

# MAC 115 – Introdução à Ciência da Computação

## Aula 5

---

Nelson Lago

IF noturno – 2023



**Previously on MAC 115...**

# Programando

- 1 algoritmo vs implementação
- 2 entrada de dados → processamento → resultado
  - ▶ Mostra o resultado para o usuário
  - ▶ **Utiliza o resultado como dado para fazer outra coisa**
- 3 Existem *tipos de dados* diferentes em python (*int, float, string, bool...*)
- 4 Expressões são coisas que têm um *valor* (de algum *tipo*)
  - ▶ E podem ser combinadas ou utilizadas como partes de outras expressões



2 + 3 + 7 (int)

2 > 3 **and** 5 > 4 (bool)

# Nomes (variáveis)

- *Nomes* permitem que pensemos mais sobre o problema a ser resolvido e menos sobre as idiossincrasias do computador
- Nomes em geral representam valores que *variam* (basicamente, alguma informação “real” que está em algum lugar na memória do computador)
  - ▶ Como na matemática!
- Por isso, chamamos esses nomes de “variáveis”

## Nomes (variáveis)

$x \leftarrow 5$  (atribuição)

Há um número finito de caracteres no teclado, então fazemos atribuição em python com “=” :

```
x = 5
x = x + 1
x = input("Digite seu nome: ")
```

# Execução condicional

```
n = int(input("Digite um número natural: "))
if n % 2 == 0:
    print("O número", n, "é par!")
print("Ahazei!")
```

- **`n % 2 == 0` → True ou False**

- ▶ Embora possamos ler “se condição”, na verdade python faz “se o valor da expressão é verdadeiro (**True**)”
- ▶ É **como se** python executasse **if** condição == **True**

- **Como ele sabe onde “acaba o efeito” do **if**?**

- ▶ Qualquer quantidade de espaços, desde que seja consistente (4 espaços é o mais comum)

## Execução condicional – **else**

```
n = int(input("Digite um número natural: "))
if n % 2 == 0:
    print("O número", n, "é par!")
else:
    print("O número", n, "é ímpar!")
print("Ahazei!")
```

- A *indentação* indica os “lados” do condicional
  - ▶ Sem variável (“n”), o programa sempre executaria o mesmo “lado”
- O estado da variável só importa no momento do teste
  - ▶ Se ela mudar em seguida, não afeta o condicional
- Os “lados” são mutuamente excludentes
  - ▶ **Um e apenas um** deles é executado

# Por que computação?

O computador é extremamente rápido, mas é uma ferramenta com o mesmo nível de “inteligência” que um martelo

**Não é mais fácil fazer manualmente?**

- **“Algo” precisa acontecer para indicar que as repetições chegaram ao fim**  
*(ok, às vezes queremos repetir indefinidamente, mas vamos ignorar isso por enquanto)*
- **As repetições são controladas por algum tipo de *condição* baseada no estado de uma *variável***

# Partes mínimas de um laço

- **Um laço correto precisa**

- ▶ Inicializar a variável de controle antes do início do laço
- ▶ Verificar a condição adequada a cada iteração para que as repetições aconteçam o número correto de vezes
- ▶ Alterar o valor da variável de acordo com a lógica do programa (no mínimo, na última iteração) para garantir que o laço termine

```
usuarioQuerJogar = True
while usuarioQuerJogar:
    # Joga uma partida...
    resposta = input("Você quer jogar novamente? ")
    if resposta != "S":
        usuarioQuerJogar = False
print("Cabô!")
```

# Tipos de repetição

- **Dois tipos fundamentais de repetição**

- ① Repetições até atingir um resultado

- » *Encontrar o próximo primo*

- » *Reiniciar o jogo até o usuário escolher “sair”*

- » ...

- ② Repetições sobre os elementos de um conjunto

- » *Apresentar todos os pixels de uma foto na tela*

- » *Trocar todas as letras de um texto para maiúsculas*

- » ...

# Tipos de repetição

- **Dois tipos fundamentais de repetição**

- ① **Repetições até atingir um resultado**

- » *Encontrar o próximo primo*
    - » *Reiniciar o jogo até o usuário escolher “sair”*
    - » ...

- ② **Repetições sobre os elementos de um conjunto**

- » *Apresentar todos os pixels de uma foto na tela*
    - » *Trocar todas as letras de um texto para maiúsculas*
    - » ...

