,14,11,10]
```

```
#Exibir o texto das questões
```

```
for quest in lQuestoes:  
    print("%s"%quest)
```

```
#Exibir o número da questão e o gabarito da questão
```

```
for (ind,gab) in enumerate(lGabarito):  
    print("Questão %d - Gabarito:%d"%(ind+1,gab))
```

```
#Exibir cada questão seguida do gabarito:
```

```
for i in range(0,len(lGabarito)):  
    print("Questão %s - Gabarito:%d"%(lQuestoes[i],lGabarito[i]))
```

# Funções úteis

# Criando uma lista de um intervalo

Relembrando:

```
nomeVar Lista = list(range(inicio , fim , passo))
```

A função list(*sequência*) transforma a *sequência* em lista

A função range(): gera uma sequência de números, de *passo em passo*, no intervalo [*inicio*,*fim*)  
***início*** (opcional): 1º valor da sequência (default: 0)  
***fim***: sucessor do último valor da sequência  
***passo*** (opcional): intervalo entre os valores (default:1)

## Exemplos:

**print(lista)**

```
lista=list(range(3))           [ 0, 1, 2 ]
lista=list(range(-3,3))        [-3,-2,-1, 0, 1, 2 ]
lista=list( range(-3,3,2))      [-3,-1, 1 ]
lista=list( range(5,-2,-1))     [ 5, 4, 3, 2, 1, 0,-1 ]
lista=list( range(3,-3,1))      []
```

# $k$ elementos aleatórios em um intervalo

```
nomeVar Lista = random.sample(range(inicio , fim , passo), k)
```

Módulo *random*

A função `range()`: gera uma sequência de números

A função `sample` (*população*,  $k$ ), do módulo *random*, escolhe randomicamente  $k$  elementos (sem repetição) de uma sequência (*população*).

→ Retorna uma nova lista com os elementos escolhidos

## Exemplos:

**print(lista)**

```
lista=random.sample(range(0, 100), 4)    [3,61,0,46]  
lista=random.sample(range(-100,100), 4)  [22,-100,91,74]  
lista=random.sample(range(0,100,2), 4)   [40,12,8,82]
```

# $k$ elementos aleatórios de uma string

```
nomeVar Lista = random.sample(texto, k)
```

Convertendo um texto para lista:

```
nomeVar Lista = list(texto)
```

Unindo uma lista de strings em uma única string:

```
'símbolo cola'.join(lista)
```

```
texto='Oi,pessoal'
lsimb=list(texto)
print(lsimb)  #==>  ['O', 'i', ',', 'p', 'e', 's', 's', 'o', 'a', 'l']
textoRec=''.join(lsimb)
print(textoRec)  # ==>  Oi,pessoal
textoMod='-'.join(lsimb)
print(textoMod)  # ==>  O-i-,-p-e-s-s-o-a-l
```

EXERCÍCIO: Crie as listas l1 e l2

`l1 = [3, 7, 1, 90, 2]`

`l2 = [3, [98, 2, 1], 10]`

Em seguida, crie as funções abaixo e teste-as para as duas listas:

- a) Exiba os elementos da lista, um por linha
- b) Retorne a soma de todos os valores de uma lista

Ex `soma(l1) = 103`

`soma(l2) = 114`



# Uma solução: exhibe elementos

```
def exhibe(l):  
    for el in l:  
        print(el)  
    return
```

```
l1= [3, 7, 1, 90, 2]
```

```
l2=[3, [98, 2, 1], 10]
```

```
exib(l1)
```

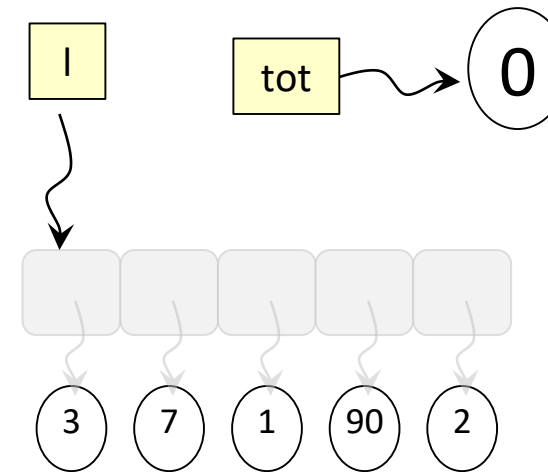
```
exib(l2)
```

# Soma valores de uma lista

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el    #tot=tot+el  
    return tot
```

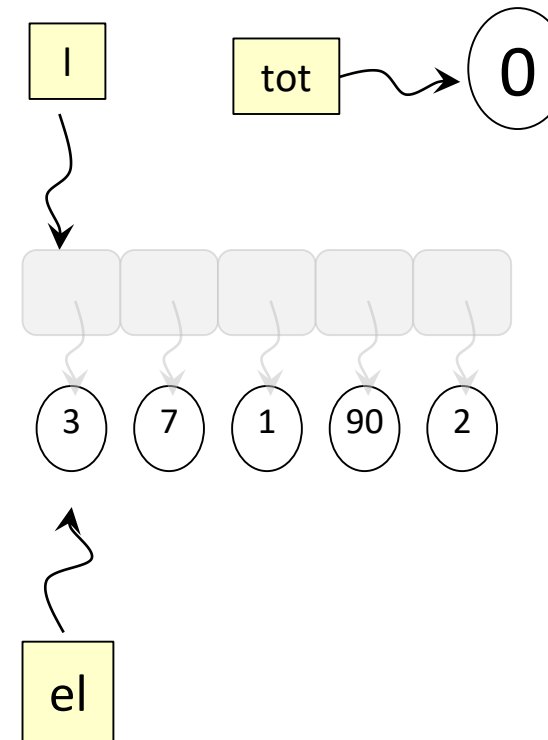
# Soma valores de l1 (1/11)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



