

IEC037

**Introdução à Programação de
Computadores**

**Aula 12 – Estruturas de Repetição por
Contagem em Python**

Conteúdo



Estruturas de Repetição por Contagem



Como carregar valores em uma lista?



Problemas comuns envolvendo listas

Conteúdo



Estruturas de Repetição por Contagem



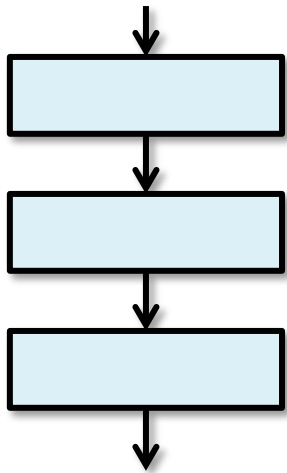
Como carregar valores em uma lista?



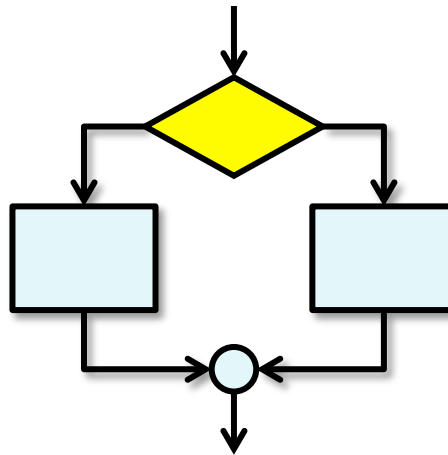
Problemas comuns envolvendo listas

Estruturas de Programação

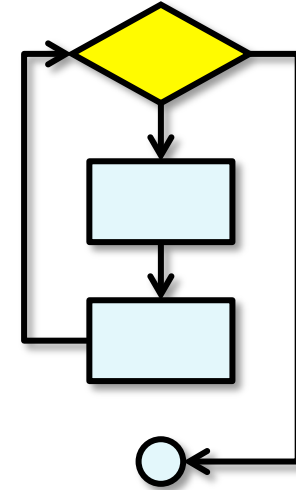
- Qualquer programa de computador pode ser escrito combinando-se os **três tipos básicos de estruturas de programação**:



Sequencial



Condicional



Repetição

Estruturas de Repetição

:: Critérios de repetição

Por condição (while)

- Repete um bloco de comandos enquanto uma **condição for verdadeira**.

Por contagem (for)

- Repete um bloco de comandos para cada **elemento de uma lista**.

Estruturas de Repetição

:: Exemplos

while

```
x = 1
while (x <= 3):
    print(x)
    x = x + 1
```

- 1
- 2
- 3

for

```
lst = [1, 2, 3]
for x in lst:
    print(x)
```

- 1
- 2
- 3



Quando usar **while** ou **for**?

while

- Usamos quando **não sabemos quantas vezes** vamos repetir.
- **Exemplo:** qual é e onde está o primeiro número negativo de uma lista?

for

- Usamos quando temos de processar **todos** os elementos de uma lista.
- **Exemplo:** qual é o maior elemento de uma lista?

Atenção



- Embora permitidas pelo Python, as seguintes práticas **NÃO** serão aceitas, pois não são formas de programação estruturada:
 - ▣ Comando **else** como alternativa aos comandos **while** e **for**.
 - ▣ Comando **break**.
 - ▣ Comando **continue**.

Função **range**

- A função **range simplifica** a criação e contagem dos elementos de uma lista, pois gera uma sequência de números inteiros para serem usados por um laço do tipo **for**.
- A função **range** pode ser usada de três maneiras:
 - **Um** argumento
 - **Dois** argumentos
 - **Três** argumentos

Função `range`

```
for i in range(5):  
    print(i)
```

0, 1, 2, 3, 4

Com **01** argumento, a sequência começa em **0**. O argumento é o primeiro valor não incluído na sequência.

Com **02** argumentos, a sequência **começa** com o primeiro argumento, mas termina **antes** do segundo.

```
for i in range(1, 5):  
    print(i)
```

1, 2, 3, 4

```
for i in range(0, 5, 1):  
    print(i)
```

0, 1, 2, 3, 4

Com **03** argumentos, o terceiro argumento é o **passo**.



Função **range**

:: Outros exemplos

```
for i in range(11, 16):  
    print(i)
```

11, 12, 13, 14, 15

```
for i in range(0, 9, 2):  
    print(i)
```

0, 2, 4, 6, 8

```
for i in range(5, 0, -1):  
    print(i)
```

5, 4, 3, 2, 1

Conteúdo



Estruturas de Repetição por Contagem



Como carregar valores em uma lista?



