MAC 115 — Introdução à Ciência da Computação

Aula 4

Nelson Lago

IF noturno – 2023

Previously on MAC 115...

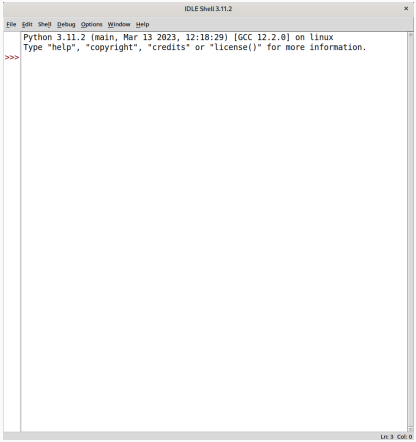
Programando

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

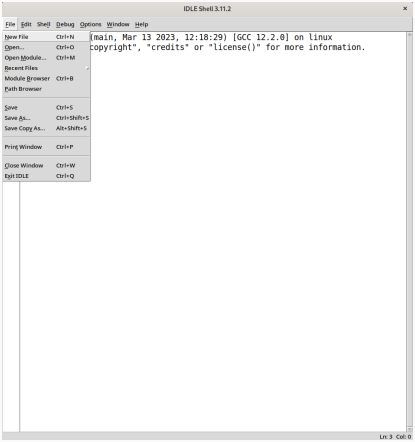
`2 + 3 + 7` (int)

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

IDLE


IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
Ln: 3 Col: 0

VS


IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
New File Ctrl+N
Open... Ctrl+O
Open Module... Ctrl+M
Recent Files
Module Browser Ctrl+B
Path Browser
Save Ctrl+S
Save As... Ctrl+Shift+S
Save Copy As... Alt+Shift+S
Print Window Ctrl+P
Close Window Ctrl+W
Exit IDLE Ctrl+Q
(main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
copyright", "credits" or "license()" for more information.
Ln: 3 Col: 0

Nomes (variáveis)

- **Ao programar, preferimos pensar no problema a ser resolvido e não nas idiossincrasias do computador**
- **Linguagens de programação de alto nível procuram oferecer os recursos para isso**
- **Uma das coisas mais importantes para esse fim é utilizar *nomes***
- **Um dos principais usos de nomes é representar 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 ← 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!")
```

Execução condicional

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

if n % 2 == 0:
    print("O número", n, "é par!")
```

- `n % 2 == 0` → **`True`** ou **`False`**

Execução condicional

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

if n % 2 == 0:
    print("O número", n, "é par!")
```

- **`n % 2 == 0` → `True` ou `False`**
  - Embora possamos ler “se condição”, na verdade python faz “se o valor da expressão é verdadeiro (**`True`**)”

Execução condicional

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

if n % 2 == 0:
    print("O número", n, "é par!")
```

- **`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`

Execução condicional

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

if n % 2 == 0:
    print("O número", n, "é par!")
```

Execução condicional

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

if n % 2 == 0:
    print("O número", n, "é par!")

print("Ahazei!")
```

Execução condicional

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

if n % 2 == 0:
    print("O número", n, "é par!")

print("Ahazei!")
```

- **Ele sempre imprime “Ahazei!” ou só quando o número é par?**

Execução condicional

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

if n % 2 == 0:
    print("O número", n, "é par!")

print("Ahazei!")
```

- Ele sempre imprime “Ahazei!” ou só quando o número é par?
- Como ele sabe onde “acaba o efeito” do `if`?