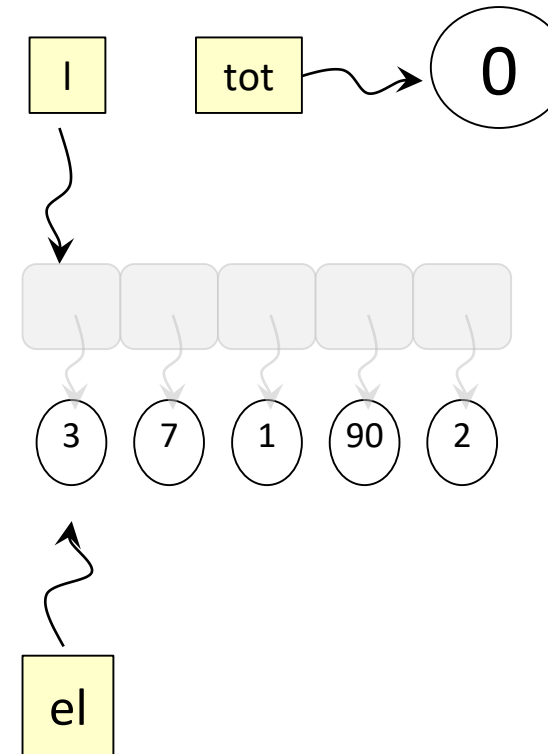
# Soma valores de l1 (2/11)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



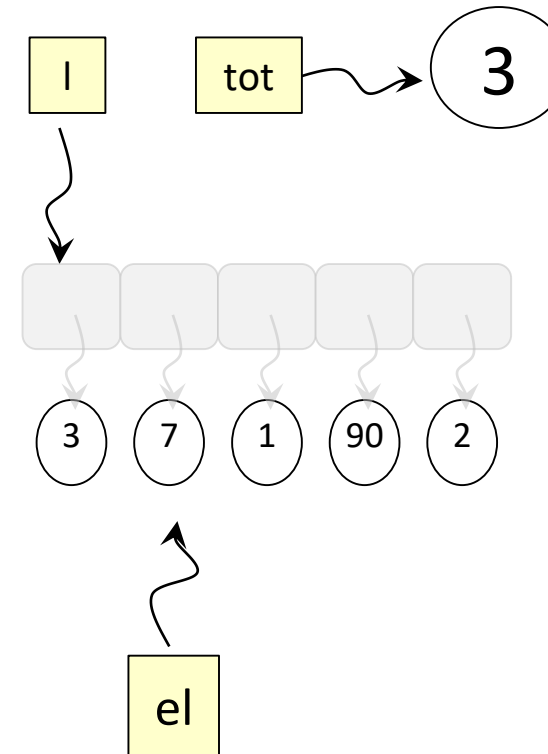
# Soma valores de l1 (3/11)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



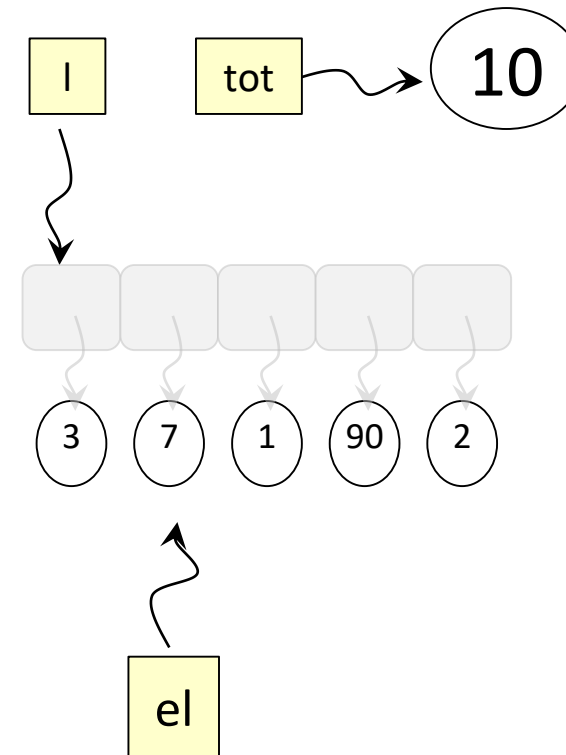
# Soma valores de l1 (4/11)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