Problemas comuns envolvendo listas

Inicialização de valores

- Os elementos de uma lista podem ser especificados de diversas formas:
 - ▣ Corpo do script (manualmente)
 - ▣ Corpo do script (automaticamente)
 - ▣ Teclado

Inicialização de valores

:: Corpo do script (manualmente)

- Desta forma, o programador insere manualmente os valores de cada um dos elementos do vetor.

```
v = [111, 34, 7, -72, 55]
```

Inicialização de valores

:: Corpo do script (automaticamente)

- Desta forma, o programador usa alguma rotina para implementar alguma **lei de formação**:

```
lst1 = list(range(8))
```

```
[0, 1, 2, 3, 4, 5, 6, 7]
```

```
lst2 = list(range(11,80,11))
```

```
[11, 22, 33, 44, 55, 66, 77]
```

```
lst3 = list(range(6,-7,-2))
```

```
[6, 4, 2, 0, -2, -4, -6]
```



Inicialização de valores

:: Valores inteiros aleatórios

- Como criar uma lista de cinco valores inteiros aleatórios?

```
from random import *  
a = []  
for i in range(5):  
    a.append(randint(100,999))
```

```
[956, 827, 219, 898, 265]
```



Inicialização de valores

:: Valores reais no intervalo unitário

- Como criar uma lista de valores reais entre 0 e 1?

```
a = list(range(11))
for i in a:
    a[i] = a[i]/10.0
```

```
a = []
for i in range(11):
    a.append(i/10.0)
```

```
[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
```

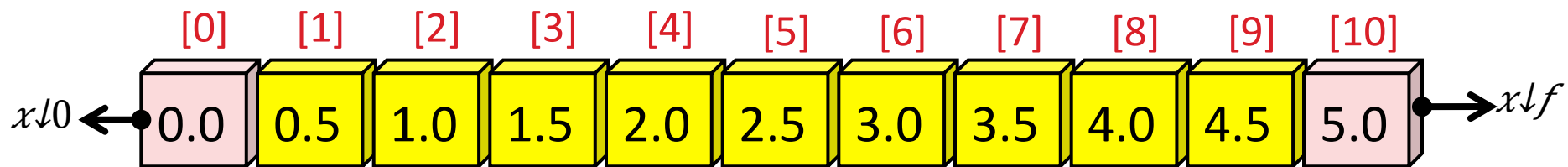


Inicialização de valores

:: Valores uniformemente espaçados

- Como criar uma lista \mathbf{x} de \mathbf{n} valores reais uniformemente espaçados em um intervalo fechado $[x\downarrow 0 ; x\downarrow f]$ qualquer?

▣ **Exemplo:** Criar **11** elementos no intervalo $[0;5]$.



$$\mathbf{x}[i] = \mathbf{x}0 + i * (\mathbf{x}f - \mathbf{x}0) / (\mathbf{n} - 1)$$

largura do
intervalo

tamanho
da lista - 1

Inicialização de valores

:: Por quantidade de elementos

- **Exemplo:** criar **11** elementos no intervalo $[-2; 2]$:

```
n = 11      # Numero de termos
x0 = -2.0   # Inicio do intervalo
xf = 2.0    # Fim do intervalo

x = []
for i in range(n):
    x.append(x0 + i * (xf - x0) / (n - 1))
```



Inicialização de valores

:: Por espaçamento entre elementos

- **Exemplo:** criar elementos no intervalo $[-2; 2]$:
espaçados de 0,25

```
x0      = -2.0    # Inicio do intervalo
xf      = 2.0    # Fim do intervalo
passo   = 0.25   # Espacamento

x = [x0]         # Lista começa com x0
i = 0

while (x[i] < xf):
    i = i + 1    # Proxima posicao da lista
    x.append(x0 + i * passo)
```



Inicialização de valores

:: Teclado

- Para ler os valores de elemento a elemento inseridos via teclado, devemos:
 1. Criar uma lista vazia
 2. Ler valor do elemento
 3. Anexar elemento à lista
 4. Repetir 2 e 3 enquanto uma condição for satisfeita

```
lst = []          # Cria lista vazia
print("Digite os valores ou -1 para sair: ")
elemento = int(input(""))
while (elemento != -1):
    lst.append(elemento) # Anexa elemento
    elemento = int(input(""))
```



Conteúdo



Estruturas de Repetição por Contagem



Como carregar valores em uma lista?