**and now for something not different at all**

# Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

# Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**

# Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**

# Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não* é uma expressão

# Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não* é uma expressão
  - ▶ **print()** diz o que fazer com o valor de uma expressão

# Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não* é uma expressão
  - ▶ `print()` diz o que fazer com o valor de uma expressão
  - ▶ O operador = *também* diz o que fazer com o valor de uma expressão

# Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não* é uma expressão
  - ▶ `print()` diz o que fazer com o valor de uma expressão
  - ▶ O operador `=` *também* diz o que fazer com o valor de uma expressão
- **int()** e **float()** são **quase** expressões (são *funções*)

# Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não* é uma expressão
  - ▶ `print()` diz o que fazer com o valor de uma expressão
  - ▶ O operador `=` *também* diz o que fazer com o valor de uma expressão
- **int()** e **float()** são **quase** expressões (são *funções*)
  - ▶ E, portanto, podem fazer parte de uma expressão

# Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não* é uma expressão
  - ▶ `print()` diz o que fazer com o valor de uma expressão
  - ▶ O operador `=` *também* diz o que fazer com o valor de uma expressão
- **int()** e **float()** são **quase** expressões (são *funções*)
  - ▶ E, portanto, podem fazer parte de uma expressão

```
x = 2 + int(3.7)
```

```
print(2 + int(3.7))
```

## Exercícios (os mesmos)

# Exercícios

Dado o lado do quadrado, calcule a área e o perímetro

# Exercícios

Dado o lado do quadrado, calcule a área e o perímetro

```
lado = int(input("Digite o lado do quadrado: "))
```

# Exercícios

Dado o lado do quadrado, calcule a área e o perímetro

```
lado = int(input("Digite o lado do quadrado: "))  
area = lado**2  
perimetro = 4*lado
```

# Exercícios

Dado o lado do quadrado, calcule a área e o perímetro

```
lado = int(input("Digite o lado do quadrado: "))  
area = lado**2  
perimetro = 4*lado  
print("A área do quadrado é", area, "e o perímetro é", perimetro)
```

# Exercícios

Dado o lado do quadrado, calcule a área e o perímetro

# Exercícios

Dado o lado do quadrado, calcule a área e o perímetro

```
lado = int(input("Digite o lado do quadrado: "))
```

# Exercícios

Dado o lado do quadrado, calcule a área e o perímetro

```
lado = int(input("Digite o lado do quadrado: "))  
print("A área do quadrado é", lado**2, "e o perímetro é", 4*lado)
```

# Exercícios

Dado um número, imprima o dígito das dezenas



Dado um número, imprima o dígito das dezenas

```
n = int(input("Digite um número natural: "))
```

Dado um número, imprima o dígito das dezenas

```
n = int(input("Digite um número natural: "))  
n = n % 100
```

Dado um número, imprima o dígito das dezenas

```
n = int(input("Digite um número natural: "))  
n = n % 100  
n = n // 10
```

Dado um número, imprima o dígito das dezenas

```
n = int(input("Digite um número natural: "))  
n = n % 100  
n = n // 10  
print("O dígito das dezenas é", n)
```

Dado um número, imprima o dígito das dezenas

```
n = int(input("Digite um número natural: "))
n = n % 100
n = n // 10

print("O dígito das dezenas é " + str(n))
```

# Exercícios

Dado um número, imprima o dígito das dezenas

```
n = int(input("Digite um número natural: "))
n = n % 100
n = n // 10

print("O dígito das dezenas é_" + str(n))
```

**print()** não acrescenta o espaço neste caso!

Dado um número, imprima o dígito das dezenas

```
n = int(input("Digite um número natural: "))  
print("O dígito das dezenas é", (n % 100) // 10)
```

# Exercícios

Dados três números, verifique se eles estão em ordem crescente



# Exercícios

Dados três números, verifique se eles estão em ordem crescente

```
a = int(input("Digite o primeiro número: "))
```

# Exercícios

Dados três números, verifique se eles estão em ordem crescente

```
a = int(input("Digite o primeiro número: "))  
b = int(input("Digite o segundo número: "))  
c = int(input("Digite o terceiro número: "))
```

# Exercícios

Dados três números, verifique se eles estão em ordem crescente

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))
if
```

# Exercícios

Dados três números, verifique se eles estão em ordem crescente

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))
if a <= b
```

# Exercícios

Dados três números, verifique se eles estão em ordem crescente

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))
if a <= b and
```

# Exercícios

Dados três números, verifique se eles estão em ordem crescente

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))
if a <= b and b <= c:
```

