

# Comandos Python

# Agenda

---

- Variável indicadora
- Variável contadora

# Comandos de Repetição

- Vimos quais são os comandos de repetição em Python.
- Veremos mais alguns exemplos de sua utilização.

```
while condicao:  
    comando(s)
```

```
for variável in  
    lista: comando(s)
```

Variável  
Indicadora

# Variável Indicadora

- Um uso comum de laços é para a verificação se um determinado objeto, ou conjunto de objetos, satisfaz uma propriedade ou não.
- Um padrão que pode ser útil na resolução deste tipo de problema é o uso de uma **variável indicadora**.
  - Supomos que o objeto satisfaz a propriedade (**indicadora = True**).
  - Com um laço verificamos se o objeto realmente satisfaz a propriedade.
  - Se em alguma iteração descobrirmos que o objeto não satisfaz a propriedade, então fazemos **indicadora = False**.

# Exemplo: Número Primo

- Problema: Determinar se um número  $n$  é primo ou não.
- Um número é primo se seus únicos divisores são 1 e ele mesmo.
- Dado um número  $n$  como detectar se este é ou não primo?
  - Leia o número  $n$ .
  - Teste se nenhum dos números entre 2 e  $(n - 1)$  divide  $n$ .
- Lembre-se que o operador % retorna o resto da divisão.
- Portanto  $(a \% b)$  é zero se e somente se  $b$  divide  $a$ .

# Exemplo: Número Primo

- Dado um número  $n$  como detectar se este é ou não primo?
  - Leia o número  $n$ .
  - Faça a variável **indicadora = True**, assumindo que é primo.
  - Teste se nenhum dos números entre 2 e  $(n - 1)$  divide  $n$ .
  - Se o resto da divisão for igual a zero então faça **indicadora = False**. Com isto descobrimos que não é primo.

# Exemplo: Número Primo

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

numero = 2
primo = True # primo é a variável indicadora

while (numero <= n-1) and (primo):
    if (n % numero == 0): # se n é divisível por numero
        primo = False
    numero = numero + 1

if (primo):
    print("É primo.")
else:
    print("Não é primo.")
```



# Exemplo: Número Primo (com `break`)

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

numero = 2
primo = True # primo é a variável indicadora

while (numero <= n-1):
    if (n % numero == 0): # se n é divisível por numero
        primo = False
        break
    numero = numero + 1

if (primo):
    print("É primo.")
else:
    print("Não é primo.")
```

# Exemplo: Sequência ordenada

- Problema: Fazer um programa que lê  $n$  números inteiros do teclado, e no final informa se os números lidos estão ou não em ordem crescente. Se o programa identificar que a sequência de números não é crescente antes de ler os  $n$  números, a leitura deve ser interrompida.
- Usaremos uma variável indicadora na resolução deste problema.

# Exemplo: Sequência ordenada

- Um laço principal será responsável pela leitura dos números.
- Vamos usar duas variáveis, uma que guarda o número lido na iteração atual, e uma que guarda o número lido na iteração anterior.
- Informamos que a sequência de números está ordenada se a condição ( $\text{anterior} < \text{atual}$ ) for válida durante a leitura de todos os números.

```
n = int(input("Digite uma quantidade de números para ser analisada: "))
print("Informe o número: ")

anterior = int(input())

i = 1 # leu um número
ordenado = True # ordenado é a variável indicadora

while (i < n) and (ordenado):
    print("Informe o número: ")
    atual = int(input())
    i = i + 1 # leu mais um número
    if (atual < anterior):
        ordenado = False
    anterior = atual

if (ordenado):
    print("Sequência ordenada.")
else:
    print("Sequência não ordenada.")
```

```
n = int(input("Digite uma quantidade de números para ser analisada: "))
```

```
print("Informe o número: ")
```

```
anterior = int(input())
```

```
ordenado = True # ordenado é a variável indicadora
```

```
for i in range(n-1):
```

```
    print("Informe o número: ")
```

```
    atual = int(input())
```

```
    if atual < anterior :
```

```
        ordenado = False
```

```
        break
```

```
    anterior = atual
```

```
if (ordenado):
```

```
    print("Sequência ordenada.")
```

```
else:
```

```
    print("Sequência não ordenada.")
```

# Variável Contadora

# Variável Contadora

- Considere ainda o uso de laços para a verificação se um determinado objeto, ou conjunto de objetos, satisfaz uma propriedade ou não.
- Um outro padrão que pode ser útil é o uso de uma **variável contadora**.
  - Esperamos que um objeto satisfaça  $x$  vezes uma sub-propriedade. Usamos um laço e uma variável que **conta** o número de vezes que o objeto tem a sub-propriedade satisfeita.