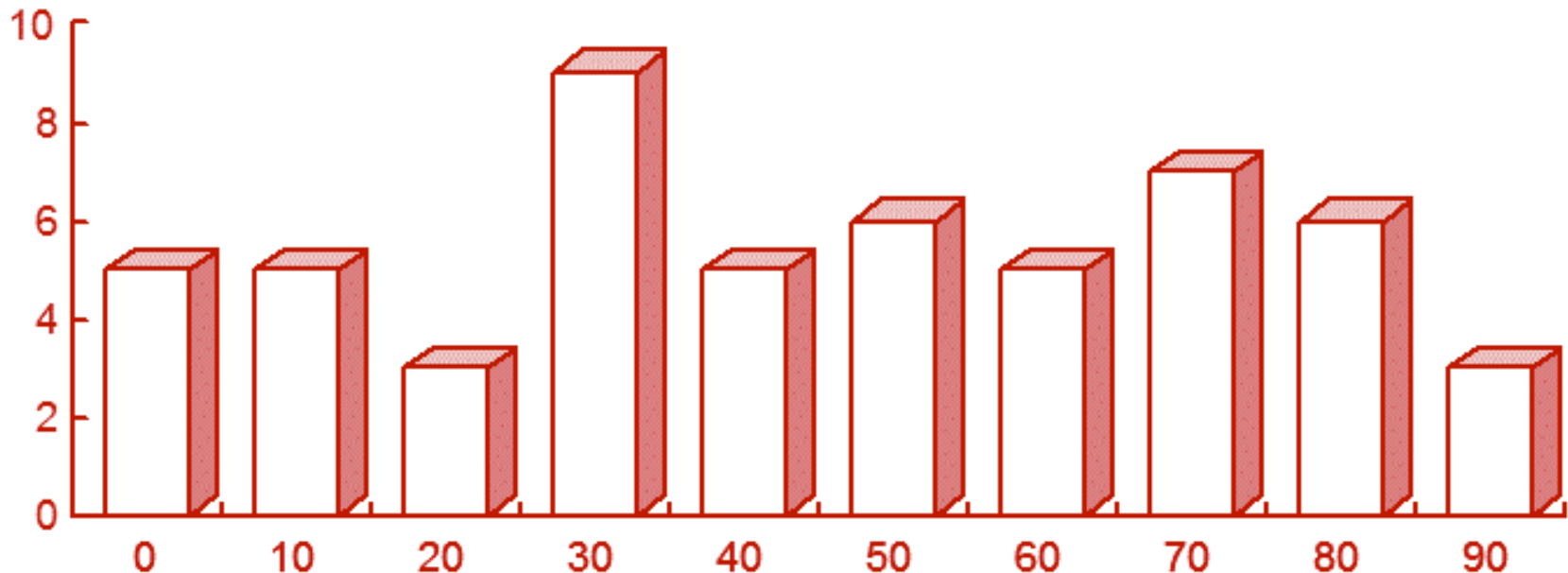


Problemas comuns envolvendo listas

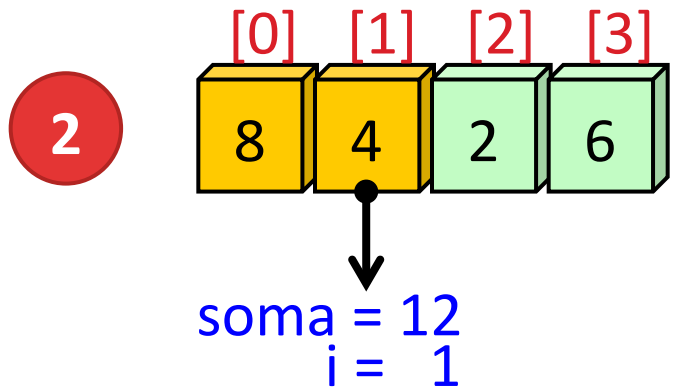
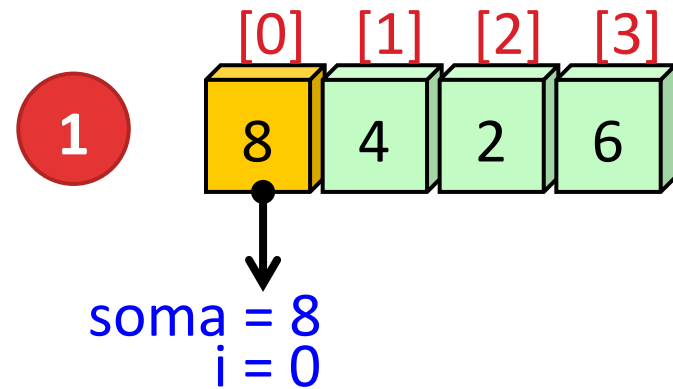
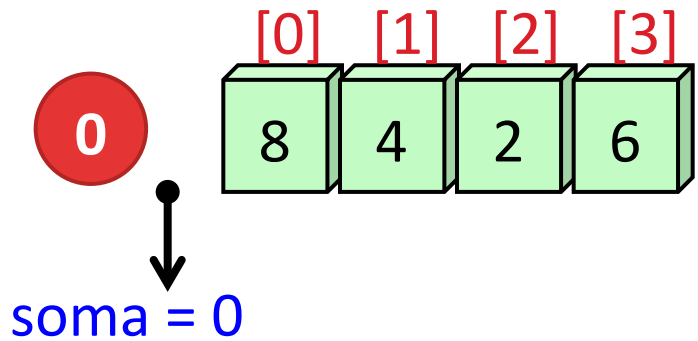
Problema 01