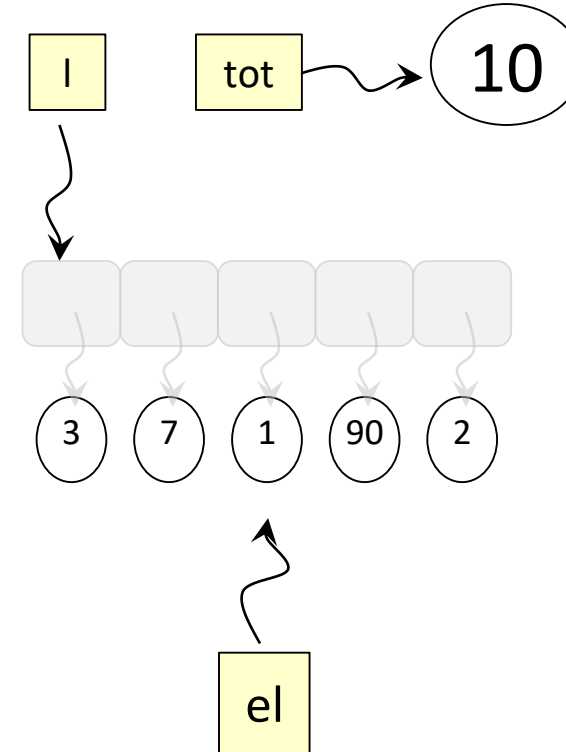
# Soma valores de l1 (5/11)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



# Soma valores de l1 (6/11)

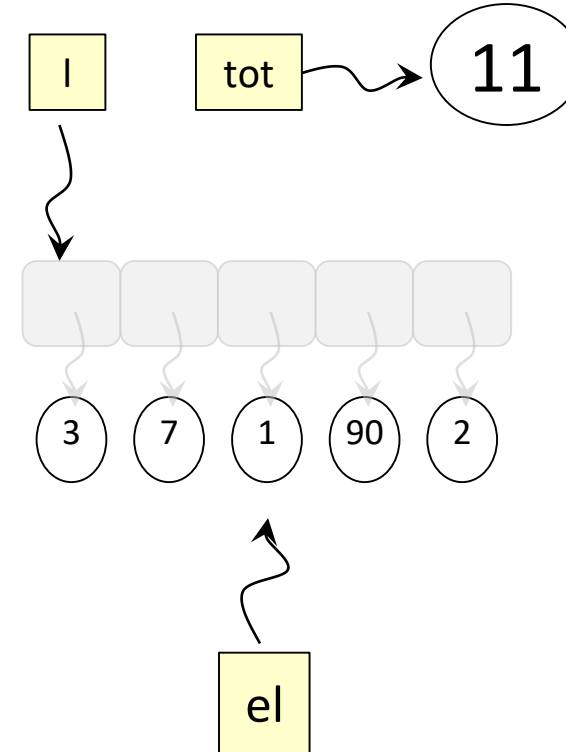
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```





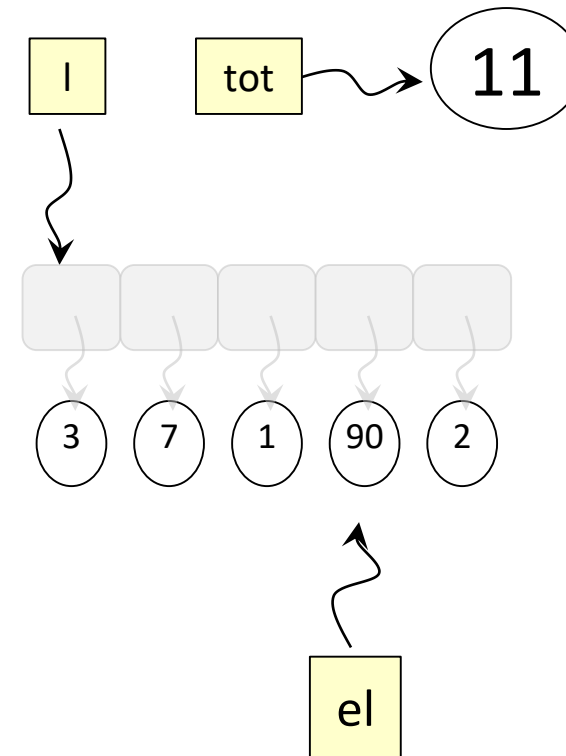
# Soma valores de l1 (7/11)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



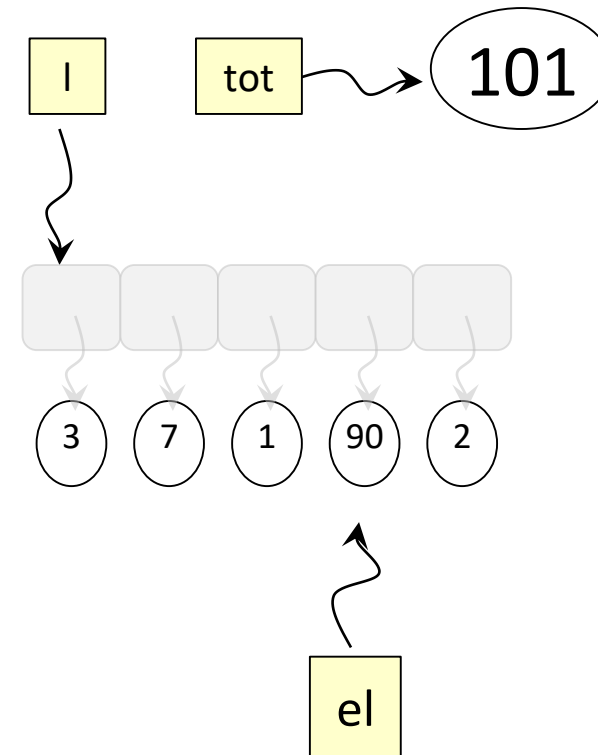
# Soma valores de l1 (8/11)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



