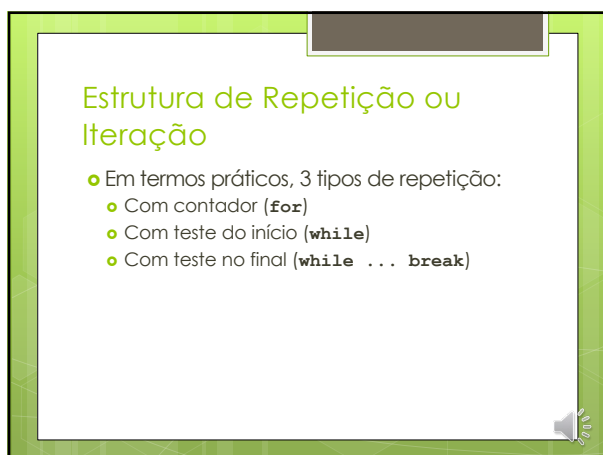


1



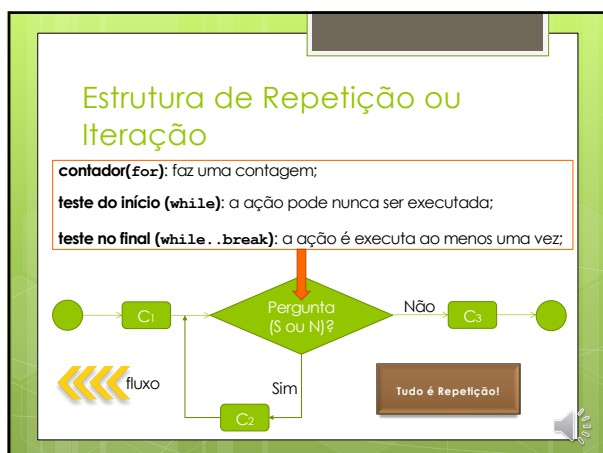
2



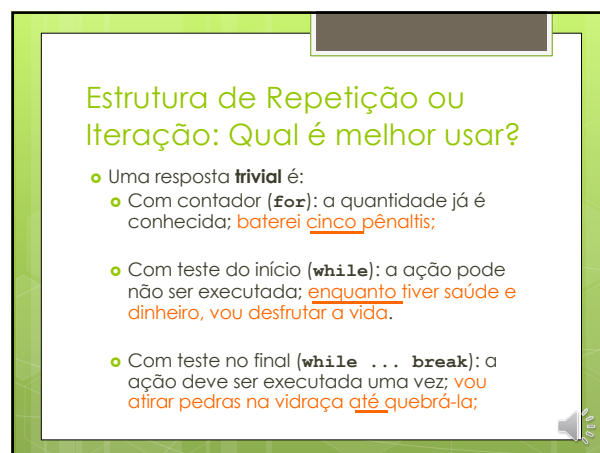
3



4



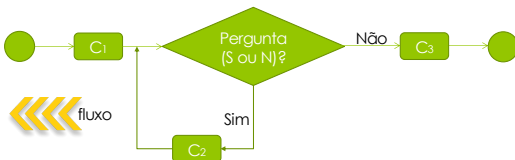
5



6

Estrutura de Repetição ou Iteração

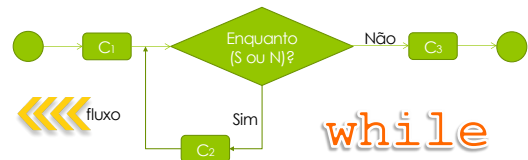
Qual a estrutura de repetição verdadeira? Qual melhor representa o fluxo de uma estrutura repetição?



7

Estrutura de Repetição ou Iteração

Qual a estrutura de repetição verdadeira? Qual melhor representa o fluxo de uma estrutura repetição?



8

Estrutura de Repetição ou Iteração

>>> Escreve seu nome da tela cinco vezes.
Esse é um problema típico para repetição com contador.

```

for contador in range(5):
    print('Nome da pessoa...')

```

✓

```

contador = 0
while contador <= 5:
    print('Nome da pessoa...')
    contador += 1

```

Inicia um contador antes da repetição; O contador faz parte da pergunta; O contador é incrementado dentro da repetição.

9

Estrutura de Repetição ou Iteração

>>> Leia números e imprima o triplo até o usuário digitar zero (flag).
Esse é um problema típico para repetição com teste no final.

```

while True:
    numero = int(input("Número: "))
    if numero != 0: print(numero * 3)
    else: break

```

✓

```

numero = int(input("Número: "))
while numero != 0:
    print(numero * 3)
    numero = int(input("Número: "))

```

Ler uma variável antes da repetição; A variável faz parte da pergunta; A variável é lida novamente dentro da repetição.

10

Situações Comuns

- Contador



11

Situações Comuns

- Contador

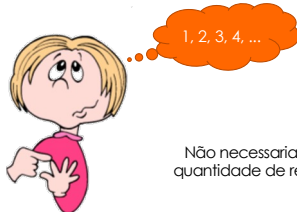
for

MAS NÃO SOMENTE
ASSIM

12

Contador

- Quantas vezes que um evento correu?



