



Engenharia de Software

Computacional thinking with Python

Prof. Dr. Francisco Elânio

Objetivo

- Operadores de atribuição
- Introdução a estrutura de repetição while
- Exercícios

Operadores de Atribuição

<i>Operadores de Atribuição</i>	<i>Exemplo</i>	<i>Equivalência</i>	<i>Descrição</i>
<code>+=</code>	<code>a += b</code>	<code>a = a + b</code>	Adiciona operando do lado direito com operando do lado esquerdo e, em seguida, atribua ao operando da esquerda
<code>-=</code>	<code>a -= b</code>	<code>a = a - b</code>	Subtrair o operando direito do operando esquerdo e, em seguida, atribuir ao operando esquerdo
<code>*=</code>	<code>a *= b</code>	<code>a = a * b</code>	Multiplicar operando direito com operando esquerdo e, em seguida, atribuir ao operando esquerdo
<code>/=</code>	<code>a /= b</code>	<code>a = a / b</code>	Divide o operando esquerdo com o operando direito e, em seguida, atribua ao operando esquerdo
<code>%=</code>	<code>a %= b</code>	<code>a = a % b</code>	Obtém o módulo usando operandos esquerdo e direito e atribui o resultado ao operando esquerdo

Operadores de Atribuição +=

Inicialize uma variável para armazenar a soma.

```
soma = 0
```

```
for numero in range(1, 3):
```

```
    soma += numero
```

```
print("A soma dos números de 1 a 3 é:", soma)
```

Primeira Iteração

numero é igual a 1, e a operação soma +=
numero é equivalente a soma = soma + 1.
Portanto, soma agora é igual a 1.

Segunda Iteração

numero é igual a 2, e a operação soma +=
numero é equivalente a soma = soma + 2.
Como soma era 1 após a primeira iteração,
agora ela é atualizada para 1 + 2,
resultando 3

Operadores de Atribuição -=

Inicialize uma variável para armazenar.

```
sub = 0
```

```
for numero in range(1, 3):
```

```
    sub -= numero
```

```
print("O resultado da subtração dos números da lista será:", sub)
```

Primeira Iteração

Numero é igual a 1, e a operação sub -= numero é equivalente a sub = sub - 1. Como sub começa em 0, após a primeira iteração, sub será -1.

Segunda Iteração

Numero é igual a 2, e a operação soma -= numero é equivalente a sub = sub - 2. Como sub agora é -1 (a partir da primeira iteração), após a segunda iteração, sub será -3.

Operadores de Atribuição *=

Inicialize uma variável para armazenar.

```
mult = 0
```

```
for numero in range(1, 3):
```

```
    mult *= numero
```

```
print("O resultado da subtração dos números da lista será:", mult)
```

Primeira e Segunda Iteração

Como mult é inicializada com zero e não é atualizada antes da multiplicação, o valor de mult permanece zero durante todo o loop.

Operadores de Atribuição /=

Inicialize uma variável para armazenar.

```
div = 1
```

```
for numero in range(1, 3):
```

```
    div /= numero
```

```
print("O resultado da subtração dos números da lista será:", div)
```

Primeira Iteração

numero é igual a 1, e a operação `div /= numero` é equivalente a `div = div / 1`. Como `div` já é igual a 1, a divisão não altera o valor de `div`, e ele permanece como 1.

Segunda Iteração

numero é igual a 2, e a operação `div /= numero` é equivalente a `div = div / 2`. Como `div` é 1 e estamos dividindo por 2, o resultado é 0,5.

Operadores de Atribuição %=

Inicialize uma variável para armazenar.

numero =

if numero % 2 == 0:

print(f"{numero} é um número par.")

else:

print(f"{numero} é um número ímpar.")

Resultado

Como 10 é divisível por 2 sem deixar um resto (10 % 2 = 0).

Exercícios

Exercício 1: Calcule a soma dos números de 1 a 100.

Exercício 2: Crie um programa que imprima quais são os números pares de 1 a 50 e calcule a soma dos valores pares.

Exercício 3: Escreva um programa que identifique quais valores são ímpares em uma range de 1 a 51 e calcule a média destes valores.

