

Programação Python

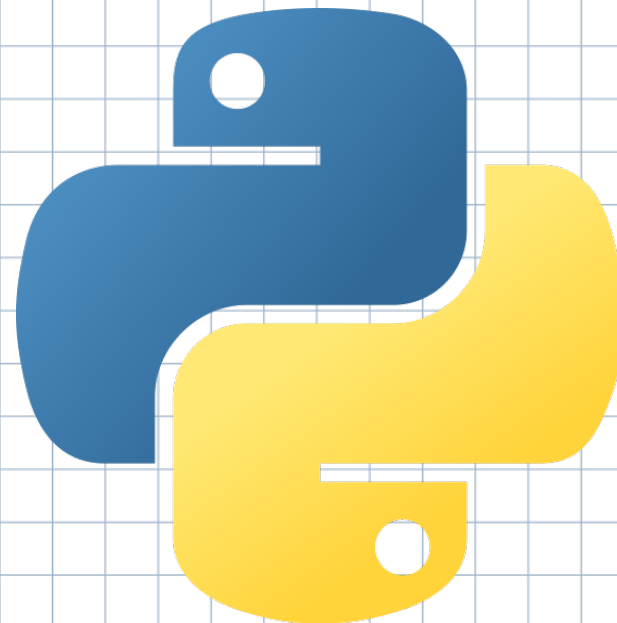
Estruturas de Repetição

Algoritmos Interessantes

A solid blue horizontal bar at the bottom of the slide.

Tópicos:

- ▼ Introdução à computação
- ▼ Lógica de Programação
- ▼ Introdução ao Python
- ▼ Conceitos de Programação em Python
- ▼ Estruturas de Repetição
 - ▶ **Algoritmos Interessantes**
- Estrutura de dados no Python
- ▼ Módulos e Pacotes



Exemplo 1 – Tabuada

```
1 usage
15  def tabuada(x):
16      tabuada = []
17      for i in range(1, 11):
18          tabuada.append(str(x) + ' x ' + str(i) + ' = ' + str(x*i))
19      return tabuada
20
```

Teste - Tabuada

```
21 # Teste tabuada  
22 for x in tabuada(2):  
23     print(x)  
24
```

Exemplo 2 - Fatorial

```
8
9  def fatorial(n):
10      fat = 1
11      for i in range(1, n+1):
12          fat *= i
13      return fat
14
```

Teste - Fatorial

```
27      # Teste Fatorial
28      print(f'\n {32}! = {fatorial(32)}')
29
```

Exemplo 3 - Média

```
1      # Funções matemática
2
3      def media(valores):
4          soma = 0
5          for i in range(0, len(valores)):
6              soma += valores[i]
7          return soma/len(valores)
8
```

Teste - Média

```
30 # Teste Média
31
32 #💡 Lista de valores
33 x = [2, 3, 4, 5, 6]
34
35 # Saída
36 print('A média de ', end='')
37 for valor in x:
38     print(f' {valor} ', end='')
39 print(' é {:.0f}'.format(media(x)))
40
```


Programação Python

Estruturas de Repetição

Recursividade

Exemplo – Fatorial Recursivo

2 usages

```
21 def fat_recursivo(n):
22     if n > 1:
23         fat = n * fat_recursivo(n - 1)
24         return fat
25     else:
26         return 1
27
```

Teste – Fatorial Recursivo

27

28

29

30

```
# Teste Fatorial Recursivo
```

```
print(f'{32}! = {fat_recursivo(32)}')
```

Programação Python

Estrutura de Repetição

Algoritmo de Ordenação

Entrada – Bubble Sort

```
1  # Implementação do algoritmo de ordenação BubbleSorting
2
3  # Apresentação
4  print('\n\t\t\t -- Bubble Sorting -- \n')
5
6  n = []
7
8  # Entrada
9  qtd = int(input('Quantos números você deseja ordenar? '))
10
11 for i in range(0, qtd):
12     n.append(int(input('Informe o {}º valor: '.format((i+1)))))
13
```

Processamento – Bubble Sort

```
14  # Processamento
15  for i in range(0, len(n)-1):
16      for j in range((i+1), qtd):
17          if n[i] > n[j]:
18              swap = n[j]
19              n[j] = n[i]
20              n[i] = swap
21
```

