

Controlando o fluxo

Python para Sala de Aula de Matemática

Luis Alberto D'Afonseca

CEFET-MG

Conteúdo

Decisões

Repetições

Número Fixo de Repetições

Repetir até Atingir o Objetivo

Tomando Decisões

if teste:

Executa se teste é verdadeiro

elif teste_2:

Executa se teste_2 é verdadeiro

else:

Executa se teste e teste_2 são falsos

Primeiro Exemplo

```
x = 5
```

```
if x == 5:  
    print( "x é 5" )
```

```
else:  
    print( "x não é 5" )
```

Função Tomando Decisão

Verificando se um número é par

```
def par( n ):  
  
    if n % 2 == 0:  
        print( n, 'é par')  
  
    else:  
        print( n, 'não é par')
```

Usando a Função

```
par(2)  
2 é par
```

```
par(3)  
3 não é par
```

Conteúdo

Decisões

Repetições

Número Fixo de Repetições

Repetir até Atingir o Objetivo

Repetições

Muitos algoritmos envolvem repetições

- ▶ Calcular o fatorial de um número
- ▶ Desenhar cada lado de um polígono
- ▶ Imprimir cada letra de um texto

Tipos de Repetições

- ▶ Repetir um número fixo de vezes
- ▶ Repetir até atingir o objetivo

Conteúdo

Decisões

Repetições

Número Fixo de Repetições

Repetir até Atingir o Objetivo

Repetir Para Cada Elemento

```
for variavel in lista_elementos:
```

```
    Comandos a serem repetidos
```

Sequências Numéricas

Progressão aritmética com números inteiros

```
range( início, parada, passo )
```

Sequências Numéricas

`range(10)`

Começa em 0 para *antes* de 10: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

`range(1, 10)`

Começa em 1 para *antes* de 10: 1, 2, 3, 4, 5, 6, 7, 8, 9

`range(1, 10, 2)`

Começa em 1 para *antes* de 10 com passo 2: 1, 3, 5, 7, 9

Primeiro Exemplo

Escrever a tabuada do 7

```
for n in range(1, 11):  
    v = 7 * n  
  
    print( f" 7 x {n:2d} = {v:2d}")
```

Exemplo não Numérico

Listar as vogais do alfabeto

```
vogais = [ "a", "e", "i", "o", "u" ]  
  
for v in vogais:  
    print( v )
```

Função para Calcular $n!$

Fatorial de n

$$f = n! = 1 \times 2 \times 3 \times 4 \times \cdots \times n$$

Algoritmo

dado n

faça $f = 1$

para todo k entre 1 e n

faça $f = f * k$

Em Python

```
def fatorial( n ):  
    f = 1  
  
    for k in range( 1, n+1 ):  
        f = f * k  
  
    return f
```

Conteúdo

Decisões

Repetições

Número Fixo de Repetições

Repetir até Atingir o Objetivo

Repetir Enquanto Teste for Verdadeiro

```
while teste:
```

```
    Comandos a serem repetidos
```

Algoritmo

dado n

faça $f = 1$

enquanto $n > 1$

 faça $f = f * n$

 faça $n = n - 1$

Fatorial Novamente

```
def fatorial_2( n ):
    f = 1

    while n > 1:
        f = f * n
        n = n - 1

    return f
```