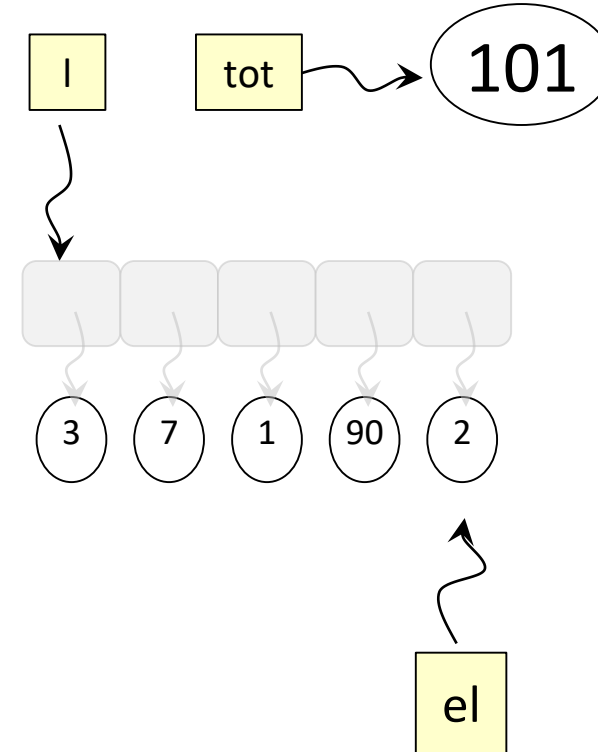
# Soma valores de l1 (9/11)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



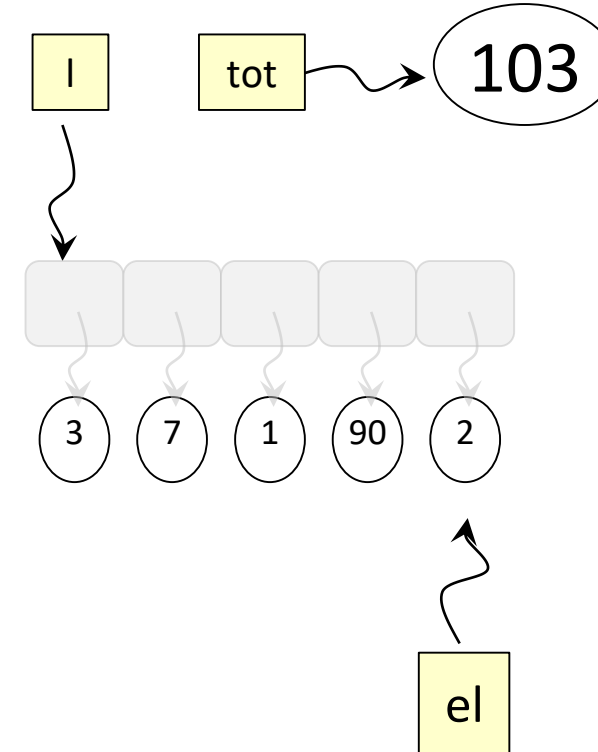
# Soma valores de l1 (10/11)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



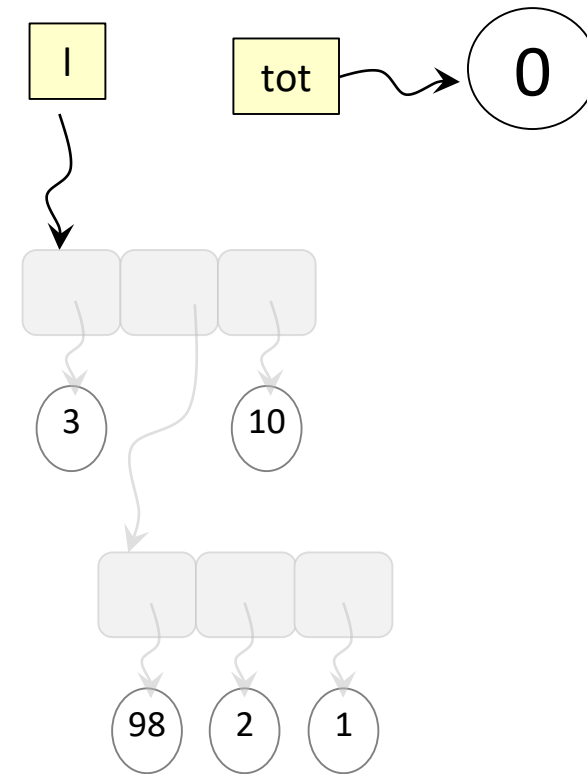
# Soma valores de l1 (11/11)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