:: Cálculo da Média

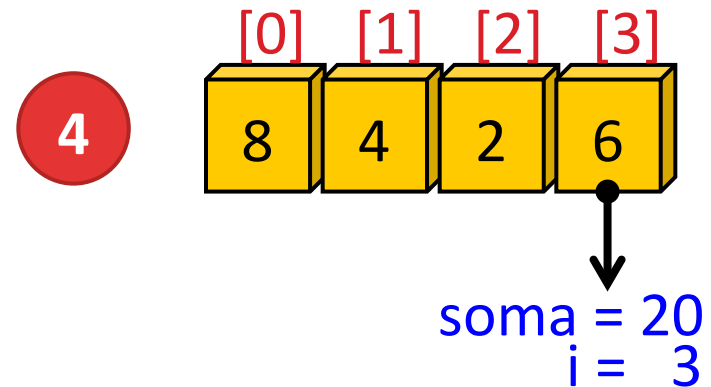
- As notas das provas aplicadas a uma classe estão armazenadas em uma lista.
- Como calcular a **média** da classe?



Cálculo da Média



...



$$\text{media} = \text{soma} / n$$

Cálculo da Média

:: Script em Python – laço for

```
# Atribui valores a uma lista 'v'
v = [12, 34, -56, 78, 90]

# Soma dos valores da lista. Começa em ZERO.
soma = 0

# Calcula a soma
for i in range(0, len(v)):
    soma = soma + v[i]

# Divide a soma pelo numero de
# elementos da lista, para determinar a media
media = soma / len(v)

print(media)
```



Cálculo da Média

:: Script em Python – laço `while`

```
# Atribui valores a uma lista 'v'
v = [12, 34, -56, 78, 90]

soma = 0      # Iniciar acumulador
i = 0        # Iniciar contador

# Calcula a soma
while (i < len(v)):
    soma = soma + v[i]
    i = i + 1

# Divide a soma pelo numero de
# elementos da lista, para determinar a media
media = soma / len(v)

print(media)
```



