

# 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

3

## 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.

4



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

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? .....

5



## Corrigir provas - Principal

Questão 1

Os personagens de personagens da Disney chamados de Palo

Questão 2

Questão 3

Questão 4

Questão 5

Questão 6

Questão 7

Questão 8

Questão 9

Questão 10

Questão 11

Questão 12

Questão 13

Questão 14

Questão 15

Questão 1

Questão 2

Questão 3

Questão 4

Questão 5

Questão 6

Questão 7

Questão 8

Questão 9

Questão 10

Questão 11

Questão 12

Questão 13

Questão 14

Questão 15

### 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

6

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

<http://logicandgames.blogspot.com.br/2011/09/mastermind.html>

7

## Jogo do MasterMind (ou Senha)

Escolha a quantidade de cores: 3 4 5 6 7 8 9 10 11 12

Selecione a cor:

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

Escolha o tamanho da sequência: 3 4 5 6

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

<http://logicandgames.blogspot.com.br/2011/09/mastermind.html>

8

Passos de uma possível solução:

- Exibir mensagem de Parabéns!!!

senão

Exibir mensagem de Consolação!!!

Sequências numéricas:

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



## Sequências numéricas



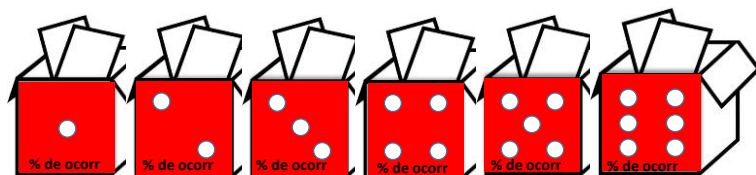
Passos de uma possível solução:

- 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!!!

11

## Contextualização #5

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



12

## Dados viciados?



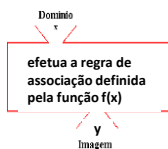
Passos de uma possível solução:

- 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

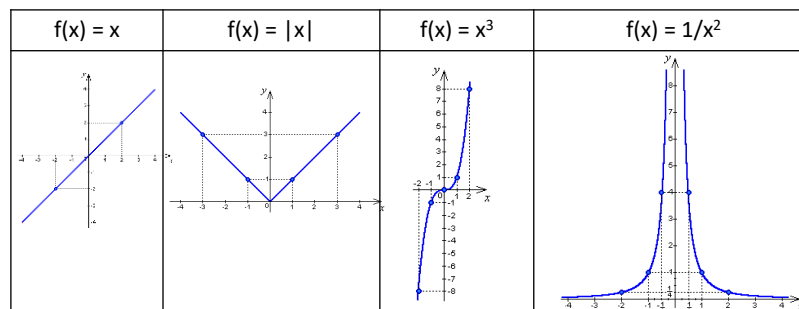
13

## Contextualização #6

Traçar gráfico de uma função:

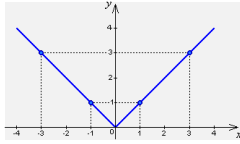


O gráfico cartesiano de uma função  $f$  é o conjunto de todos os pontos  $(x, f(x))$  do plano com  $x$  variando no domínio de  $f$



14

## 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

15

## 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?

16



## Tipo Lista em Python

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

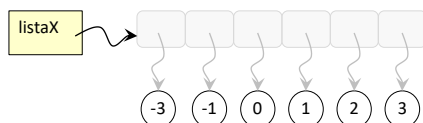
**Exemplo: Lista com o gabarito da 2ª prova:**

```
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 ]
```

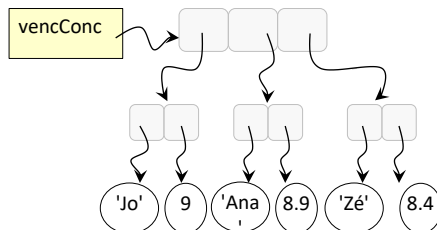
17

## Lista em Python: representação

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



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



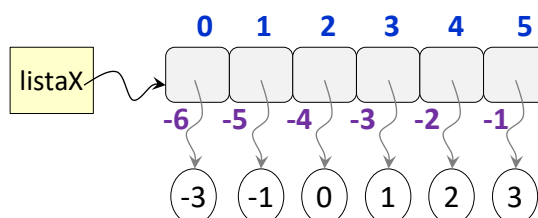
18

## 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)

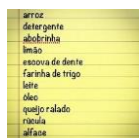
Exemplo: listaX = [-3,-1,0,1,2,3]



19

## Tipo Lista em Python - Dinâmica

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



**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

20



## Criando uma lista por enumeração

Lista Vazia:

```
nomeVariável Lista = []
```

Lista com elementos enumerados :

```
nomeVariável Lista = [ el1, el2, ..., eln ]
```

21



## 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

22

## 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
```

23

## 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: []
```

24

## Acesso aos elementos: indexação

Pelo operador de indexação `[]`.