# Exercícios

Dados três números, verifique se eles estão em ordem crescente

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))
if a <= b and b <= c:
    print("Os números estão em ordem crescente")
```

# Exercícios

Dados três números, verifique se eles estão em ordem crescente

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))
if a <= b and b <= c:
    print("Os números estão em ordem crescente")
else:
    print("Os números não estão em ordem crescente")
```

- **O que é um ano bissexto?**

- ▶ O tempo de translação da terra ao redor do sol não é exatamente 365 dias; assim, a cada 4 anos, temos um ano bissexto para compensar essa diferença
- ▶ No entanto, essa compensação não é perfeita; por conta disso, a cada 100 anos, um ano que normalmente seria bissexto não é
- ▶ Essa segunda compensação também não é perfeita e, por isso, a cada 400 anos, um ano que excepcionalmente deixaria de ser bissexto é bissexto normalmente
  - » *Um ano é bissexto se é múltiplo de 4, exceto quando é múltiplo de 100 mas não de 400*
  - » *Um ano é bissexto se é múltiplo de 400 ou se é múltiplo de 4 mas não de 100*
  - » *Um ano é bissexto se é múltiplo de 4 mas não de 100, exceto se for múltiplo de 400*

# Exercícios

Dado um ano, informar se ele é bissexto ou não

# Exercícios

Dado um ano, informar se ele é bissexto ou não

```
ano = int(input("Digite o ano: "))
```

# Exercícios

Dado um ano, informar se ele é bissexto ou não

```
ano = int(input("Digite o ano: "))  
if
```

# Exercícios

Dado um ano, informar se ele é bissexto ou não

```
ano = int(input("Digite o ano: "))  
if ano % 4 == 0
```

# Exercícios

Dado um ano, informar se ele é bissexto ou não

```
ano = int(input("Digite o ano: "))  
if ano % 4 == 0 and ano % 100 != 0
```

# Exercícios

Dado um ano, informar se ele é bissexto ou não

```
ano = int(input("Digite o ano: "))  
if ano % 4 == 0 and ano % 100 != 0 or ano % 400 == 0:
```

# Exercícios

Dado um ano, informar se ele é bissexto ou não

```
ano = int(input("Digite o ano: "))  
if ano % 4 == 0 and ano % 100 != 0 or ano % 400 == 0:  
    print("O ano é bissexto")
```

# Exercícios

Dado um ano, informar se ele é bissexto ou não

```
ano = int(input("Digite o ano: "))
if ano % 4 == 0 and ano % 100 != 0 or ano % 400 == 0:
    print("O ano é bissexto")
else:
    print("O ano não é bissexto")
```

# Exercícios

Dado um ano, informar se ele é bissexto ou não

```
ano = int(input("Digite o ano: "))
if (ano % 4 == 0 and ano % 100 != 0) or ano % 400 == 0:
    print("O ano é bissexto")
else:
    print("O ano não é bissexto")
```

# Exercício

Cálculo do fatorial de um número



# Exercício

## Cálculo do fatorial de um número

```
n = int(input("Digite um inteiro positivo: "))
```

# Exercício

## Cálculo do fatorial de um número

```
n = int(input("Digite um inteiro positivo: "))  
fatorial = n
```

# Exercício

## Cálculo do fatorial de um número

```
n = int(input("Digite um inteiro positivo: "))
fatorial = n
while (n > 1):
```

# Exercício

## Cálculo do fatorial de um número

```
n = int(input("Digite um inteiro positivo: "))
fatorial = n
while (n > 1): # Ou será >= 1 ?
```

# Exercício

## Cálculo do fatorial de um número

```
n = int(input("Digite um inteiro positivo: "))
fatorial = n
while (n > 1): # Ou será >= 1 ?
    n = n - 1
```

# Exercício

## Cálculo do fatorial de um número

```
n = int(input("Digite um inteiro positivo: "))
fatorial = n
while (n > 1): # Ou será >= 1 ?
    n = n - 1
    fatorial = fatorial * n
```

## Cálculo do fatorial de um número

```
n = int(input("Digite um inteiro positivo: "))
fatorial = n
while (n > 1): # Ou será >= 1 ?
    n = n - 1
    fatorial = fatorial * n
print(fatorial)
```

## Cálculo do fatorial de um número

```
n = int(input("Digite um inteiro positivo: "))
fatorial = n
while (n > 2):
    n = n - 1
    fatorial = fatorial * n
print(fatorial)
```

## Cálculo do fatorial de um número

```
n = int(input("Digite um inteiro positivo: "))
fatorial = n
while (n > 2): # Número mágico
    n = n - 1
    fatorial = fatorial * n
print(fatorial)
```