Saída – Bubble Sort

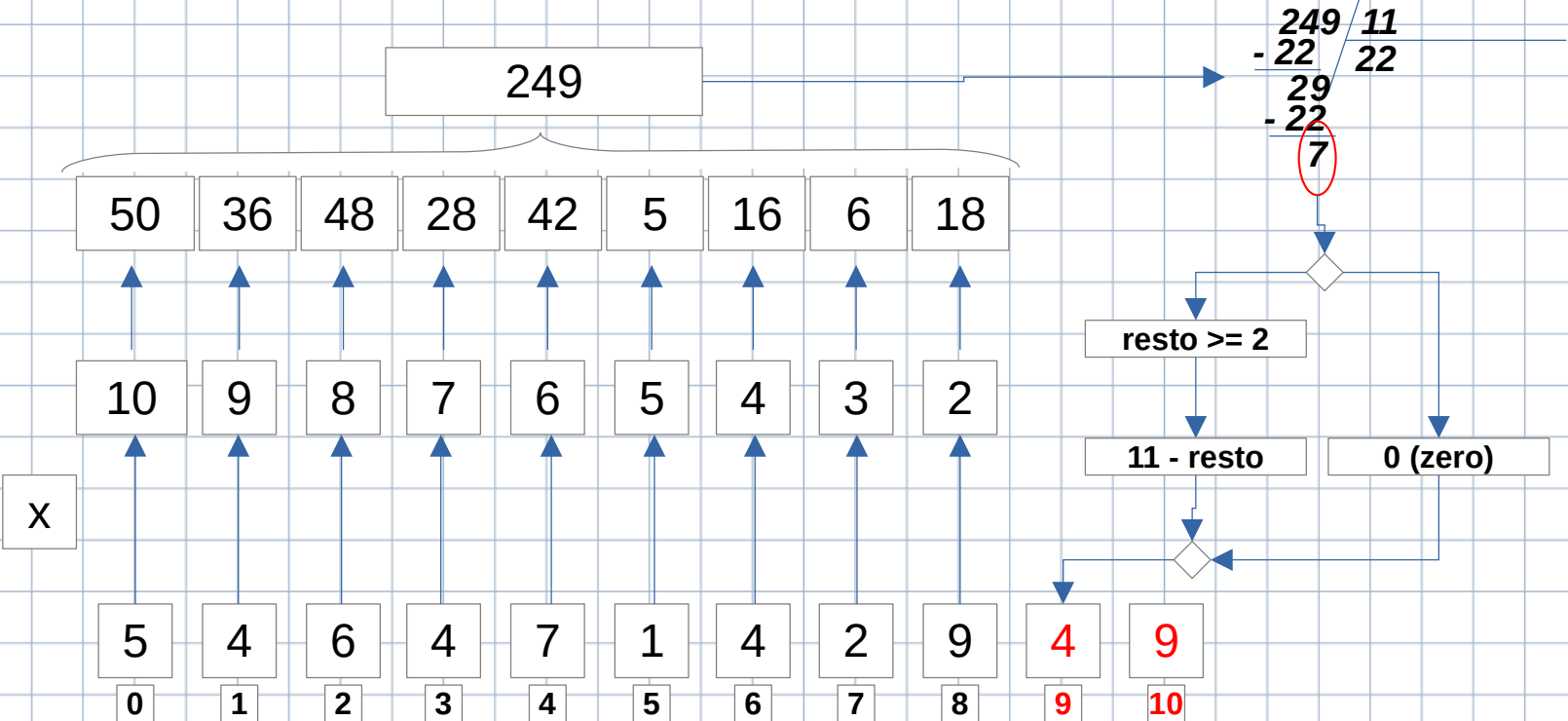
```
22 # Saída
23 for i in range(0, len(n)):
24     if i < len(n)-1:
25         print('{}.'.format(n[i]), end='')
26     else:
27         print('{}.'.format(n[i]), end='')
```