**`varLista[expressão inteira]`**

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

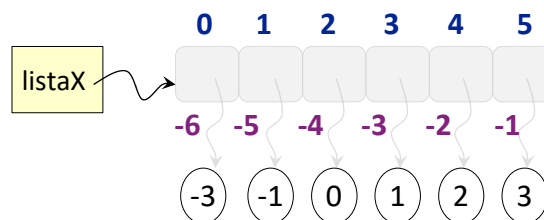
- ✓ Índices positivos ( 0 até comprimento da lista-1) indicam a posição do elemento na lista da esquerda para a direita.
- ✓ Índices negativos ( -1 até -comprimento da lista) indicam a posição do elemento da direita para a esquerda

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

25

## Indexando elementos da lista (1/7)

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



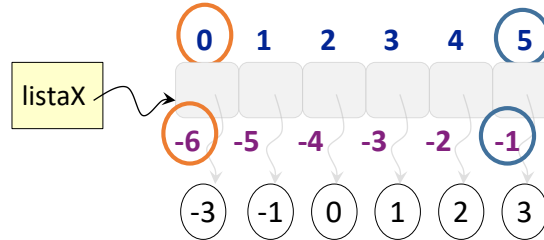
Qual o índice do primeiro elemento?

Qual o índice do último elemento?

26

## Indexando elementos da lista (2/7)

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



Qual o índice do primeiro elemento? 0

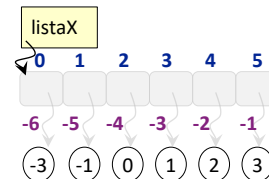
Qual o índice do último elemento? -1 ou len(listaX)-1

27

## Indexando elementos da lista (3/7)

**Exemplo 1:**

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



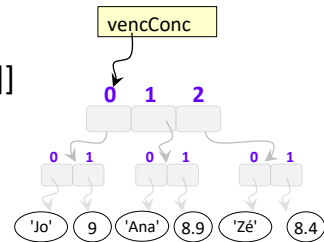
```
tam = len(listaX)
print(listaX[ 2 ])      → 0
print(listaX[ 9-8 ])    → -1
print(listaX[ -2 ])     → 2
print(listaX[ tam-1 ])  → 3
print(listaX[ -1 ])     → 3
print(listaX[ tam ])    → IndexError: list index out of range
```

28

## Indexando elementos da lista (4/7)

### Exemplo 2:

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



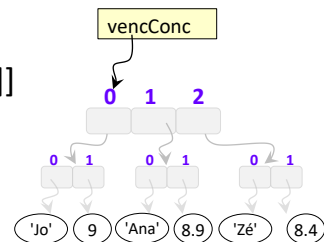
<code>print(vencConc[0])</code>	→	<code>['Jo', 9]</code>
<code>print(vencConc[-len(vencConc)])</code>	→	<code>['Jo', 9]</code>
<code>print(vencConc[-4])</code>	→	<b>IndexError:</b> list index out of range

29

## Indexando elementos da lista (5/7)

### Exemplo 2:

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

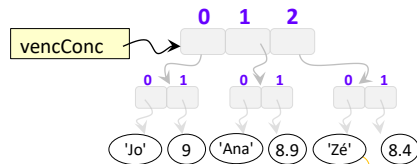


<code>print(vencConc[len(vencConc)-1])</code>	→	<code>['Zé', 8.4]</code>
<code>print(vencConc[-1])</code>	→	<code>['Zé', 8.4]</code>
<code>print(vencConc[9-10])</code>	→	<code>['Zé', 8.4]</code>

30

## Indexando elementos da lista (6/7)

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

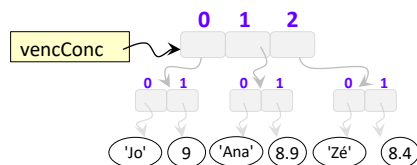


Como acessar  
para exibir o  
nome do terceiro  
colocado?

31

## Indexando elementos da lista (7/7)

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



Como  
acessar  
para exibir  
o nome do  
terceiro  
colocado?

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

32



## Alteração de um elemento (1/5)

**`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`

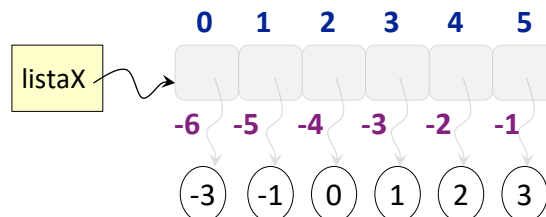
`listaX[ 5 ] = [4,7]`

33

## Alteração de um elemento (2/5)

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

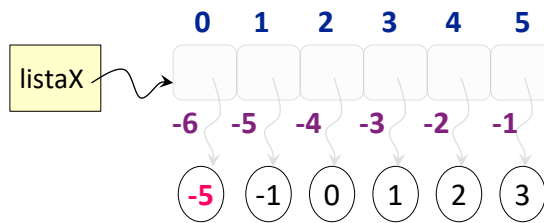
**`listaX[ 0 ] = -5`**



34

## Alteração de um elemento (3/5)

Exemplo 1: **[ -5,-1, 0, 1, 2, 3 ]**

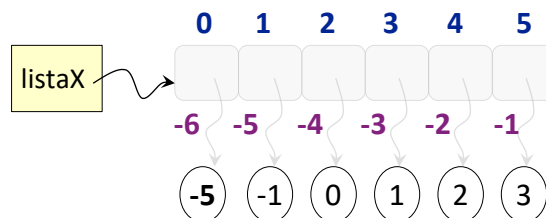


35

## Alteração de um elemento (4/5)

Exemplo 2:

