

EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!
EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!



Comando while

Comando `while`

- Executa um bloco de comando(s) enquanto a condição é verdadeira (`True`).

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

Comando `while`

- Programa que imprime os n primeiros números.

```
# Imprime os n primeiros números  
n = int(input("Digite um número: "))  
numero = 1  
while numero <= n:  
    print(numero)  
    numero = numero + 1
```

Comando for

Comando `for`

- É a estrutura de repetição mais usada no Python.
- Para cada elemento da lista, em ordem de ocorrência, é atribuído este elemento à variável e então é executado o(s) comando(s).

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

Comando `for` e a função `range`

- Programa que imprime os n primeiros números.

```
# Imprime os n primeiros números  
n = int(input("Digite um número: "))  
for numero in range(1,n+1):  
    print(numero)
```

while e for

- Programa que imprime os n primeiros números.

```
# Imprime os n primeiros números
n = int(input("Digite um número: "))
numero = 1
while numero <= n:
    print(numero)
    numero = numero + 1
```

```
# Imprime os n primeiros números
n = int(input("Digite um número: "))
for numero in range(1, n+1):
    print(numero)
```

Exercícios

1. Faça um programa que lê um número inteiro positivo n e imprima os valores entre 2 e n , que são divisores de n .
2. Repita o Jogo de Adivinhação dando a opção do jogador de desistir, por exemplo, escolhendo o número 0.

Jogo de Adivinhação

```
import random # módulo random
numero = random.randrange(1, 101) # número entre 1 e 100

palpites = 0
meu_palpite = int(input("Adivinhe meu número entre 1 e 100: "))

while meu_palpite != numero:
    palpites = palpites + 1
    if meu_palpite > numero:
        print(meu_palpite, "está acima.")
    elif meu_palpite < numero:
        print(meu_palpite, "está abaixo.")
    meu_palpite = int(input("tente novamente: "))
print("\nÓtimo, você acertou em", palpites, "tentativas!")
```

```
import random # módulo random
numero = random.randrange(1, 101) # número entre 1 e 100

palpites = 0
meu_palpite = int(input("Adivinhe meu número entre 1 e 100: "))

while meu_palpite != numero:
    if meu_palpite == 0:
        print("Aaah. Você desistiu do jogo.")
        break
    palpites = palpites + 1
    else:
        if meu_palpite > numero:
            print(meu_palpite, "está acima.")
        elif meu_palpite < numero:
            print(meu_palpite, "está abaixo.")
        meu_palpite = int(input("tente novamente: "))

if meu_palpite != 0:
    print("\nÓtimo, você acertou em", palpites, "tentativas!")
```



Algoritmos e Programação de Computadores

Comandos Repetitivos: Variáveis Indicadora e Contadora

Profa. Sandra Avila

Instituto de Computação (IC/Unicamp)

MC102, 22 Março, 2019

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**.
 - Assumimos 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.

Faça um programa que lê um número n e imprima os valores entre 2 e n , que são divisores de n .

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

numero = 2
while numero <= n:
    if (n % numero == 0): # se n é divisível por numero
        print(numero, end=" ")
    numero = numero + 1
```

Fizemos esse programa na aula passada

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

```
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: Números em Ordem Crescente

- 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.
- Usaremos uma variável indicadora na resolução deste problema.

Exemplo: Números em Ordem Crescente

- 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.
- Os números estarão ordenados se a condição ($\text{anterior} \leq \text{atual}$) for válida durante a leitura de todos os números.

```
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.")
```

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.
 - 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.
- Um número n é primo se nenhum número de 2 até $(n - 1)$ dividi-lo.
 - Podemos usar uma variável que **conta** quantos números dividem n .
 - 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.")
else:
    print("Não 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) and (divisores == 0):
    if (n % numero == 0): # se n é divisível por numero
        divisores = divisores + 1
        numero = numero + 1

if (divisores == 0):
    print("É primo.")
else:
    print("Não é primo.")
```



É melhor terminar o laço assim que descobrirmos algum divisor de n.

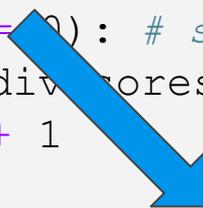
Exemplo: Número Primo

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

numero = 2
divisores = 0 # divisores é a variável contadora

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

if (divisores == 0):
    print("É primo.")
else:
    print("Não é primo.")
```



Basta testarmos até $n/2$. Por que?

Exemplo: Números em Ordem Crescente

- 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.
- 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.")
```

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

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

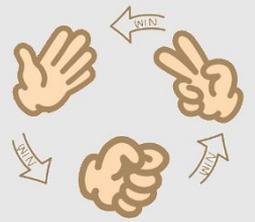
while (i < n) and (ordenado == 0):
    atual = int(input())
    i = i + 1 # leu mais um número
    if (atual < anterior):
        ordenado = ordenado + 1
    anterior = atual

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

Resumo

- O uso de variáveis **acumuladora**, **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.

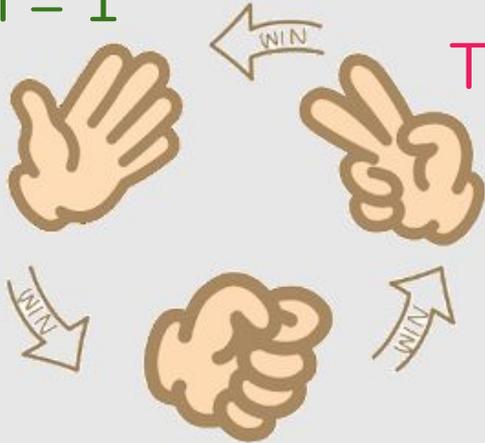
Exercício: Pedra, Papel e Tesoura



- 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

Papel = 1



Tesoura = 2

Pedra = 0

$$(0 - 2) \% 3 = 1 \text{ (Ganhou)}$$

$$(2 - 1) \% 3 = 1 \text{ (Ganhou)}$$

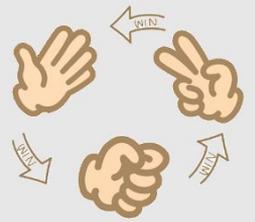
$$(1 - 0) \% 3 = 1 \text{ (Ganhou)}$$

$$(0 - 1) \% 3 = 2$$

$$(1 - 2) \% 3 = 2$$

$$(2 - 0) \% 3 = 2$$

Exercício: Pedra, Papel e Tesoura



```
jogador1 = int(input("Jogador1, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))
jogador2 = int(input("Jogador2, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))

pedra = 0
papel = 1
tesoura = 2

if (jogador1 == jogador2):
    print("Empate! Ninguém ganhou.") # empate
elif (jogador1 - jogador2) % 3 == 1:
    print("Jogador 1 ganhou.")
else:
    print("Jogador 2 ganhou.")
```

Exercício: Número Adjacente

- Faça 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.

Mais Exercícios =)

- <https://wiki.python.org.br/EstruturaDeRepeticao>: 51 exercícios \o/
- Curso de Python:
 - <https://www.codecademy.com/learn/learn-python-3>