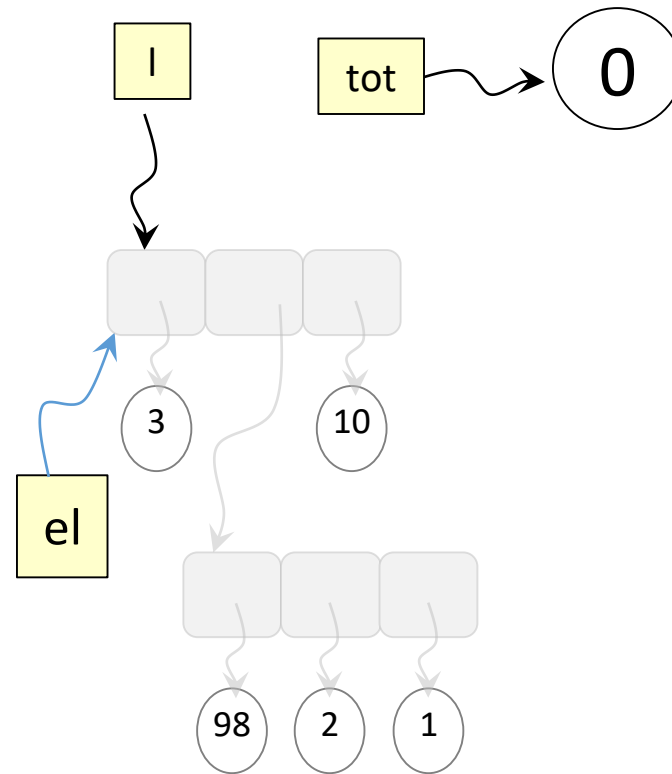
# Soma valores de l2 (1/6)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



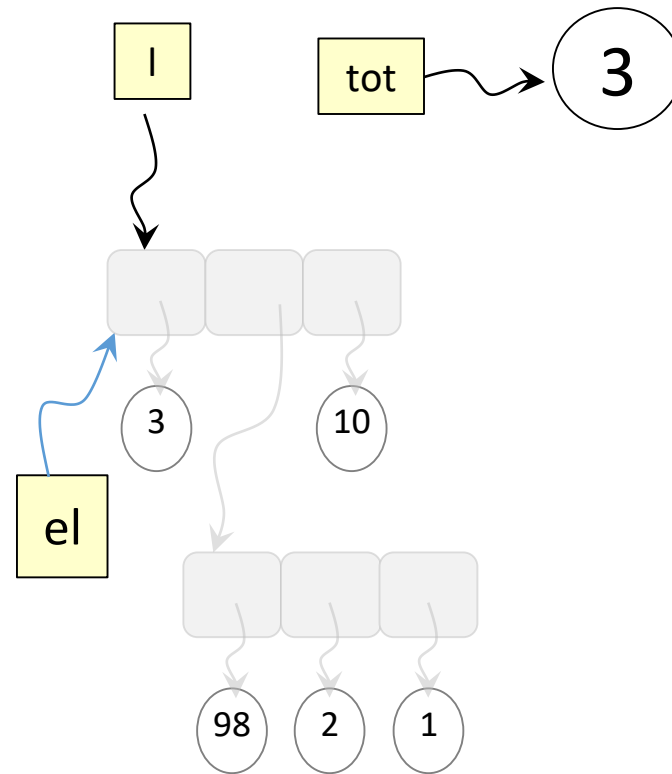
# Soma valores de l2 (2/6)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



# Soma valores de l2 (3/6)

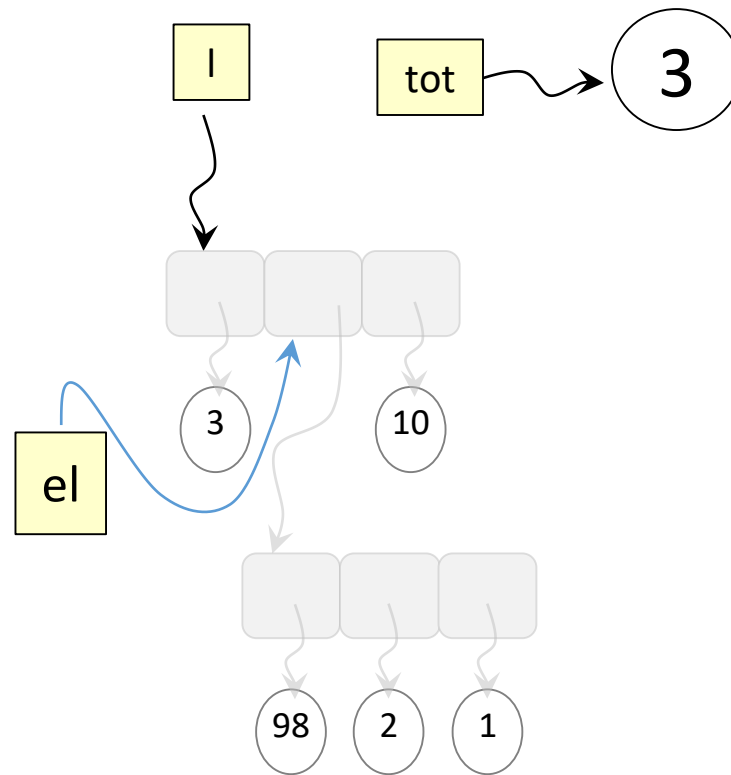
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```