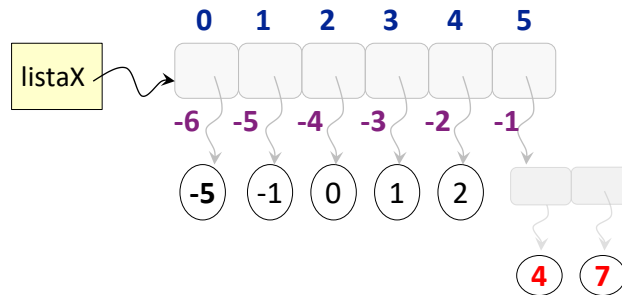
**listaX[ 5 ] = [4,7]**



36

## Alteração de um elemento (5/5)

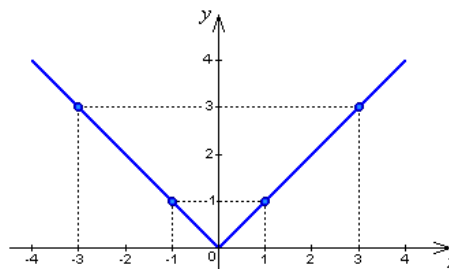
Exemplo 2:  $[-5, -1, 0, 1, 2, [4, 7]]$



37

## 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  $(x, y = f(x))$

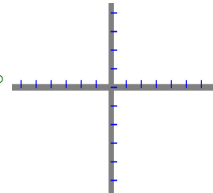
38

## Eixos Cartesianos e ponto

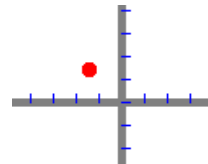
Importar as funções abaixo do módulo *eixoscartesianos*

Obs. baixar do EAD da disciplina INF1025

```
def plotEixosCartesianos(t,npontos,larg)
    """ Desenha um eixo cartesiano, com espessura larg e dividido
    npontos. Linha em cinza e traços em azul
    """
```



```
def plotponto(t,x,y,larg):
    """ Desenha um círculo no ponto (x,y) no plano, com
    diâmetro larg em vermelho
    """
```



39

## Funções e Eixos Cartesianos

- 1) Desenvolver a função  $f(x) = |x|$  (ou usar `abs`)
- 2) Criar uma lista de valores para x (9 valores  $\in [-25,25]$ )
- 3) Usando as funções fornecidas para traçar o eixo cartesiano (*plotEixoCartesiano*) e um ponto neste eixo (*plotPonto*), desenhar o(s) ponto(s) equivalente(s) ao:

- a) primeiro elemento da lista
- b) primeiro e último elemento da lista
- c) primeiro, último e elemento do meio da lista

40



## Uma Solução: eixos cartesianos

```
from eixocartesiano import plotEixosCartesianos, plotponto
import turtle

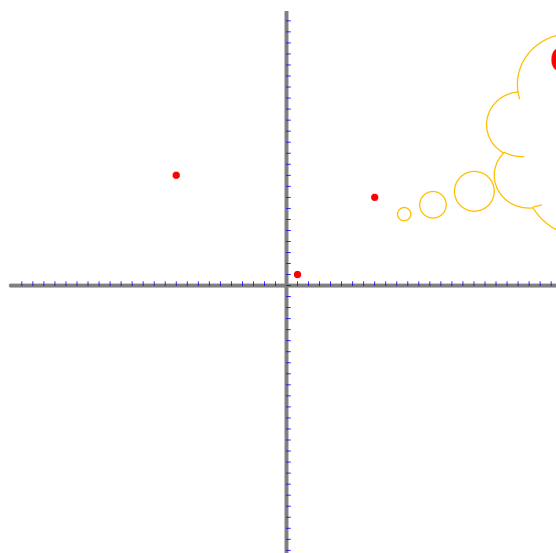
def modulo(x):
    if x < 0:
        return -x
    else:
        return x

t = turtle.Turtle()
plotEixosCartesianos(t, 25, 14)
pontos = [-10, -8, -4, 0, 1, 2, 3, 4]
plotponto(t, pontos[0], modulo(pontos[0]), 14)
plotponto(t, pontos[-1], modulo(pontos[-1]), 14)
meio = len(pontos) // 2
plotponto(t, pontos[meio], modulo(pontos[meio]), 14)
```

41



## Solução para todos pontos!

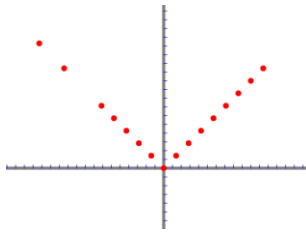


Como plotar  
todos os  
pontos da  
lista?