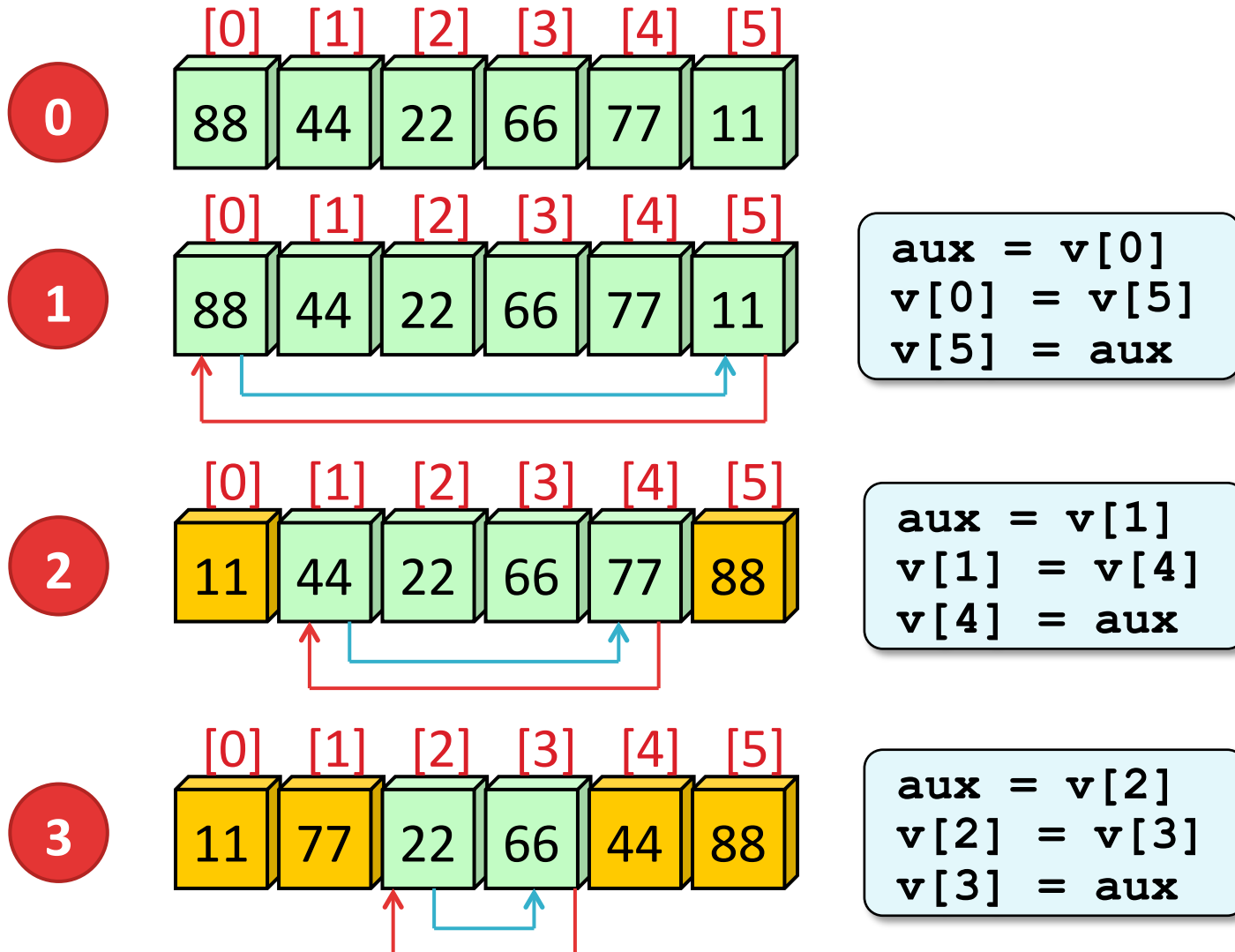
Problema 02

:: Inversão dos elementos de uma lista

- Inverter uma lista, de tamanho qualquer, trocando o 1º elemento com o último, o 2º com o penúltimo, e assim sucessivamente.
- Ao final, imprimir a lista invertida.



Inversão dos elementos de uma lista



Inversão dos elementos de uma lista

:: Projetar algoritmo

- Qual o critério de parada?
 - ▣ Lista com tamanho **par**?
 - ▣ Lista com tamanho **ímpar**?
- Qual os índices **i** e **j** que devem ser trocados em uma iteração qualquer?
 - ▣ Note que **$i + j = \text{len}(v) - 1$**

Inversão dos elementos de uma lista

:: Script em Python – laço `for`

```
# Atribui valores a uma lista 'v'
v = [12, 34, 56, 78, 90]

# Inverte a lista 'v'
for i in range(0, len(v)//2):
    aux = v[i]
    v[i] = v[len(v)-i-1]
    v[len(v)-i-1] = aux

print(v)
```



Inversão dos elementos de uma lista

:: Script em Python – laço **while**

```
# Atribui valores a uma lista 'v'
v = [12, 34, 56, 78, 90]

i = 0      # Iniciar contador

# Inverte a lista 'v'
while (i < len(v)//2):
    aux = v[i]
    v[i] = v[len(v)-i-1]
    v[len(v)-i-1] = aux
    i = i + 1

print(v)
```



Problema 03

:: Contar por condição

- Considere uma lista de números inteiros quaisquer.
- Quantos números **pares** ela contém?



Contar por condição

:: Projetar algoritmo

- Qual o melhor tipo de laço (**while** ou **for**)?
 - ▣ Laço **for**, pois todos os elementos devem ser verificados.
- Quais contadores envolvidos?
 - ▣ **i**: contador do laço **for**
 - ▣ **npar**: conta os números pares pertencentes à lista
- Quando os contadores são atualizados?
 - ▣ **i**: automático dentro do laço **for**
 - ▣ **npar**: somente quando o número lido for par (implica usar uma estrutura **if** dentro do laço **for**)

Contar por condição

:: Script em Python

```
# Cria lista 'a' com valores aleatorios
from random import *
a = []
for i in range(5):
    a.append(randint(100,999))
print(a)

npar = 0      # Zera contador de nos. pares
for i in a:
    if (i % 2 == 0):
        npar = npar + 1

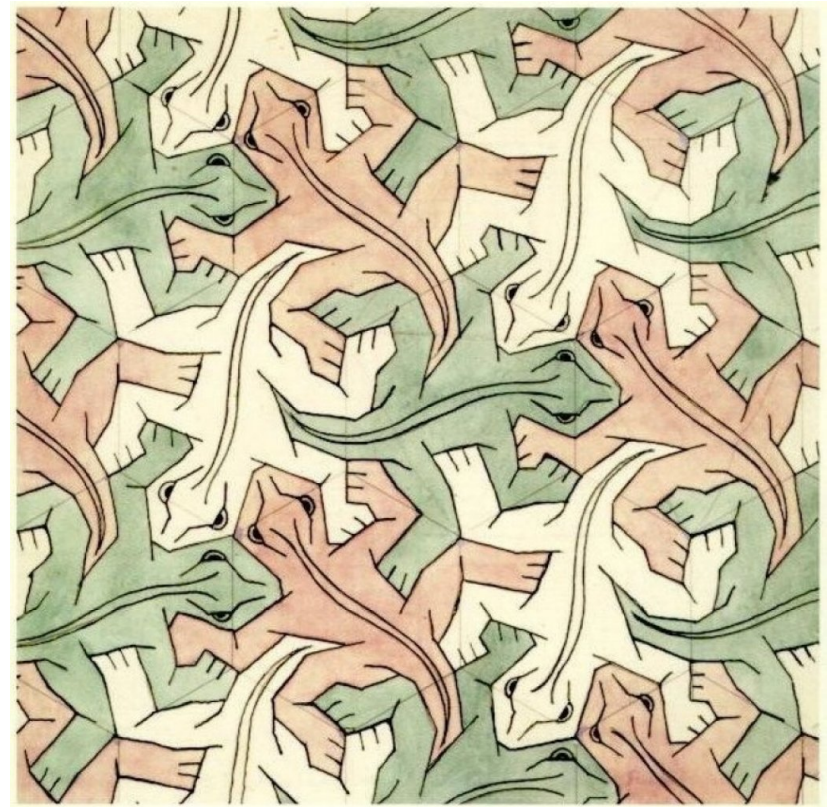
print(npar)
```