# Variável Contadora

- Considere ainda o uso de laços para a verificação se um determinado objeto, ou conjunto de objetos, satisfaz uma propriedade ou não.
- Um outro padrão que pode ser útil é o uso de uma **variável contadora**.
  - Esperamos que um objeto satisfaça  $x$  vezes uma sub-propriedade. Usamos um laço e uma variável que **conta** o número de vezes que o objeto tem a sub-propriedade satisfeita.
  - Ao terminar o laço, se a variável contadora for igual à  $x$  então o objeto satisfaz a propriedade.



## Exemplo: Número Primo

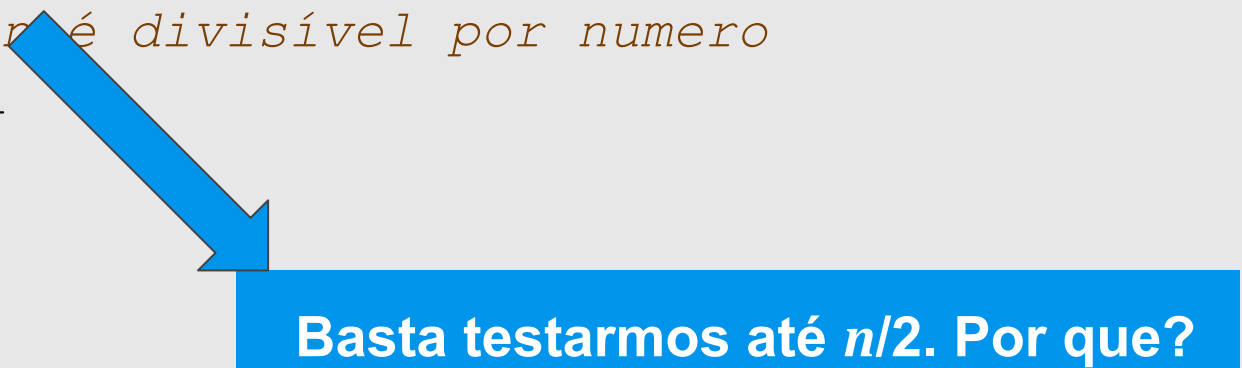
- Problema: Determinar se um número  $n$  é primo ou não.  
**Quando não for, indicar o número de divisores diferentes de 1 e  $n$ .**
- Um número  $n$  é primo se nenhum número de 2 até  $(n - 1)$  dividi-lo de forma inteira.
  - Podemos usar uma variável que **conta** quantos números dividem  $n$  de forma inteira.
  - Se o número de divisores for 0, então  $n$  é primo.

# Exemplo: Número Primo

```
n = int(input("Digite um número inteiro positivo: "))
numero = 2
divisores = 0 # divisores é a variável contadora

while (numero <= n-1):
    if (n % numero == 0): # se n é divisível por numero
        divisores = divisores + 1
    numero = numero + 1

if (divisores == 0):
    print("É primo.")
elif (divisores == 1):
    print("Não é primo. Possui 1 divisor diferente de 1 e",n)
else:
    print("Não é primo. Possui",divisores,"divisores diferentes de 1 e ",n)
```



Basta testarmos até  $n/2$ . Por que?