42



## 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!!!**

43



## 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)
```

44

## Comando FOR e Função RANGE

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

Gera os índices da lista sequencialmente, de 0 a nº de elementos-1

Exemplo:

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

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

45

## Plotar todos pontos da lista (1/30)

```
def desenhafuncao(t,l,larg):
    """Percorre a lista e, para cada elemento,desenha pto(x,y)"""
    for x in l:
        y=modulo(x)
        plotponto(t,x, y ,larg)
    return

pontos=[-2,-1,0,1,2]
desenhafuncao(t,pontos,20)
```

46

## Plotar todos pontos da lista (2/30)

```
def desenhafuncao(t,l,larg):  
    """Percorre a lista e, para cada elemento,desenha pto(x,y)"""  
    for x in l:  
        y=modulo(x)  
        plotponto(t,x,y,larg)  
    return  
    plotponto(t,x,modulo(x),larg)
```

```
pontos=[-2,-1,0,1,2]  
desenhafuncao(t,pontos,20)
```

47

## Plotar todos pontos da lista (3/30)

```
def desenhafuncao(t,l,larg):  
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)  
    for x in l:  
        plotponto(t,x, modulo(x),larg)  
    return
```

```
➔ pontos=[-2,-1,0,1,2]  
desenhafuncao(t,pontos,20)
```

48



## Plotar todos pontos da lista (4/30)

```
def desenhafuncao(t,l,larg):  
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)  
    for x in l:  
        plotponto(t,x, modulo(x),larg)  
    return
```

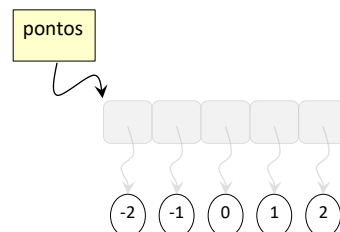
➔ pontos=[-2,-1,0,1,2]  
desenhafuncao(t,pontos,20)

49

## Plotar todos pontos da lista (5/30)

```
def desenhafuncao(t,l,larg):  
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)  
    for x in l:  
        plotponto(t,x, modulo(x),larg)  
    return
```

➔ pontos=[-2,-1,0,1,2]  
desenhafuncao(t,pontos,20)

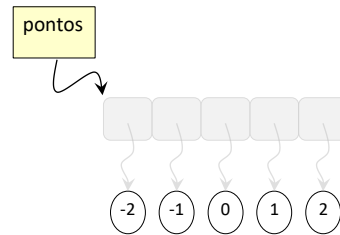


50

## Plotar todos pontos da lista (6/30)

```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```

```
pontos=[-2,-1,0,1,2]
➔ desenhafuncao(t,pontos,20)
```

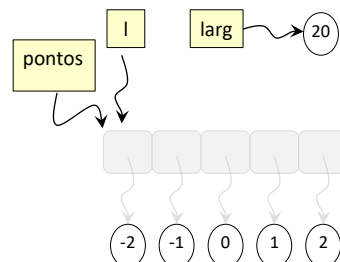


51

## Plotar todos pontos da lista (7/30)

```
➔ def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```

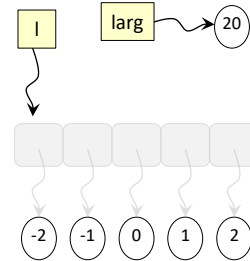
```
pontos=[-2,-1,0,1,2]
desenhafuncao(t,pontos,20)
```



52

## Plotar todos pontos da lista (8/30)

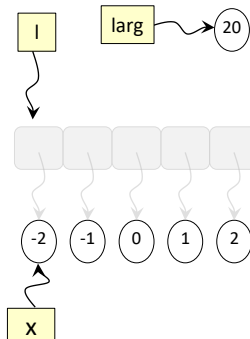
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    ➔ for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```



53

## Plotar todos pontos da lista (9/30)

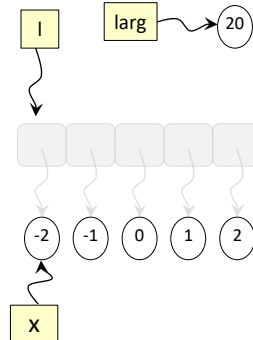
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    ➔ for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```



54

## Plotar todos pontos da lista (10/30)

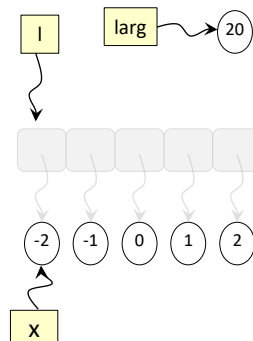
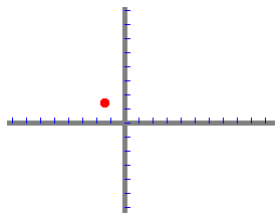
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        ➔ plotponto(t,x, modulo(x),larg)
    return
```