Contar por condição

:: Variações

- Contar ou somar:
 - ▣ Elementos de valor par ou ímpar
 - ▣ Elementos em posições pares ou ímpares da lista
 - ▣ Números menores, maiores ou iguais a X



Problema 04

:: Cópia com filtro

- Considere uma lista de números inteiros quaisquer.
- Como criar uma nova lista **removendo os números negativos**?



Cópia com filtro

:: Projetar algoritmo

- Qual o melhor tipo de laço (**while** ou **for**)?
 - ▣ Laço **for**, pois todos os elementos devem ser verificados.
- Quais contadores envolvidos?
 - ▣ **i**: contador do laço **for**
- O que fazer durante o laço?
 - ▣ Ler elemento **i**
 - ▣ Se for positivo ou zero, copiar para nova lista (método **append**)
 - ▣ Caso contrário, não faz nada
 - ▣ Repetir 1, 2 e 3 até acabar o laço **for**

Cópia com filtro

:: Script em Python

```
# Cria lista 'a' com valores aleatorios
from random import *
a = []
for i in range(5):
    a.append(randint(-5,5))
print(a)

newlst = []      # Cria nova lista vazia
for i in a:
    if (i >= 0):
        newlst.append(i)

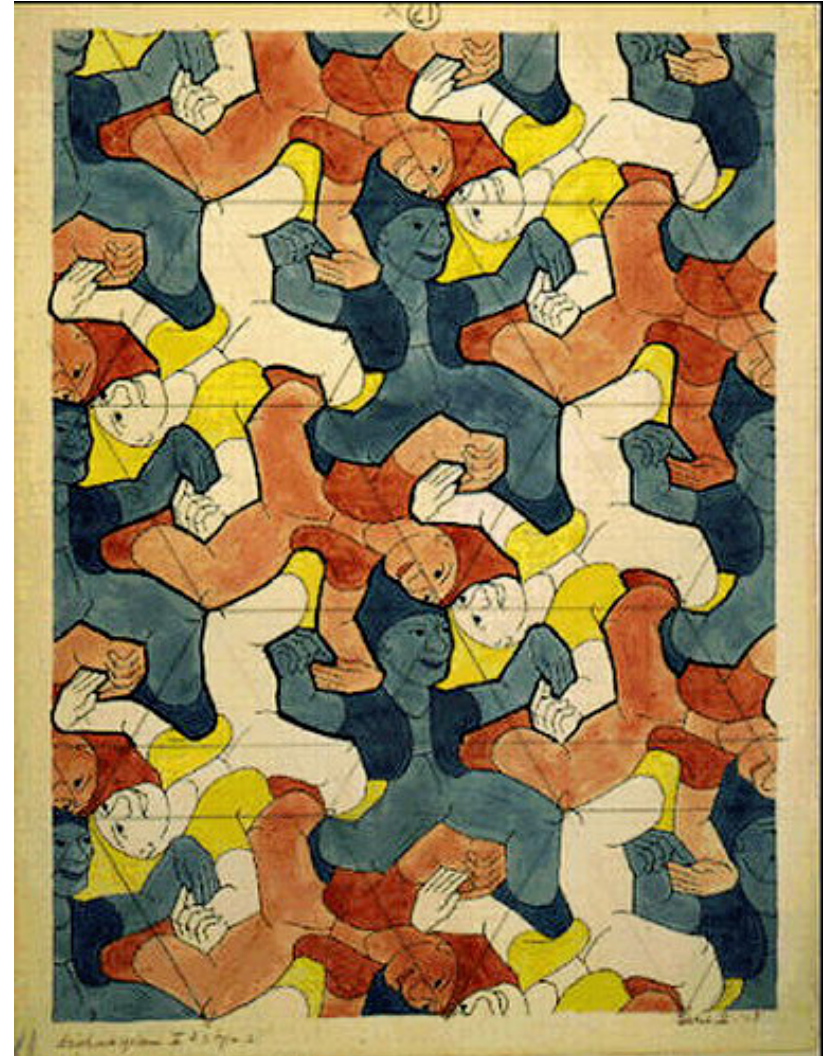
print(newlst)
```



Cópia com filtro

:: Variações

- Copiar removendo:
 - ▣ Elementos de valor par ou ímpar
 - ▣ Elementos em posições pares ou ímpares da lista
 - ▣ Múltiplos de X
 - ▣ Números menores, maiores ou iguais a X



Problema 05

:: Contagem de ocorrências

- Um dado não viciado de 6 faces é lançado **6000 vezes**.
- Quantas vezes ocorre cada face?