# Exemplo: Número Primo



```
n = int(input("Digite um número inteiro positivo: "))
if n <= 1:
    print("Não é primo. Números menores ou iguais a 1 não são considerados primos.")
else:
    numero = 2
    divisores = 0 # divisores é a variável contadora

    while (numero <= n-1):
        if (n % numero == 0): # se n é divisível por número
            divisores = divisores + 1
            numero = numero + 1

    if (divisores == 0):
        print("É primo.")
    elif (divisores == 1):
        print("Não é primo. Possui 1 divisor diferente de 1 e",n)
    else:
        print("Não é primo. Possui",divisores,"divisores diferentes de 1 e ",n)
```

# Exemplo: Sequência ordenada

- Problema: Fazer um programa que lê  $n$  números inteiros do teclado, e no final informa se os números lidos estão ou não em ordem crescente. Se o programa identificar que a sequência de números não é crescente antes de ler os  $n$  números, a leitura deve ser interrompida.
- Vamos refazer o programa com uma variável contadora.

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

i = 1 # leu um número
ordenado = True # ordenado é a variável indicadora

while (i < n) and (ordenado):
    atual = int(input())
    i = i + 1 # leu mais um número
    if (atual <= anterior):
        ordenado = False
    anterior = atual

if (ordenado):
    print("Sequência está ordenada.")
else:
    print("Sequência não está ordenada.")
```

## Com variável contadora

```
n = int(input("uma quantidade de números para ser analisada: "))  
print("Informe o número: ")  
anterior = int(input())
```

```
i = 1 # leu um número  
ordenado = 0 # ordenado é a variável contadora
```

```
while (i < n) and (ordenado == 0):  
    print("Informe o número: ")  
    atual = int(input())  
    i = i + 1 # leu mais um número  
    if (atual <= anterior):  
        ordenado = ordenado + 1  
    anterior = atual
```

```
if (ordenado == 0):  
    print("Sequência está ordenada.")  
else:  
    print("Sequência não está ordenada.")
```

# Resumo

- O uso de variáveis **indicadora** e **contadora** são úteis em várias situações.
- Mas não existem fórmulas para a criação de soluções para problemas.
- Em outros problemas, o uso destes padrões pode aparecer em conjunto, ou nem mesmo aparecer como parte da solução.

## Exercício: Número Adjacente

- Fazer um programa que lê  $n$  números inteiros do teclado, e no final informa se os números lidos tem dois dígitos adjacentes iguais.

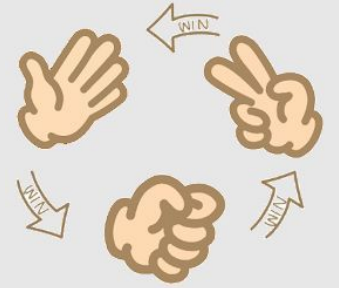
Exemplos:

Para  $n = 5$  números inteiros e 21212, a resposta é não.

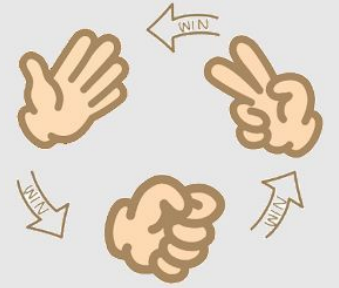
Para  $n = 5$  números inteiros e 21221, a resposta é sim.



# Exercício: Pedra, Papel e Tesoura



- Vamos continuar o programa “Pedra, Papel e Tesoura”.
  - O jogador só pode digitar 0 (pedra), 1 (papel) ou 2 (tesoura). Imprima a mensagem “Opção inválida” se não for nenhuma dessas opções.
  - Vamos jogar novamente? Se “Sim”, recomece o jogo. Se “Não”, encerre o jogo.



# Exercício: Pedra, Papel e Tesoura

- Vamos continuar o programa “Pedra, Papel e Tesoura”.
  - O jogador só pode digitar 0 (pedra), 1 (papel) ou 2 (tesoura). Imprima a mensagem “Opção inválida” se não for nenhuma dessas opções.
  - Vamos jogar novamente? Se “Sim”, recomece o jogo. Se “Não”, encerre o jogo.

# Mais Exercícios =)

- <https://wiki.python.org.br/EstruturaDeRepeticao> : 51 exercícios  
\o/
- Curso de Python:
  - <https://www.codecademy.com/learn/learn-python>

# Créditos

---

Os *slides* deste curso foram baseados nos slides produzidos e cedidos gentilmente pela Professora Sandra Ávila, do Instituto de Computação da Unicamp.