55

## Plotar todos pontos da lista (11/30)

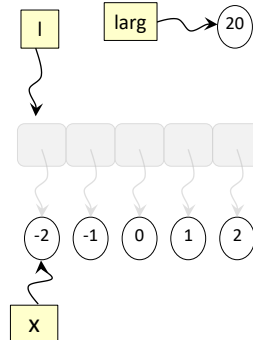
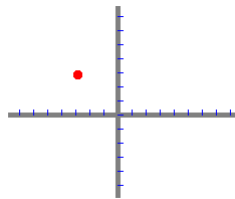
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        ➔ plotponto(t,x, modulo(x),larg)
    return
```



56

## Plotar todos pontos da lista (12/30)

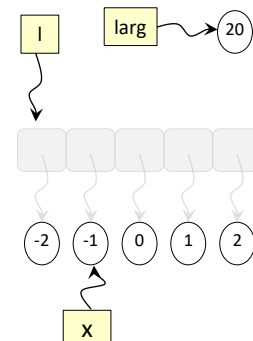
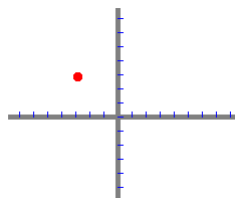
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    ➤ for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```



57

## Plotar todos pontos da lista (13/30)

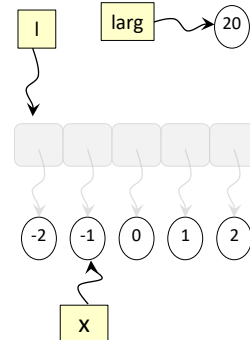
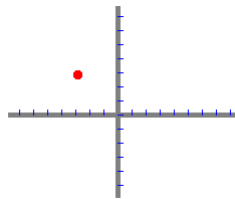
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    ➤ for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```



58

## Plotar todos pontos da lista (14/30)

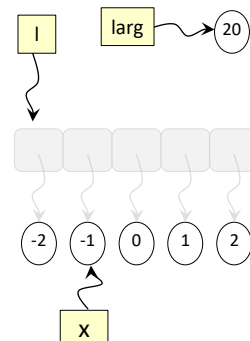
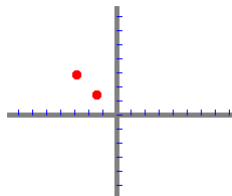
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        ➔ plotponto(t,x, modulo(x),larg)
    return
```



59

## Plotar todos pontos da lista (15/30)

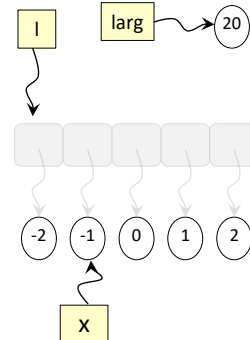
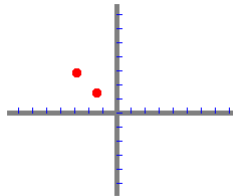
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        ➔ plotponto(t,x, modulo(x),larg)
    return
```



60

## Plotar todos pontos da lista (16/30)

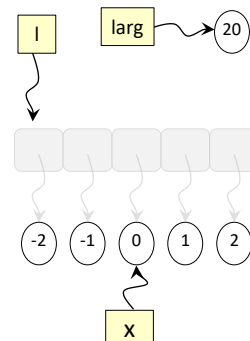
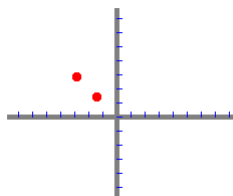
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    ➤ for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```



61

## Plotar todos pontos da lista (17/30)

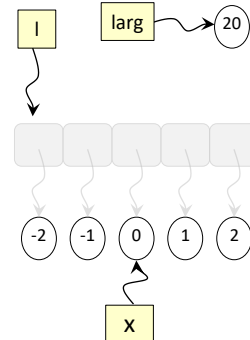
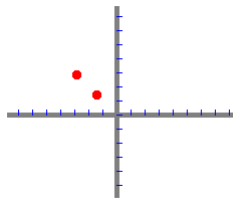
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    ➤ for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```



62

## Plotar todos pontos da lista (18/30)

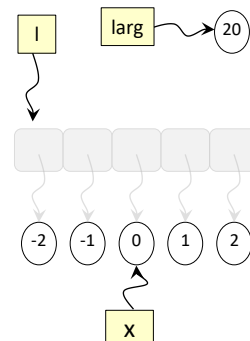
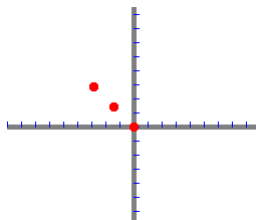
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        ➔ plotponto(t,x, modulo(x),larg)
    return
```



63

## Plotar todos pontos da lista (19/30)

```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        ➔ plotponto(t,x, modulo(x),larg)
    return
```

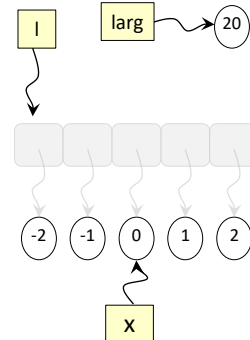
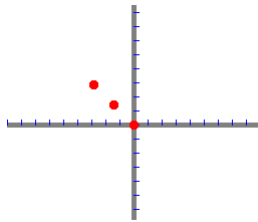


64



## Plotar todos pontos da lista (20/30)

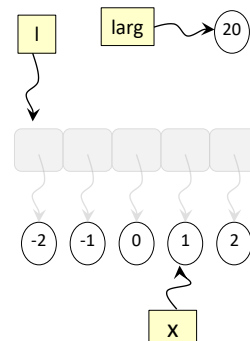
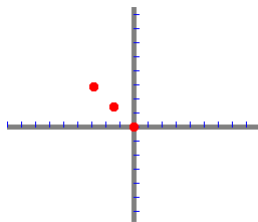
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    ➤ for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```



65

## Plotar todos pontos da lista (21/30)

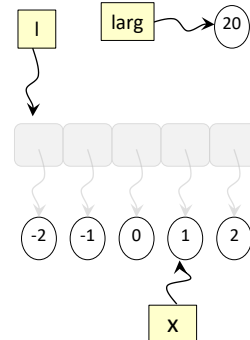
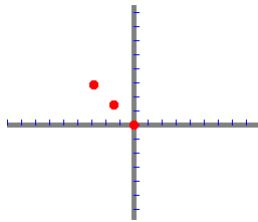
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    ➤ for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```