Não necessariamente a quantidade de repetições.

13

Contador

- Gere 100 números inteiros aleatórios e informe quantos são múltiplos de 5 (cinco);

```
from random import randrange, seed
seed()

# Inicializa o contador do zero (situação inicial)
qtd = 0

for i in range(100):
    n = randrange(1000)
    if n % 5 == 0: # Se o evento ocorrer
        qtd += 1 # Incrementa o contador

print(f'Foram gerados {qtd} múltiplos de cinco.
```

contadores: i e qtd

14

Contador

- Leia a idade de um número indeterminado de pessoas e mostre quantas são maiores de 65 anos; termine a leitura com o valor -1 para flag.

```
# Inicializa o contador do zero (situação inicial)
maiores_de_65 = 0

while True:
    idade = int(input("Idade ou -1 para encerrar: "))
    if idade != -1 :
        if idade > 65: # Se o evento ocorrer
            maiores_de_65 += 1 # Incrementa o contador
    else:
        break

print(f'{maiores_de_65} pessoa(s) com mais de 65 anos.
```

15

Situações Comuns

- Contador

$+= 1$

Acumula o próprio valor + constante

16

Situações Comuns

- Acumulador

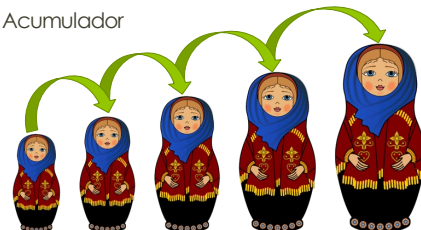


Acumula o próprio valor + variável

17

Situações Comuns

- Acumulador



Acumula o próprio valor + variável

18

Situações Comuns

- Acumulador

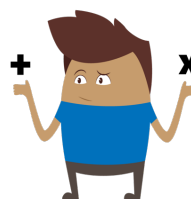
$+=n$ $*=n$
 $-=n$ $/=n$

Acumula o próprio valor + variável

19

Acumulador

- Acumulador da Soma (ou subtração)
- Acumulador do Produto (ou divisão)



20

Acumulador

- Acumulador da Soma

$soma = 0$ Inicializa o acumulador
com o elemento neutro da
soma (zero).
 $soma += n$

21

Acumulador

- Gere 5 números inteiros, positivos e aleatórios e mostre a soma deles; Por exemplo: $(2 + 6 + 8 + 4 + 3 = 23)$

```
from random import randrange, seed
seed()

# Inicializa com zero (elemento neutro da soma)
total = 0

for i in range(5):
    n = randrange(1, 100) # Positivo aleatório
    total += n # Acumula o total da soma

print(f'A soma foi {total}.')
```

contador: i
acumulador: total

22

Acumulador

- Acumulador do Produto

$prod = 1$ Inicializa o acumulador
com o elemento neutro do
produto (um).
 $prod *= n$

23

Acumulador

- Gere 5 números inteiros, positivos e aleatórios e mostre o produto deles; Por exemplo: $(2 \times 6 \times 8 \times 4 \times 3 = 1152)$

```
from random import randrange, seed
seed()

# Inicializa com um (elemento neutro do produto)
total = 1

for i in range(5):
    n = randrange(1, 100) # Positivo aleatório
    total *= n # Acumula o total do produto

print(f'O produto foi {total}.')
```

contador: i
acumulador: total

24

Repetições Aninhadas

Aninhar (ou encaixar) é por uma repetição dentro de outra. Dizemos que uma repetição é **externa** e outra **interna**.

```
for x in range(2):
    for y in range(2):
        for z in range(2):
            print(f'{x}{y}{z}')
```

Cada uma das variáveis **x**, **y** e **z** assumirá os valores binários 0 (zero) e 1 (um). Assim, o código acima, mostra os valores de 0 a 7 em números binários.

```
xyz
000
001
010
011
100
101
110
111
```

25

Repetições Aninhadas

- Quais e quantos são números menores que 1000 (mil) possuem a soma dos dígitos igual a 10?

```
qtd = 0
for i in range(10):
    for j in range(10):
        for k in range(10):
            if i + j + k == 10:
                qtd += 1
                print((i*100)+(j*10)+k)
print(f'Quantidade: {qtd}')
```

Se **i + j + k == 10**, incrementa a quantidade e reconstrói o número para imprimir. No final, mostra a quantidade.

```
ijk
19
28
37
46
...
811
820
901
910
Quantidade: 63
```

26

Repetições Aninhadas

- Leia 5 (cinco) números e mostre a soma dos dígitos de cada número lido.

```
for i in range(5):
    n = aux = int(input('Número (zero finaliza): '))
    soma = 0

    while n > 0:
        soma += n % 10
        n //= 10

    print(f'A soma dos dígitos de {aux} é {soma}.')
```

Cria uma variável auxiliar para guardar o valor original de **n** para mostrar no final.

Acumula o valor de cada dígito em soma até reduzir o **n** à zero.

27