Execução condicional

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

if n % 2 == 0:
    print("O número", n, "é par!")

print("Ahazei!")
```

- Ele sempre imprime “Ahazei!” ou só quando o número é par?
- Como ele sabe onde “acaba o efeito” do `if`?

Execução condicional

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

if n % 2 == 0:
    print("O número", n, "é par!")

print("Ahazei!")
```

- **Ele sempre imprime “Ahazei!” ou só quando o número é par?**
- **Como ele sabe onde “acaba o efeito” do `if`?**
  - Qualquer quantidade de espaços, desde que seja consistente (4 espaços é o mais comum)

and now for something not that different

Execução condicional

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

Execução condicional

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

- **É muito comum que queiramos "cuidar" de todos os casos possíveis...**

Execução condicional

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

- **É muito comum que queiramos "cuidar" de todos os casos possíveis...**
  - Mas aqui só há dois casos possíveis!

Execução condicional

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

- **É muito comum que queiramos "cuidar" de todos os casos possíveis...**
  - Mas aqui só há dois casos possíveis!
    - » *(e eles são mutuamente excludentes)*

Execução condicional

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

- **É muito comum que queiramos "cuidar" de todos os casos possíveis...**
  - Mas aqui só há dois casos possíveis!
    - » *(e eles são mutuamente excludentes)*
  - Para que verificar a "mesma" condição duas vezes?

Execução condicional

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

- **É muito comum que queiramos "cuidar" de todos os casos possíveis...**
  - Mas aqui só há dois casos possíveis!
    - » *(e eles são mutuamente excludentes)*
  - Para que verificar a "mesma" condição duas vezes?
  - Isso não é muito fácil de ler!

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!")
```

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

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"

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**

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

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**

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

Exercícios

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

Exercícios

Dado um número, imprima o dígito das dezenas

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

Exercícios

Dado um número, imprima o dígito das dezenas

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

Exercícios

Dado um número, imprima o dígito das dezenas

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

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", 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))
```

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!

Exercícios

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")
```

Exercícios

- **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")
```

Exercícios para casa

- **Dado um número de segundos, como 150328, informe o tempo correspondente em dias, horas, minutos e segundos (neste exemplo, "1 dias, 17 horas, 45 minutos e 28 segundos").**
- **Melhore o exemplo de programa que resolve equações de segundo grau para verificar se $\Delta < 0$.**

Repetições

Por que computação?

O computador é extremamente rápido, mas

- **É uma ferramenta com o mesmo nível de "inteligência" que um martelo**
  - Tudo tem que ser esmiuçado nos mínimos detalhes
    - » *"Vá à padaria e compre três pães"*
    - » *"Localize esta palavra no texto"*
    - » *...*

Não é mais fácil fazer manualmente?

Por que computação?

Às vezes, sim 🙂 mas:

- **Sistemas de controle**
- **Comunicação**
- **...**
- **Repetições**

Repetições “externas” e “internas”

- **Algumas repetições são externas ao programa**
  - Calculadora (o usuário faz inúmeros cálculos)
  - Jogo (cada partida é uma “repetição”)
  - ...
- **Mas, em geral, qualquer programa não-trivial vai realizar repetições internamente**
  - Xadrez (cada jogada é uma repetição)
  - Procurar uma palavra em um texto (várias comparações)
  - Apresentar uma foto na tela (cada pixel precisa ser “pintado” com a cor adequada)
  - ...

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*
     - » *...*

Repetições e condições

- **Em ambos os casos, "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***

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*
     - » *...*

Repetições até atingir um resultado

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

Repetições até atingir um resultado

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

- **Sem variável, o programa nunca pararia de repetir**
  - (aqui, a variável é "`usuarioQuerJogar`")
- **A *condição* precisa mudar ao menos na última iteração!**
- **A *condição* testada é "`usuarioQuerJogar` é verdadeiro"**

Repetições até atingir um resultado

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

- **Sem variável, o programa nunca pararia de repetir**
  - (aqui, a variável é "`usuarioQuerJogar`")
- **A *condição* precisa mudar ao menos na última iteração!**
- **A *condição* testada é "`usuarioQuerJogar` é verdadeiro"**
  - » *(Onde está "* `== True` *"?)*

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

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!")
```

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
```

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
print(fatorial)
```

Exercício

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)
```

Exercício

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”)