66

## Plotar todos pontos da lista (22/30)

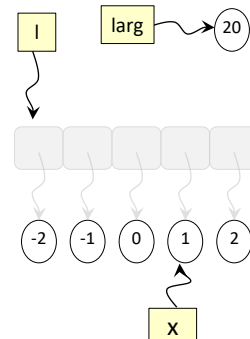
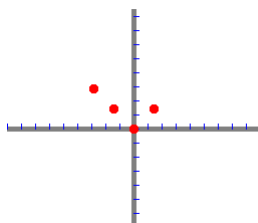
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        ➔ plotponto(t,x, modulo(x),larg)
    return
```



67

## Plotar todos pontos da lista (23/30)

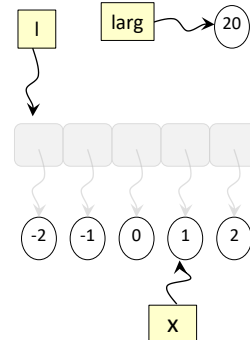
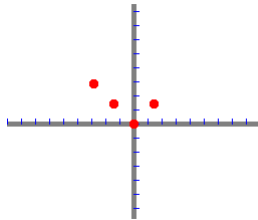
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        ➔ plotponto(t,x, modulo(x),larg)
    return
```



68

## Plotar todos pontos da lista (24/30)

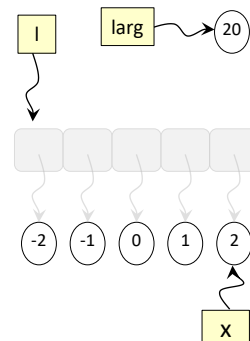
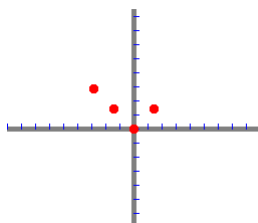
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    ➤ for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```



69

## Plotar todos pontos da lista (25/30)

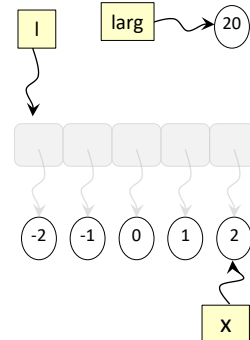
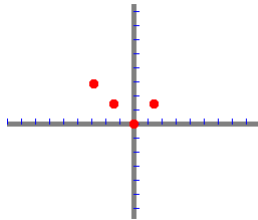
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    ➤ for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```



70

## Plotar todos pontos da lista (26/30)

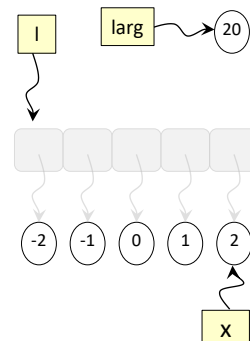
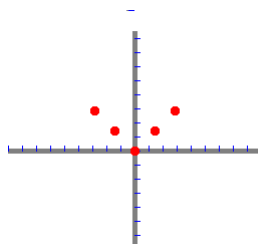
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        ➔ plotponto(t,x, modulo(x),larg)
    return
```



71

## Plotar todos pontos da lista (27/30)

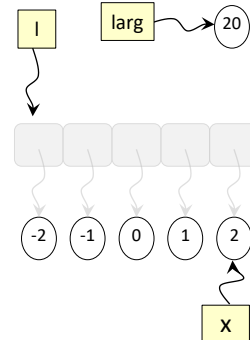
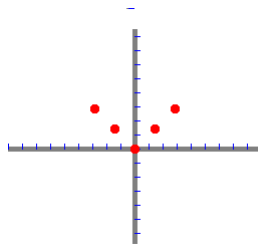
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        ➔ plotponto(t,x, modulo(x),larg)
    return
```



72

## Plotar todos pontos da lista (28/30)

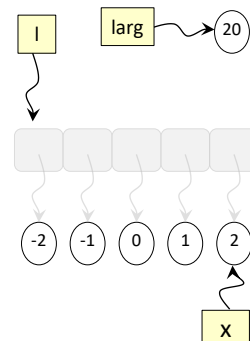
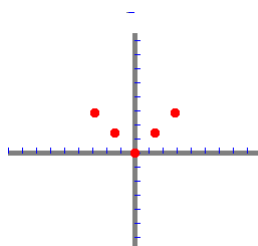
```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    ➤ for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```



73

## Plotar todos pontos da lista (29/30)

```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        plotponto(t,x, modulo(x),larg)
    ➤ return
```

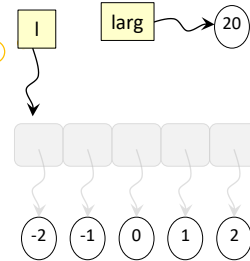


74

## Plotar todos pontos da lista (30/30)

```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```

O que muda, para  
plotar a função com  
 $x \in [-25,25]$ ?