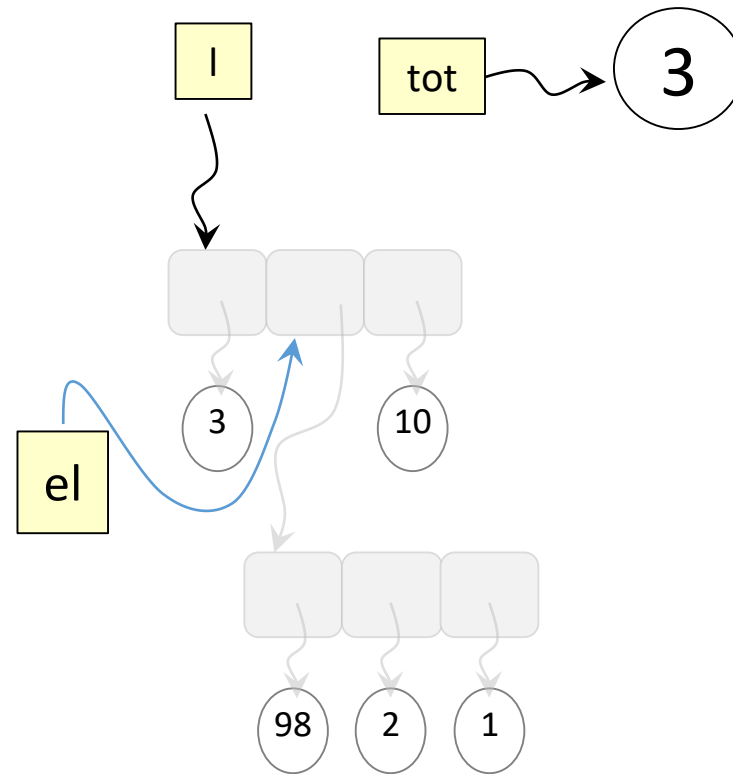
# Soma valores de l2 (4/6)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



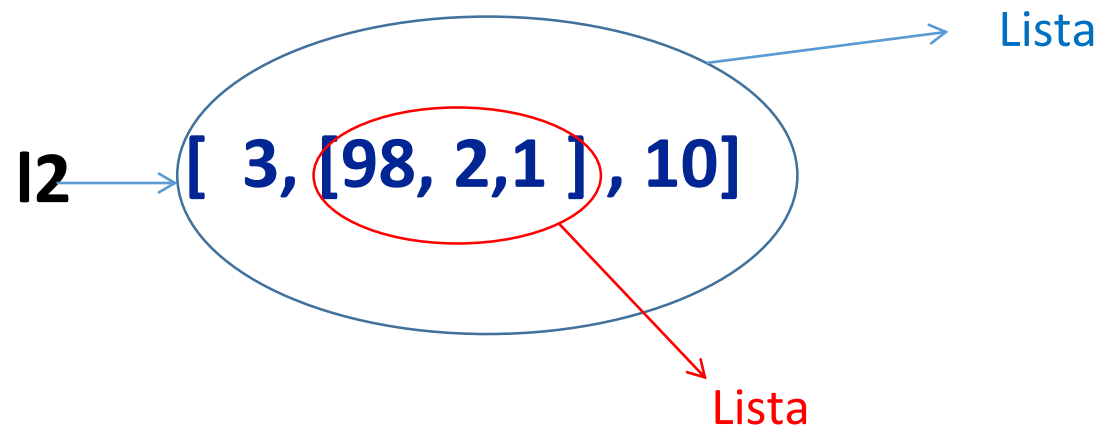
# Soma valores de l2 (5/6)

```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



Como o 2º elemento de `l2` é uma lista,  
não pode ser somado à variável `tot`:

**TypeError: unsupported operand type(s) for +=: 'int' and 'list'**



Somar lista:

Caso 1) o elemento é um inteiro:

somar ao total

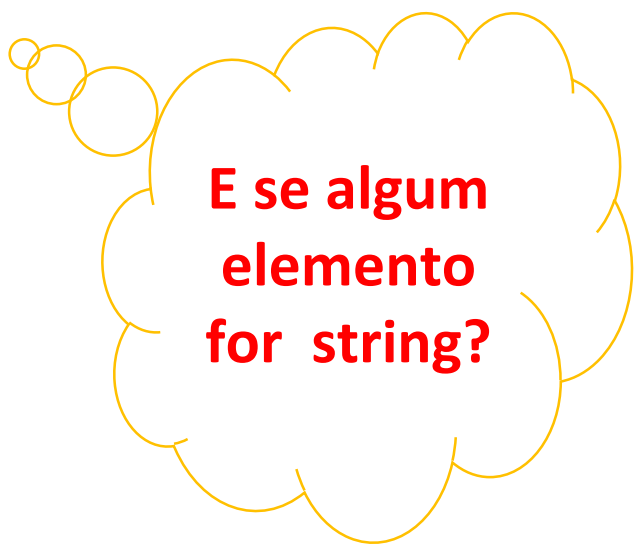
Caso2) o elemento é uma lista:

somar lista e resultado somar ao total

**RECURSÃO!!!**

# Soma elementos com strings