## Exercício

Cálculo do fatorial de um número (usando o elemento neutro)



# Exercício

Cálculo do fatorial de um número (usando o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))
```

# Exercício

Cálculo do fatorial de um número (usando o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))  
fatorial = 1
```

# Exercício

Cálculo do fatorial de um número (usando o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))
fatorial = 1
while (n > 1):
```

# Exercício

Cálculo do fatorial de um número (usando o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))
fatorial = 1
while (n > 1): # Ou será >= 1 ?
```

# Exercício

Cálculo do fatorial de um número (usando o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))
fatorial = 1
while (n > 1): # Ou será >= 1 ?
    fatorial = fatorial * n
```

# Exercício

Cálculo do fatorial de um número (usando o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))
fatorial = 1
while (n > 1): # Ou será >= 1 ?
    fatorial = fatorial * n
    n = n - 1
```

# Exercício

Cálculo do fatorial de um número (usando o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))
fatorial = 1
while (n > 1): # Ou será >= 1 ?
    fatorial = fatorial * n
    n = n - 1
print(fatorial)
```

# Exercício

Cálculo do fatorial de um número (usando o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))
fatorial = 1
while (n > 1): # Ou será >= 1 ?
    fatorial = fatorial * n
    n = n - 1
print(fatorial)
```

É mais comum usar o valor da variável recebido  
no início do laço e atualizar seu valor no final

(“principle of least surprise”)

# Exercícios (novos)

## Exercício

Dado um número inteiro positivo, calcular a soma dos  $n$  primeiros inteiros

## Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros

```
n = int(input("Digite um inteiro positivo: "))
```

# Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros

```
n = int(input("Digite um inteiro positivo: "))
soma = n*(1+n) // 2
print("A soma dos", n, "primeiros inteiros é", soma)
```

## Exercício

Dado um número inteiro positivo, calcular a soma dos  $n$  primeiros inteiros (usando um laço — com o elemento neutro)



## Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros (usando um laço – com o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))
```

## Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros (usando um laço — com o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))  
soma = 0
```

## Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros (usando um laço — com o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))
soma = 0
while n > 0:
```

# Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros (usando um laço — com o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))
soma = 0
while n > 0:
    soma = soma + n
```

# Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros (usando um laço — com o elemento neutro)

```
n = int(input("Digite um inteiro positivo: "))
soma = 0
while n > 0:
    soma = soma + n
    n = n - 1
print("A soma dos", n, "primeiros inteiros é", soma)
```

## Exercício

Dado um número inteiro positivo, calcular a soma dos  $n$  primeiros inteiros (usando um laço — com um contador)



## Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros (usando um laço — com um contador)

```
n = int(input("Digite um inteiro positivo: "))
```

## Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros (usando um laço — com um contador)

```
n = int(input("Digite um inteiro positivo: "))  
soma = 0
```

## Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros (usando um laço — com um contador)

```
n = int(input("Digite um inteiro positivo: "))  
soma = 0  
i = 1
```

# Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros (usando um laço — com um contador)

```
n = int(input("Digite um inteiro positivo: "))
soma = 0
i = 1
while i <= n:
```

# Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros (usando um laço — com um contador)

```
n = int(input("Digite um inteiro positivo: "))
soma = 0
i = 1
while i <= n:
    soma = soma + i
```

# Exercício

Dado um número inteiro positivo, calcular a soma dos n primeiros inteiros (usando um laço — com um contador)

```
n = int(input("Digite um inteiro positivo: "))
soma = 0
i = 1
while i <= n:
    soma = soma + i
    i = i + 1
print("A soma dos", n, "primeiros inteiros é", soma)
```

## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma



## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
n = int(input("Digite um número (zero para sair): "))
```

## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
n = int(input("Digite um número (zero para sair): "))  
soma = 0
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
n = int(input("Digite um número (zero para sair): "))
soma = 0
while n != 0:
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
n = int(input("Digite um número (zero para sair): "))
soma = 0
while n != 0:
    soma = soma + n
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
n = int(input("Digite um número (zero para sair): "))
soma = 0
while n != 0:
    soma = soma + n
    n = int(input("Digite um número (zero para sair): "))
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
n = int(input("Digite um número (zero para sair): "))
soma = 0
while n != 0:
    soma = soma + n
    n = int(input("Digite um número (zero para sair): "))
print("A soma dos números é " + str(soma))
```

## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma



# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
acabou = False
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
acabou = False
```

