

Lógica de programação e algoritmo.

AULA 03

Nessa Aula (04)

- Estruturas de Repetição
 - while
 - for
- Exemplos
- Exercícios

Estruturas de Repetição

- Permitem executar a mesma parte de um programa várias vezes, sob determinadas condições.

Exemplo 01

- Imprimindo de 1 a 3 - forma básica.

```
# -*- coding: UTF-8 -*-
```

```
print(1)  
print(2)  
print(3)
```

Exemplo 02

- Imprimindo de 1 a 3, com variável.

```
# -*- coding: UTF-8 -*-
```

```
x = 1  
print(x)  
x = 2  
print(x)  
x = 3  
print(x)
```

Exemplo 03

- Imprimindo de 1 a 3, com incremento.

```
# -*- coding: UTF-8 -*-
```

```
x = 1  
print(x)  
x = x + 1  
print(x)  
x = x + 1  
print(x)
```

Sintaxe da estrutura de repetição while

```
while <condição>:  
    bloco de código
```

Onde:

condição: condição a ser avaliada.

Exemplo 04

- Imprimindo de 1 a 3, com while.

```
# -*- coding: UTF-8 -*-
```

```
x = 1
while x <= 3:
    print(x)
    x = x + 1
```


Exemplo 05

- Imprimindo de 1 até um número digitado pelo usuário, **com contador**.

```
# -*- coding: UTF-8 -*-
```

```
fim = int(input("Digite o último número a imprimir: "))
```

```
x = 1
```

```
while x <= fim:
```

```
    print(x)
```

```
    x = x + 1
```

Exemplo 06

- Impressão de números pares de 0 até um número digitado pelo usuário, **com if**.

```
# -*- coding: UTF-8 -*-
```

```
fim = int(input("Digite o último número a imprimir: "))
```

```
x = 0
```

```
while x <= fim:
```

```
    if x % 2 == 0:
```

```
        print(x)
```

```
    x = x + 1
```

Exemplo 07

- Impressão de números pares de 0 até um número digitado pelo usuário, **sem if**.

```
# -*- coding: UTF-8 -*-
```

```
fim = int(input("Digite o último número a imprimir: "))
```

```
x = 0
```

```
while x <= fim:
```

```
    print(x)
```

```
    x = x + 2
```

Contadores

- Modifique o programa anterior para imprimir de 1 até o número digitado pelo usuário, mas, dessa vez, apenas os números ímpares.
- Reescreva o programa anterior para escrever os 10 primeiros múltiplos de 3.

Exemplo 08

- Tabuada simples.

```
# -*- coding: UTF-8 -*-
```

```
n = int(input("Tabuada de: "))  
x = 0  
while x <= 10:  
    print (n, "x", x, "=", (n * x))  
    x = x + 1
```

Exemplo 09

- Soma de 10 números, com acumulador.

```
# -*- coding: UTF-8 -*-
```

```
n = 1
soma = 0
while n <= 10:
    x = int(input("Digite o %d número: " % n))
    soma = soma + x
    n = n + 1
print ("Soma: %d" % soma)
```

Exemplo 10

- Cálculo da média de 5 números, com acumulador.

```
# -*- coding: UTF-8 -*-
```

```
x = 1
```

```
soma = 0
```

```
while x <= 5:
```

```
    n = int(input("%d Digite o número: " % x))
```

```
    soma = soma + n
```

```
    x = x + 1
```

```
print ("Média: %5.2f" % (soma/5))
```

Interrompendo a repetição

- A estrutura **while** só verifica a sua condição de parada no início de cada repetição.
- Instrução **break**
 - Interrompe a execução de **while**, independentemente do valor atual de sua condição.

Exemplo 11

- Interrompendo a repetição, com break.

```
# -*- coding: UTF-8 -*-
```

```
s = 0
while True:
    v = int(input("Digite um número a somar ou 0 para sair: "))
    if v == 0:
        break
    s = s + v
print (s)
```

Repetições Aninhadas

- Combinação de vários **while**.
- Repetição com incremento de duas variáveis.

Exemplo 12

- Tabuada aninhada.

```
# -*- coding: UTF-8 -*-
```

```
tabuada = 1
while tabuada <= 10:
    n = 0
    print("Tabuada %d " % tabuada)
    while n <= 10:
        print("%d * %d = %d" % (tabuada, n, tabuada * n))
        n = n + 1
    tabuada = tabuada + 1
```

Sintaxe da estrutura de repetição for

```
for v in range(argumento(s)):  
    bloco de código
```

Onde:

v: um nome de variável.

range(argumento(s)): função para gerar valores com até três inteiros passados a ela.

Exemplo 13

- Uso da função **range(fim)**.

```
# -*- coding: UTF-8 -*-
```

```
# Programa para imprimir 10 valores (de 0 a 9).
```

```
for v in range(10):  
    print(v)
```

Exemplo 14

- Uso da função **range(início, fim)**, com intervalos.

```
# -*- coding: UTF-8 -*-
```

```
# Programa para imprimir os números 5, 6, 7, 8 e 9.
```

```
for v in range(5, 10):  
    print(v)
```

Exemplo 15

- Uso da função `range(início, fim, incremento)`, com saltos.

```
# -*- coding: UTF-8 -*-
```

```
# Programa para imprimir os números 0, 2, 4, 6 e 8.
```

```
for v in range(0, 10, 2):  
    print(v)
```

Exercícios

- 1)** Entrar com números e imprimir o triplo de cada número. O programa acaba quando entrar o número -999.
- 2)** Entrar com números enquanto forem positivos e imprimir quantos números foram digitados.
- 3)** Entrar com vários números positivos e imprimir a média dos números digitados.
- 4)** Ler vários números e informar quantos números entre 100 e 200 foram digitados. Quando o valor 0 (zero) for lido, o programa deve cessar sua execução.
- 5)** Entrar com sexo de várias pessoas e imprimir quantas pessoas são do sexo masculino (considerar m ou M).

Referências

- MENEZES, N. N. C. **Introdução à Programação com Python**: algoritmos e lógica de programação para iniciantes. 2ª ed. São Paulo: Novatec, 2014.
- PYTONSOFTWAREFOUNDATION. **Download the latest version for Windows**: Python 3.6.4. Disponível em: <https://www.python.org/downloads/>. Acesso em: 21 fev. 2018.
- Em homenagem à Janaine Arantes

Referências

Bibliografia Complementar

- MENEZES, N. N. C. **Introdução à Programação com Python**: algoritmos e lógica de programação para iniciantes. 2ª ed. São Paulo: Novatec, 2014.
- SWEIGART, Al. **Automatize Tarefas Maçantes com Python** - Programação Prática para Verdadeiros Iniciantes. São Paulo: Novatec, 2015.
- BORGES, L. E. **Python para Desenvolvedores**. 3ª ed. São Paulo: Novatec, 2014.