

Exercícios resolvidos com **while**

Este documento apresenta exercícios resolvidos que demonstram a utilização do comando de repetição **while**. Digite os exemplos, execute com diferentes entradas e, se for preciso, faça simulações usando lápis e papel para entender claramente a situação das variáveis dentro de cada iteração do laço.

EXERCÍCIO 1: Faça uma simulação em papel e indique qual será valor da variável *s* ao final da execução do programa (não digite e nem rode o programa... faça em papel !!!)

```
x = 1
s = 3
while (x < 8):
    s = s + 5
    x = x * 2
print(s)
```

RESOLUÇÃO:

- Desenhe no papel uma tabela, colocando nas colunas tudo que está no bloco de código do **while**. Nesse caso, são as variáveis *x* e *s* e o teste lógico *x < 8*.
- Preencha a tabela linha por linha.
 - Comece, na primeira linha, colocando a situação das variáveis e do teste lógico até o programa chegar pela primeira vez no **while**.
 - Depois, a cada iteração (repetição), acrescente uma linha para indicar os novos valores que variáveis terão na repetição em questão e qual será o resultado do teste do **while** após o fim da iteração.
 - Faça isso até que não haja mais iteração a ser executada

	x	s	x < 8
Até chegar ao while	1	3	True
Iteração 1	2	8	True
Iteração 2	4	13	True
Iteração 3	8	18	False

O valor de *s* será 18 ao final da execução do programa

EXERCÍCIO 2: A conversão de graus Celsius para graus Fahrenheit é obtida pela seguinte fórmula:

$$F = C * 1,8 + 32$$

Elabore um algoritmo que calcule e imprima uma tabela de equivalência entre graus Celsius e Graus Farenheit, conforme ilustra a figura abaixo. A temperatura inicial em graus Celsius deve ser igual a -20°C e a temperatura final igual a +100°C. A escala da tabela em graus Celsius deve variar de 10 em 10.

-20 °C	-4 °F	
-10 °C	14 °F	
...		...
100 °C	212 °F	

RESOLUÇÃO:

```
c = -20
print('* * * Tabela de conversão de graus Celsius para graus Fahrenheit')
while c <= 100:
    f = c * 1.8 + 32
    print(c, ' °C ----> ', f, ' °F')
    c += 10

print('FIM!!!')
```

```
>>>
* * * Tabela de conversão de graus Celsius para graus Fahrenheit
-20 °C ---->  -4.0 °F
-10 °C ---->  14.0 °F
0 °C ---->   32.0 °F
10 °C ---->   50.0 °F
20 °C ---->   68.0 °F
30 °C ---->   86.0 °F
40 °C ---->  104.0 °F
50 °C ---->  122.0 °F
60 °C ---->  140.0 °F
70 °C ---->  158.0 °F
80 °C ---->  176.0 °F
90 °C ---->  194.0 °F
100 °C ----> 212.0 °F
FIM!!!
```

EXERCÍCIO 3: Elabore um programa onde o usuário possa digitar diversos números reais positivos via teclado. Quando o usuário digitar **-1** a entrada deve encerrar e deve-se imprimir a soma dos números digitados e encerre o programa.

RESOLUÇÃO:

```
print("somando números")
print("=====")
print()

# laço que vai recendo os números e somando
# n é inicializado com 0 apenas para entrarmos ao menos 1x no laço

n = 0          #receberá cada número digitado pelo usuário
soma = 0       #acumulará a soma
while n != -1:
    print('Digite um número real positivo (ou digite -1 para encerrar) ')
    n = float(input())

    if (n > 0):    #atualiza a soma apenas se n for positivo
        soma += n

print('A soma dos números digitados é: ', soma)
```

Obs.: Neste exemplo **-1** é chamado de **flag de saída**. Esse é mais um jargão da área de programação. Significa o valor que faz o laço encerrar (ou seja, que faz o programa sair do laço).

EXERCÍCIO 4: Elabore um programa que leia vários números inteiros e informe quantos números entre 100 e 500 foram digitados. Quando o valor 0 for lido, o programa deverá encerrar a sua execução (dessa vez nosso flag de saída é 0)

RESOLUÇÃO:

```
total = 0 # usada para contabilizar o total de números entre 0 e 100

# pede o 1º número fora do laço
# se o usuário digitar 0, o programa nem entrará no laço
print('Digite um número inteiro ou 0 para sair.')
a = int(input())

while (a != 0):
    if (a >=100) and (a <=500):
        total += 1

    print('Digite um número inteiro ou 0 para sair.')
    a = int(input())

print('total de números entre 100 e 500 = ', total)
```

EXERCÍCIO 5: Entrar com o nome e idade de várias pessoas e imprimir quantas têm idade maior ou igual a 65 anos. O programa deve ser executado enquanto o usuário desejar.