Exercícios

Exercício 4: Crie um algoritmo na qual o usuário insira valores, identifique quais números são pares, some e calcule a média dos valores pares escolhidos pelo usuário.

Resolução – Exercício 1

Exercício 1: Calcule a soma dos números de 1 a 100. Mostre o resultado.

```
soma = 0
for numero in range(1, 100):
    soma += numero
print("O resultado será", soma)
```

Resolução – Exercício 2

Exercício 2: Crie um programa que imprima quais são os números pares de 1 a 50 e calcule a soma dos valores pares.

```
numeros_pares = []
```

```
soma = 0
```

```
for numero in range(1, 51):
```

```
    if numero % 2 == 0:
```

```
        numeros_pares.append(numero)
```

```
        soma += numero
```

```
print("Números pares de 1 a 50:", numeros_pares)
```

```
print("A soma dos números pares de 1 a 50 é:", soma)
```

append

Método que pertence a objetos do tipo lista (list). É utilizada para adicionar um elemento ao final da lista.

Resolução – Exercício 3

Exercício 3: Escreva um programa que identifique quais valores são ímpares em uma range de 1 a 51 e calcule a média destes valores.

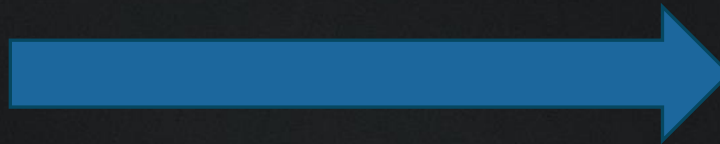
```
numeros_impares = []
soma = 0
for numero in range(1, 51):
    if numero % 2 != 0:
        numeros_impares.append(numero)
        soma += numero
media = soma / len(numeros_impares)
print(f"Números ímpares é {numeros_impares}")
print(f"A soma é {soma}")
print(f"A média entre os valores ímpares é {media}")
```


Estrutura de repetição while

Repetições representam a base de vários programas. São utilizadas para executar a mesma parte de um programa várias vezes, normalmente dependendo de uma condição. Por exemplo, para imprimir três números na tela, poderíamos escrever um programa como:

```
x = 1  
print(x)  
x = x + 1  
print(x)  
x = x + 1  
print(x)
```

Estrutura usando while



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

Contadores

Imagine um problema em que deveríamos imprimir os números inteiros entre 1 e um valor digitado pelo usuário. Vamos escrever um programa de forma que o último número a imprimir seja informado pelo usuário:

```
fim = int(input( "Digite o último número a imprimir: "))  
x = 1
```

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

Imprime os valores de 1
até o valor digitado pelo
usuário

Contadores

Faça um algoritmo para imprimir a tabuada de adição de um número digitado pelo usuário. Essa tabuada deve ser impressa de 1 a 10, sendo no número digitado pelo usuário. Teríamos, $n+1$, $n+2$, ... $n+10$.

```
n = int(input( "Tabuada de:"))  
x = 1  
while x <= 10:  
    print(n + x)  
    x = x + 1
```

Laço de repetição - while

```
valores_pares = []

while True:
    try:
        valor = float(input("Digite um valor (ou digite 0 para parar): "))
        if valor == 0:
            break
        if valor % 2 == 0:
            valores_pares.append(valor)

    except ValueError:
        print("Entrada inválida. Por favor, insira um número válido.")

if valores_pares:
    soma_pares = sum(valores_pares)
    media_pares = soma_pares / len(valores_pares)

    print("Valores pares inseridos:", valores_pares)
    print("Soma dos valores pares:", soma_pares)
    print("Média dos valores pares:", media_pares)
else:
    print("Nenhum valor par foi inserido.")
```


*O importante é não parar
de questionar (Einstein)*

Referências

ASCENCIO, A. F. G, CAMPOS, E. A. V. Fundamentos da Programação de Computadores: algoritmos, Pascal, C/C++ e Java, 2ª Edição, São Paulo: Pearson 2007.

SALVETTI, Dirceu Douglas; BARBOSA, Lisbete Madson. Algoritmos. São Paulo: Pearson, 2004.