75

## Função RANGE

Para *plotar* a função para  $x \in [-25,25]$ :

Criar uma lista com os valores

**-25,-24,-23,...,0,...,23,24,25**

```
def desenhafuncao(t,l,larg):
    #Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)
    for x in l:
        plotponto(t,x, modulo(x),larg)
    return
```

Por enumeração é impraticável !!!! Utilizar a **função range()**

76



## Criando uma lista de um intervalo

Relembrando:

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

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

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

77



## Listas e intervalos

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))      []
```

78



## $k$ elementos aleatórios em um intervalo

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

Módulo *random*

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

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

→ Retorna uma nova lista com os elementos escolhidos

79



## Listas aleatória com $k$ elementos

### 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]
```

80



## Plotando funções

1. Criar a função  $f(x) = x+1$
  2. Criar a função  $g(x) = (x\%3)^5$
  3. Crie duas funções: `desenhafuncao f` e `desenhafuncao g`
  4. Crie duas listas inicializadas (l1 e l2):
    - l1 com todos os valores entre [0,-10]
    - l2 com 11 valores, gerados aleatoriamente, entre [-25, 25]
- e:
- a) desenhe os pontos das funções acima para l1.  $f(x)$  em vermelho e  $g(x)$  em amarelo.
  - b) desenhe os pontos das funções acima para l2.  $f(x)$  em azul e  $g(x)$  em preto.

81

## Uma Solução: funções F e G

```
def f(x):
    return x+1

def g(x):
    return (x%3)**5

def desenhafuncao f(t,l,larg,cor):
    """Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)"""
    t.color(cor)
    for x in l:
        y=f(x)
        plotponto(t,x, y ,larg)
    return

def desenhafuncao g(t,l,larg,cor):
    """Percorrer a lista, elemento a elemento , desenhando o ponto (x,y)"""
    t.color(cor)
    for x in l:
        y=g(x)
        plotponto(t,x, y ,larg)
    return
```

82



## Solução: listas e chamadas de funções

```
l1=list(range(0,-11,-1))
l2=random.sample(range(-25,25),11)

t=turtle.Turtle()
plotEixosCartesianos(t,25,14)
desenhafuncao(t,l1,10,'red')
desenhafuncao(t,l2,10,'blue')
desenhafuncao(t,l1,10,'yellow')
desenhafuncao(t,l2,10,'black')
```

83



## Listas: exhibe e soma elementos

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

84

## Uma solução: exibe elementos

```
def exibe(l):  
    for el in l:  
        print(el)  
    return  
  
l1= [3,7,1,90,2]  
l2=[3,[98,2,1],10]  
exibe(l1)  
exibe(l2)
```

85

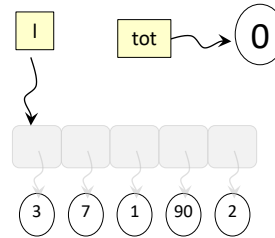
## Soma valores de uma lista

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

86

## Soma valores de l1 (1/11)

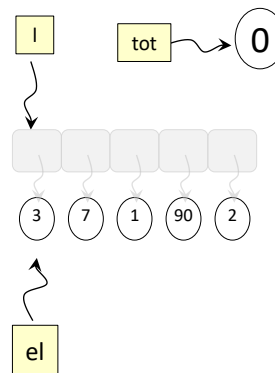
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



87

## Soma valores de l1 (2/11)

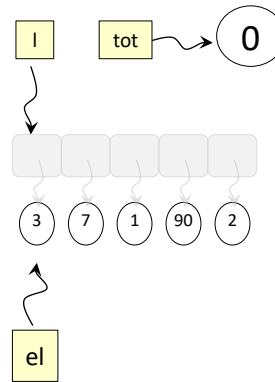
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



88

## Soma valores de l1 (3/11)

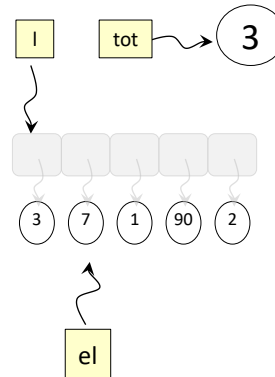
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



89

## Soma valores de l1 (4/11)

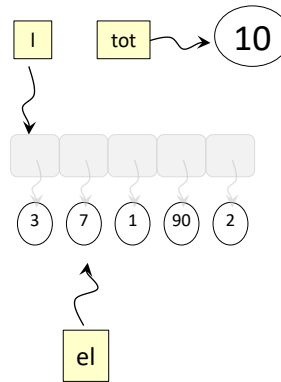
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



90

## Soma valores de l1 (5/11)

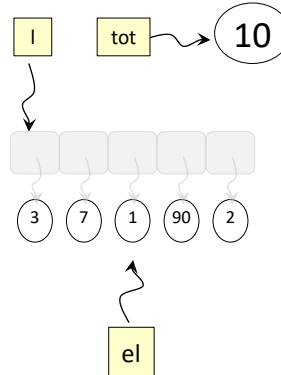
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



91

## Soma valores de l1 (6/11)

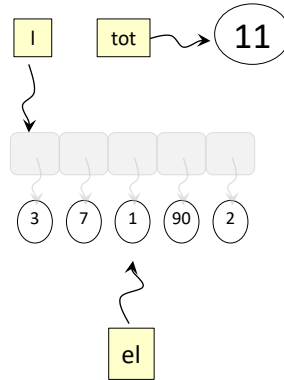
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