Generalização

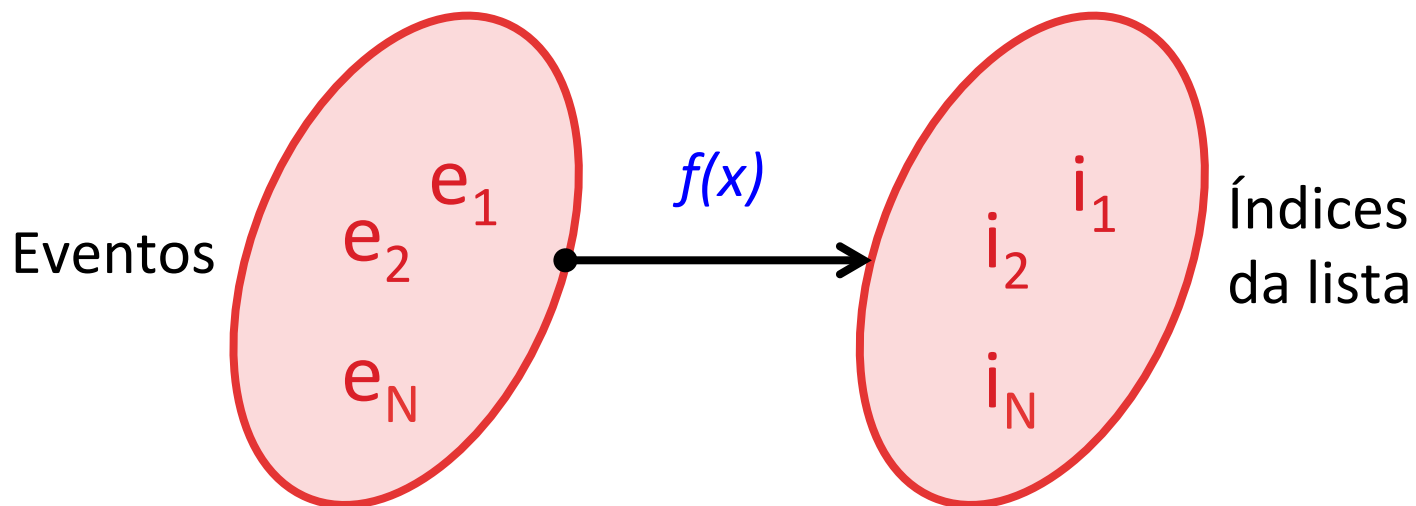
- Dado um conjunto de N eventos $\{e_1, e_2, \dots, e_N\}$, como contar o número de ocorrências de cada evento?
- Nos problemas anteriores, contamos padrões encontrados numa lista. Aqui, ao contrário, a lista será usada para **contar eventos**.