Programação Python

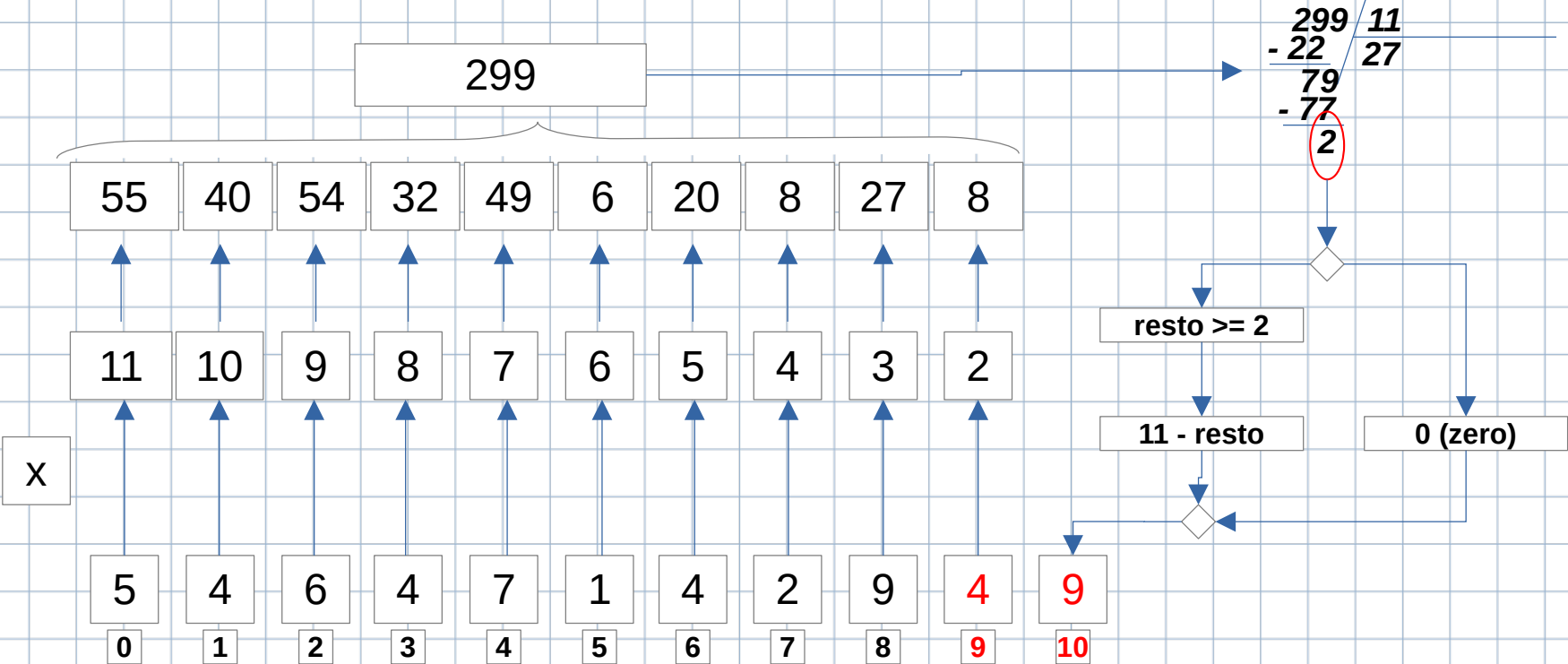
Estrutura de Repetição

Validador de CPF

Primeiro Dígito Verificador



Segundo Dígito Verificador



Validador de CPF

- Entrada
 - CPF Válido
 - 54647142949
 - 17824630706
 - Altere os dígitos para produzir saídas alternativas
- Saída
 - Os CPFs válidos devem ser confirmados
 - Os CPFs alterados deve produzir saídas pertinentes

Processamento - dv1

```

27
4 usages  Gilberto
28 def validadorCPF(cpf):
29     # Variáveis
30     soma = 0
31     cont = 10
32     dv = 0
33
34     if (len(cpf) == 11):
35
36         for i in range(0, 9):
37             soma += int(cpf[i]) * cont
38             cont -= 1
39
40     resto = soma % 11

```

```

40
41
42     resto = soma % 11
43
44     if resto >= 2:
45         dv = 11 - resto
46     else:
47         dv = 0
48
49     if int(cpf[9]) == dv:

```

Processamento - dv2

```

46
47     if int(cpf[9]) == dv:
48
49         # Reinicialização das variáveis
50         cont = 11
51         soma = 0
52
53         for i in range(0, 10):
54             soma += int(cpf[i]) * cont
55             cont -= 1
56
57         resto = soma % 11

```

```

57         resto = soma % 11
58
59         if resto >= 2:
60             dv = 11 - resto
61         else:
62             dv = 0
63
64         if int(cpf[10]) == dv:
65             return True
66         else:
67             return False
68     else:
69         return False
70 else:
71     return False

```

Teste – Validador de CPF

```
73 # Teste Validador de CPF
74 print('CPF válido!') if validadorCPF('54647142949') else print('CPF inválido!')
75 print('CPF válido!') if validadorCPF('54747142949') else print('CPF inválido!')
76 print('CPF válido!') if validadorCPF('17824630706') else print('CPF inválido!')
77 print('CPF válido!') if validadorCPF('17824630806') else print('CPF inválido!')
```