```
soma = 0
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
acabou = False
soma = 0
while not acabou:
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
acabou = False
soma = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
acabou = False
soma = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
    if n == 0:
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
acabou = False
soma = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
    if n == 0:
        acabou = True
    else:
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
acabou = False
soma = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
    if n == 0:
        acabou = True
    else:
        soma = soma + n
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
acabou = False
soma = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
    if n == 0:
        acabou = True
    else:
        soma = soma + n
print("A soma dos números é " + str(soma))
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e calcule sua soma

```
acabou = False
soma = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
    soma = soma + n
    if n == 0:
        acabou = True
print("A soma dos números é " + str(soma))
```

## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares



# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
```

```
pares = 0
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
pares = 0
while not acabou:
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
pares = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
pares = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
    if n == 0:
        acabou = True
    else:
        if n % 2 == 0:
            pares = pares + 1
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
pares = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
    if n == 0:
        acabou = True
    else:
        if n % 2 == 0:
            pares = pares + 1
print("Você digitou", pares, "números pares")
```

## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares



# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
```

```
pares = 0
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
pares = 0
while not acabou:
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
pares = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
pares = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
    if n % 2 == 0:
        pares = pares + 1
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
pares = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
    if n % 2 == 0:
        pares = pares + 1
    if n == 0:
        acabou = True
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
acabou = False
pares = 0
while not acabou:
    n = int(input("Digite um número (zero para sair): "))
    if n % 2 == 0:
        pares = pares + 1
    if n == 0:
        acabou = True
print("Você digitou", pares - 1, "números pares")
```

## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares



## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
n = int(input("Digite um número (zero para sair): "))
```

## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
n = int(input("Digite um número (zero para sair): "))  
pares = 0
```

## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
n = int(input("Digite um número (zero para sair): "))
pares = 0
while n != 0:
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
n = int(input("Digite um número (zero para sair): "))
pares = 0
while n != 0:
    if n % 2 == 0:
        pares = pares + 1
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
n = int(input("Digite um número (zero para sair): "))
pares = 0
while n != 0:
    if n % 2 == 0:
        pares = pares + 1
    n = int(input("Digite um número (zero para sair): "))
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares

```
n = int(input("Digite um número (zero para sair): "))
pares = 0
while n != 0:
    if n % 2 == 0:
        pares = pares + 1
    n = int(input("Digite um número (zero para sair): "))
print("Você digitou", pares, "números pares")
```

## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares e quantos são ímpares



## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares e quantos são ímpares

```
n = int(input("Digite um número (zero para sair): "))
```

## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares e quantos são ímpares

```
n = int(input("Digite um número (zero para sair): "))  
pares = 0
```

## Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares e quantos são ímpares

```
n = int(input("Digite um número (zero para sair): "))  
pares = 0  
ímpares = 0
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares e quantos são ímpares

```
n = int(input("Digite um número (zero para sair): "))
pares = 0
ímpares = 0
while n != 0:
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares e quantos são ímpares

```
n = int(input("Digite um número (zero para sair): "))
pares = 0
ímpares = 0
while n != 0:
    if n % 2 == 0:
        pares = pares + 1
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares e quantos são ímpares

```
n = int(input("Digite um número (zero para sair): "))
pares = 0
ímpares = 0
while n != 0:
    if n % 2 == 0:
        pares = pares + 1
    else:
        ímpares = ímpares + 1
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares e quantos são ímpares

```
n = int(input("Digite um número (zero para sair): "))
pares = 0
ímpares = 0
while n != 0:
    if n % 2 == 0:
        pares = pares + 1
    else:
        ímpares = ímpares + 1
    n = int(input("Digite um número (zero para sair): "))
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares e quantos são ímpares

```
n = int(input("Digite um número (zero para sair): "))
pares = 0
ímpares = 0
while n != 0:
    if n % 2 == 0:
        pares = pares + 1
    else:
        ímpares = ímpares + 1
    n = int(input("Digite um número (zero para sair): "))
print("Você digitou", pares, "números pares e", ímpares, "números ímpares")
```

# Exercício

Leia uma série de números terminada por zero fornecida pelo usuário e diga quantos deles são pares e quantos são ímpares

```
n = int(input("Digite um número (zero para sair): "))
pares = 0
todos = 0
while n != 0:
    todos = todos + 1
    if n % 2 == 0:
        pares = pares + 1
    n = int(input("Digite um número (zero para sair): "))
print("Você digitou", pares, "números pares e", todos - pares, "números ímpares")
```