```
def somaEls(l):  
    tot=0  
    for el in l:  
        if type(el)== int:  
            tot+=el  
        else:  
            tot+= somaEls(el)  
    return tot  
  
l1= [3,7,1,90,2]  
l2=[3,[98,2,1],10]  
print(somaEl(l1))  
print(somaEls(l2))
```

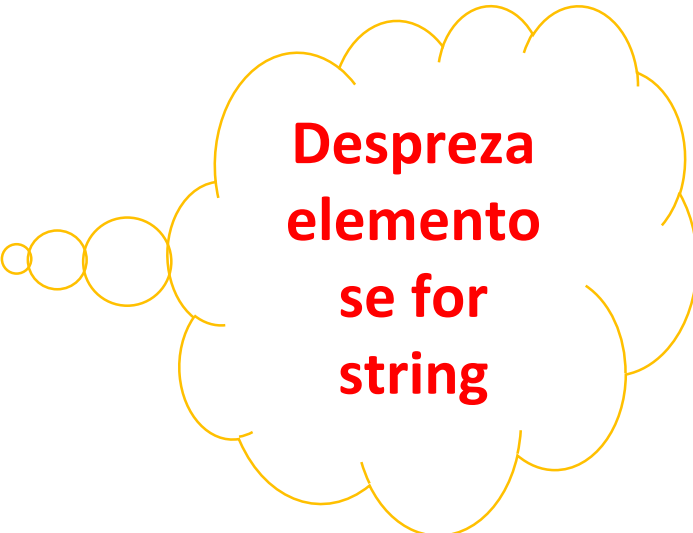


**E se algum  
elemento  
for string?**

# Resolvendo strings em listas

```
def somaEls(l):  
    tot=0  
    for el in l:  
        if type(el)== int:  
            tot+=el  
        else:  
            tot+= somaEls(el)  
    return tot
```

```
l1= [3,7,1,90,2]  
l2=[3,[98,2,1],10]  
print(somaEl(l1))  
print(somaEls(l2))
```



**Despreza  
elemento  
se for  
string**

# Mãos na massa!!! Corrigindo a provinha de matemática

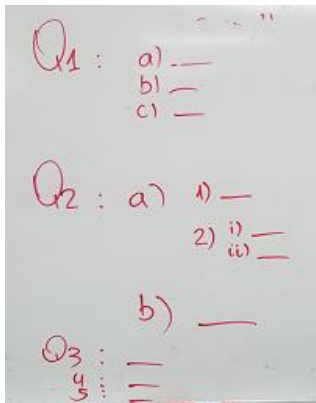
Faça um programa que apresente as questões da prova ao aluno, uma a uma, pergunte sua resposta, envie uma mensagem de acerto ou erro. No final o programa deve exibir a nota do aluno (cada acerto vale 2 pontos)

Modificação: Cada vez que o aluno errar, mostrar a resposta correta

`IQuestoes=['3+5=....', '9*3=....', '2+3*4=....', '7-5+9=....', '4*6/3+2=....']`

`IGabarito=[8,27,14,11,10]`

Desafio: Considere uma prova onde algumas questões contém várias subquestões (que podem também ser divididas em subquestões). Construa uma função que receba o gabarito e a resposta do aluno, retornando o número de acertos



Exemplo: Gabarito: `[[1,2,3],[[4,[5,6]],7],8,9,10]` → Uma prova com 5 questões.

Q1: 3 subitens com respostas únicas

Q2: 2 subitens,

Q2 a) tem duas perguntas ( a 1ª(1) com uma resposta e a 2ª (2) com 2 respostas (i,ii))

Q2 b) tem uma resposta.

Q3, Q4, Q5: resposta simples (única)

Crie a lista abaixo:

```
lgrande = random.sample(range(-20, 20),14)
```

```
lgrande[0]=random.sample(range(-10,10),3)
```

```
lgrande[4]=random.sample(range(100),2)
```

```
lgrande[12]=random.sample(range(-40,16),4)
```

Faça as funções abaixo e teste-a para a lista lgrande:

- a) retorne o produto dos números de uma lista
- b) retorne a média dos números de uma lista
- c) retorne o maior valor da lista
- d) substitua todos os múltiplos de 4 pelo número 0 na lista recebida
- e) receba também um número informado pelo usuário e retorne True, se o número está na lista e False, caso contrário

# Mãos na massa!!!

Crie a lista abaixo e utilize-a para testar as funções.

```
Lstrings=['Oi',['estamos','aprendendo',' a trabalhar'], 'com', 'listas', ['de',' strings']]
```

1. Faça uma função que concatene todas as palavras de uma lista de strings, retornando a string concatenada
2. Faça uma função que retorne a quantidade de caracteres da maior palavra de uma lista de strings
3. Faça uma função que retorne a palavra com a maior quantidade de caracteres de uma lista de strings
4. Faça uma função que retorne a quantidade de palavras contidas em uma lista de strings
5. Faça uma função que coloque todas as palavras de uma lista de strings em maiúsculo



# Exercício: campeonato de xadrez

Faça uma função que receba uma lista com 5 nomes de jogadores que participam de um campeonato de xadrez e exibe a tabela de jogos considerando que o 1º jogador joga com as peças brancas e há sempre dois jogos (com as brancas e com as pretas)

Teste para a lista: ['Jo','Ana','Zé','Pedro','Vivi']

Jo: _____	x	Ana: _____
Jo: _____	x	Zé : _____
Jo: _____	x	Pedro : _____
Jo: _____	x	Vivi : _____
....		
Vivi: _____	x	Jo: _____
Vivi: _____	x	Ana: _____
Vivi: _____	x	Zé: _____
Vivi: _____	x	Pedro: _____

# Xadrez: uma solução