92

## Soma valores de l1 (7/11)

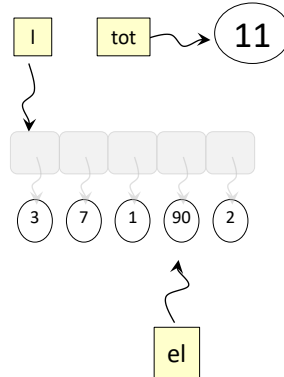
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



93

## Soma valores de l1 (8/11)

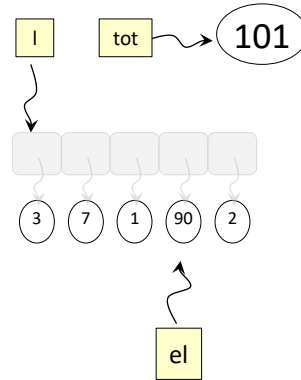
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



94

## Soma valores de l1 (9/11)

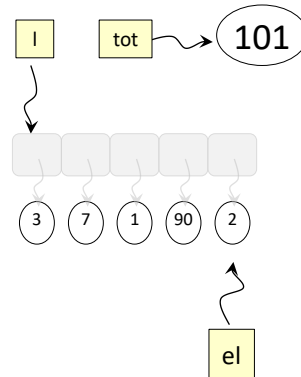
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



95

## Soma valores de l1 (10/11)

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

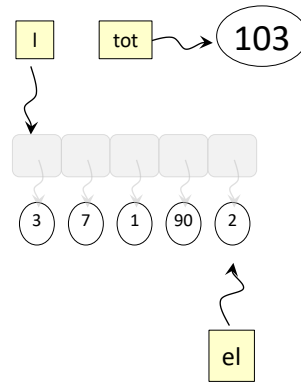


96



## Soma valores de l1 (11/11)

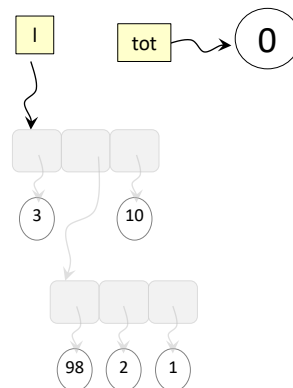
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



97

## Soma valores de l2 (1/6)

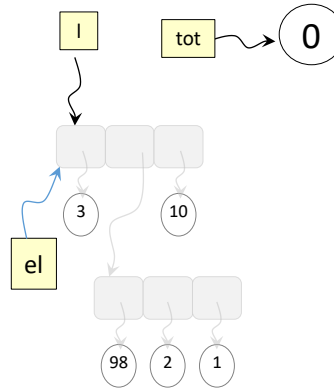
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



98

## Soma valores de l2 (2/6)

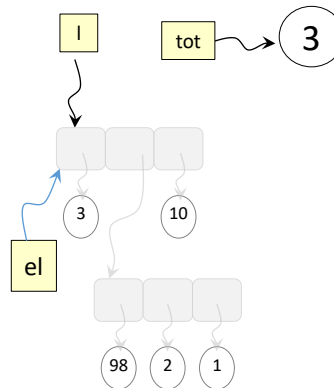
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



99

## Soma valores de l2 (3/6)

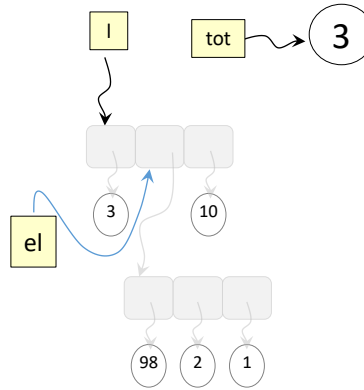
```
def somaEl(l):  
    tot=0  
    for el in l:  
        tot+=el  
    return tot
```



100

## Soma valores de l2 (4/6)

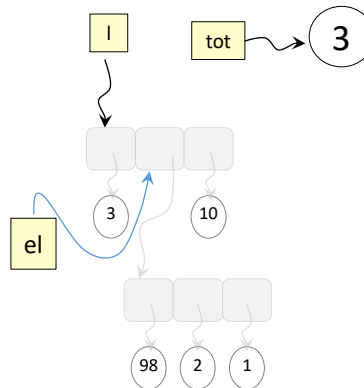
```
def somaEl(l):
    tot=0
    for el in l:
        tot+=el
    return tot
```



101

## Soma valores de l2 (5/6)

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

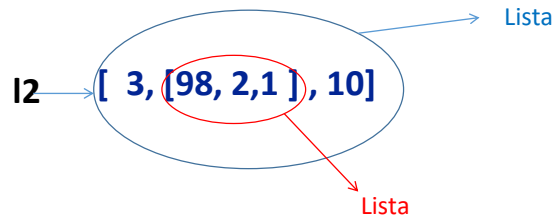


Como o 2º elemento de l2 é  
uma lista, não pode ser somado  
à variável tot:

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

102

## Soma valores de l2 (6/6)



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!!!**

103

## 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?**

104

## 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

105

## 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: _____

106

## 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)
```

107

## 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: _____
```

108

## 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)
```

109

## 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: evento 1:19 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.

110

## 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]
```

111

## Solução para diferença de datas

```
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)
```

112





## Solução modificada com validação

```
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')
```

113



## 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, . . .

114



## Solução das Temperaturas Médias

```
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)
```

115



## 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:

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

116

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