Generalização

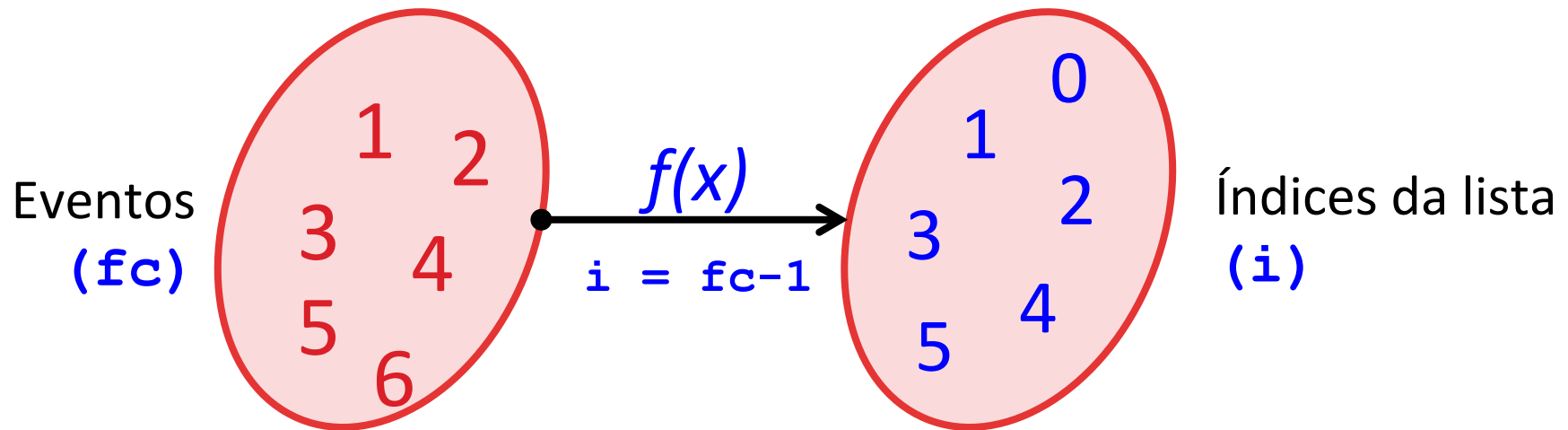
:: Solução

- Criar uma lista de contagem, do tipo **inteiro**, com tamanho igual a **N** (número de eventos possíveis no conjunto).
- **Mapear** os eventos possíveis em índices de uma lista de contagem.



Contagem de ocorrências

:: Projetar algoritmo



- Criar uma lista de contagem **freq**, de $N = 6$ elementos, todos iguais a **zero**.
- Jogar o dado.
- Para cada face **fc** sorteada, incrementar em uma unidade o elemento de índice **fc-1** da lista.

Contagem de ocorrências

:: Script em Python

```
from random import *

# Cria uma lista de 6 elementos contendo zero
freq = [0] * 6

# Contagem de ocorrencias
for i in range(0, 6000):
    # Lançar dado
    face = randint(1,6)
    # Atualizar contador de faces
    freq[face-1] = freq[face-1] + 1

# Imprime vetor de frequencia
print(freq)
```



Problema 06

:: Crivo de Eratóstenes

- Quais os números primos menores que N ?
 - ▣ Iniciamos escrevendo todos os inteiros 2 a N .
 - ▣ Em seguida, eliminamos cada segundo elemento após o 2 , cada terceiro elemento após o 3 , cada quinto elemento após o 5 , e assim por diante.
 - ▣ Terminamos quando todos os múltiplos dos números ímpares inteiros menores que \sqrt{N} tenham sido eliminados.
 - ▣ Os inteiros **restantes** na lista são justamente os números **primos** compreendidos entre 2 e N .

Crivo de Eratóstenes

□ Crivo de Eratóstenes

Peneira de arame
Avaliação minuciosa
segundo certos
critérios



Eratóstenes de Cirene
(276 – 194 a.C.)
Diretor da Biblioteca de
Alexandria



Crivo de Eratóstenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Crivo de Eratóstenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Crivo de Eratóstenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Crivo de Eratóstenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Crivo de Eratóstenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Crivo de Eratóstenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Crivo de Eratóstenes

:: Projetar algoritmo

- Criar lista de inteiros de **2** a **N**.
- Quantos laços e para que servem?
 - ▣ Laço externo (**i**): percorre cada novo inteiro não crivado
 - ▣ Laço interno (**j**): verifica se cada elemento após o inteiro do laço externo é divisível por **i**.
- Qual o melhor tipo de laço (**while** ou **for**)?
 - ▣ Laço **while**, pois a lista vai mudando de tamanho à medida que os números não primos vão sendo removidos.
- Quais contadores envolvidos?
 - ▣ Laço externo (**i**): começa em **0** e vai até \sqrt{N}
 - ▣ Laço interno (**j**): começa em **i+1** e vai até **len(lst)**

Crivo de Eratóstenes

:: Projetar algoritmo

- Qual condição testada para remover um elemento da lista?
 - ▣ No laço interno, se $lst[j]$ (móvel) for divisível por $lst[i]$ (fixo), então $lst[j]$ deve ser removido.
 - ▣ Nesse caso, o contador j deve voltar uma unidade para trás, pois o restante da lista foi deslocada durante a remoção.

Crivo de Eratóstenes

:: Script em Python

```
from math import *

N = 20          # Limite superior
num = list(range(2, N+1))
fim = floor(sqrt(N))
i = 0

while (num[i] <= fim):
    j = i + 1
    while (j < len(num)):
        if (num[j] % num[i] == 0):
            num.pop(j)
            j = j - 1
        j = j + 1
    i = i + 1

print(num)
```



Referências bibliográficas

-  □ Menezes, Nilo Ney Coutinho (2010). **Introdução à Programação com Python**. Editora Novatec.
-  □ HETLAND, Magnus Lie (2008). **Beginning Python: From Novice to Professional**. Springer eBooks, 2ª edição. Disponível em: <http://dx.doi.org/10.1007/978-1-4302-0634-7>.
- Horstmann, Cay & Necaise, Rance D. (2013). **Python for Everyone**. John Wiley & Sons.

Dúvidas?