```
def  tabela(lJog):  
    for jog1 in lJog:  
        for jog2 in lJog:  
            if jog1 != jog2:  
                print( "%6s_____ x %6s_____" % (jog1,jog2))  
    return  
  
lJog =  ['Jo','Ana','Zé','Pedro','Vivi']  
tabela(lJog)
```

# Exercício: agenda escolar

Construa uma função para exibir uma agenda para anotar as disciplinas por dia da semana, conforme modelo:

```
Segunda
 7:00: _____
 9:00: _____
11:00: _____
13:00: _____
15:00: _____
17:00: _____

Terça
 7:00: _____
 9:00: _____
11:00: _____
13:00: _____
15:00: _____
17:00: _____

Quarta
 7:00: _____
 9:00: _____
11:00: _____
13:00: _____
15:00: _____
17:00: _____

Quinta
 7:00: _____
 9:00: _____
11:00: _____
13:00: _____
15:00: _____
17:00: _____

Sexta
 7:00: _____
 9:00: _____
11:00: _____
13:00: _____
15:00: _____
17:00: _____
```

# Uma solução para agenda

```
def agenda(ldias):  
    for dia in ldias:  
        print( '\n'+dia)  
        for hora in range(7,18,2):  
            print( "%2d:00:_____"%hora)  
    return  
  
ldias = ['Segunda', 'Terça', 'Quarta', 'Quinta', 'Sexta']  
agenda(ldias)
```

# Exercício: dias entre duas datas

Faça um programa que leia o dia e mês de dois eventos no mesmo ano, mostrando quantos dias há entre eles. Não se preocupe com anos bissextos e considere que o evento 1 é sempre anterior ao evento 2.

A leitura da data de um evento é no seguinte formato: dd/mm.

Exemplos:

a) Evento é no mês corrente: evento 1:19 04 evento 2: 25 04

Meses incompletos: 04 ( 30 – 19 dias) = 11 dias

b) Evento não é no mês corrente: evento1:19 04 evento 2: 25 08

Meses incompletos: 04 ( 30 – 19 dias) = 11 dias

Meses completos: 05, 06, 07

Meses incompletos: 08 = 25 dias

**DICA p/Solução:** Crie uma lista com os dias em cada mês:

meses = [31,28,31,30,31,30,31,31,30,31,30,31]

Modifique o  
programa para que  
os valores digitados  
sejam validados.

# Solução para ler datas do ano

```
def leData(num, meses):  
    data=input('Dia/mes do %d evento? dd/mm: '%num)  
    #separa dia e mes  
    pos=data.find('/')  
    dia=int(data[:pos])  
    mes = int(data[pos+1:])  
    return [dia,mes]
```

```
meses=[31,28,31,30,31,30,31,31,30,31,30,31]
evento1 = leData(1,meses)
evento2 = leData(2,meses)
#meses incompletos
dias = (meses[evento1[1]-1]-evento1[0] ) + (evento2[0])
#meses completos
for mes in range(evento1[1],evento2[1]-1):
    dias+=meses[mes]
print('Entre os eventos há %d dias'%dias)
```

```
def leData(num,meses):
    while(True):
        data=input('Dia/mes do %d evento? dd/mm: '%num)
        #separa dia e mes
        pos=data.find('/')
        dia=int(data[:pos])
        mes = int(data[pos+1:])
        #data válida?
        if( mes >=1 and mes<=12):
            if (dia>=1 and dia <=meses[mes-1]):
                return [dia,mes]
            else:
                print( 'mes %d tem no máximo %d dias'
                    %(mes,meses[mes-1]))
        else:
            print('mes inválido-valor entre 1 a 12')
```



# Exercício: temperatura média

---

Crie uma função em Python que receba uma lista com a temperatura média de cada mês do ano.

Dadas as temperaturas, esta função deve calcular a média anual das temperaturas e mostrar todas as temperaturas acima da média anual, e em que mês elas ocorreram

Atenção: mostrar o mês por extenso:

1 – Janeiro, 2 – Fevereiro, . . .

```
def calcMedia(lista):
    tot=0
    cont = 0
    for el in lista:
        tot = tot + el # tot += el
        cont=cont+1
    media = tot/cont
    return media

def acimaMedia(lista):
    lmeses=['janeiro','fevereiro','março','abril','maio','junho',
            'julho','agosto','setembro','outubro','novembro','dezembro']
    media=calcMedia(lista)
    for i in range(len(lista)):
        if lista[i] > media:
            print(lmeses[i],'acima da média, temp =',lista[i])
    return

import random
ltemp = random.sample(range(-100,100),12)
acimaMedia(ltemp)
```

# Exercícios: números e múltiplos

Crie uma função que receba uma lista  $l$  e substitua todos os seus elementos que são múltiplos de seu índice pelo valor do elemento sucessor.

**CUIDADO:** o 1º elemento tem índice 0 e não pode "entrar".

Lembre também que o último elemento não tem sucessor!

Exemplo:

$$l = [1, 2, 3, 6, 8] \rightarrow l = [1, 3, 3, 8, 8]$$

```
def subSuc (l) :  
    for i in range (1, len (l) -1) :  
        if l[i] % i == 0 :  
            l[i] = l[i+1]  
    return
```