RESOLUÇÃO:

```
ficar = 'S' # usada para controlar o laço enquanto o usuário quiser
total_acima_65 = 0 # contabiliza quantos têm 65 anos ou mais

while (ficar == 'S' or ficar == 's'):
    nome = input('Digite o nome: ')
    idade = int(input('Digite a idade: '))

    if idade >= 65:
        total_acima_65 += 1

    ficar = input('Deseja digitar mais dados (S=Sim, N=Nao)? : ')

print('total de idosos = ', total_acima_65)
```

EXERCÍCIO 6: Dado um país A com 5.000.000 de habitantes e uma taxa de natalidade de 3% ao ano e um país B com 7.000.000 de habitantes e uma taxa de natalidade de 2% ao ano, calcular e imprimir o total de anos necessários para que a população do país A ultrapasse a população do país B.

RESOLUÇÃO:

```
A = 5e6 # população do país A = 5.000.000
B = 7e6 # população do país B = 7.000.000
anos = 0 # número de anos p/ população de A ultrapassar a de B

# Obs.: A e B foram definidos com a notação científica do Python
# 5e6 significa  $5 \times 10^6$  no Python. Da mesma forma, 7e6 é  $7 \times 10^6$ 
# Essa notação sempre gera um valor float
```

```

while (A <= B):
    A = A * 1.03
    B = B * 1.02
    anos += 1

print('A será mais populoso que B dentro de ', anos, ' anos.')

>>>
A será mais populoso que B dentro de 35 anos.

```

EXERCÍCIO 7: Elabore um programa que leia o valor de N como entrada e calcule e imprima o valor de H, sendo:

$$H = 1 + (1 / 2) + (1 / 3) + \dots + (1 / N).$$

Observações:

- Para resolver um exercício de somatório é necessário usar sempre um ou mais comandos de repetição.
- Vocês devem observar os termos da expressão e identificar “**o que**” está variando de termo para termo e “**qual é o comportamento**” desta variação. As coisas que variam sempre devem ser implementadas na forma de variáveis no programa Python.
 - o Exemplos de “coisas” que podem variar nas expressões: numerador, denominador, expoente, sinal (olhem para os termos e notem...).

RESOLUÇÃO:

```

# PASSO 1: recebe o valor de N via teclado
print('Digite o valor de N: ')
N = int(input())

# PASSO 2: calcula H usando um laço
H = 1 # receberá o valor da série H
D = 1 # denominador -> é o cara que varia na série

while (D != N):
    D = D + 1
    H += 1 / D

print('H = ', round(H, 2)) # PASSO 3: imprime o valor calculado de H

```

Esse somatório é bastante simples:

- O numerador é sempre 1 em cada termo.
- O denominador (D) começa em 1 e varia de 1 em 1 em cada termo.
- A série tem N termos. Se N=1, o valor de H é 1. Se N ≠ 1, precisamos montar um LOOP para determinar o valor de H.

EXERCÍCIO 8: Construa um programa que calcule e escreva o valor de S:

$$S = (1/1) + (3/2) + (5/3) + (7/4) + \dots + (99/50).$$

Esse somatório não é complicado também:

- o numerador começa em 1 e termina em 99, variando de 2 em 2
- o denominador começa em 1 e termina em 50 variando de 1 em 1
- veja que a série tem 50 termos (valor fixo - não há entrada no programa).
- Há formas ligeiramente diferentes de fazer. A solução abaixo dá certo.

RESOLUÇÃO:

```
# inicializa as variáveis
S=0; NUM=1; DEN=1

# laço simples que implementa a soma
# veja que estou controlando o denominador com a variável DEN
# e o numerador com a variável NUM
while (DEN <= 50):
    S += (NUM / DEN)
    DEN += 1
    NUM += 2

#imprime o resultado
print('S =',S)

>>>
S = 95.5007946616706
```

EXERCÍCIO 9: Construa um programa que receba o valor de N como entrada e calcule o valor de N! Rejeite números negativos !

RESOLUÇÃO:

```
#lê n
print('CALCULO DO FATORIAL')
print('*****')

# pede o número
n = int(input('Digite n (0 ou positivo): '))

# prende o usuário enquanto ele não digitar 0 ou um valor positivo
while n < 0:
    print('Por favor, não pode ser um número negativo!')
    n = int(input('Digite n (0 ou positivo): '))

# calcula n!
fat = 1
i = 2
while (i <= n):
    fat = fat * i
    i = i + 1

# imprime o resultado
print('n!  =', fat)
```

Observe que quando "n" é zero ou 1, o programa não entra no laço e determina o valor de "fat" como 1, o que é correto, pois $0! = 1$ e $1! = 1$. Em qualquer outro caso o programa entra no laço.

EXERCÍCIO 10: Modifique o programa abaixo, empregando o while para remover as linhas de código repetidas.

```
n = int(input('digite o primeiro número: '))
maior = n; menor = n

n = int(input('digite o segundo número: '))
if n > maior: maior = n
elif n < menor: menor = n

n = int(input('digite o terceiro número: '))
if n > maior: maior = n
elif n < menor: menor = n

n = int(input('digite o quarto número: '))
if n > maior: maior = n
elif n < menor: menor = n

print('O menor número é:', menor)
print('O maior número é:', maior)
```

RESOLUÇÃO:

```
import math

# inicializa "menor" com ∞ e "maior" com -∞
# assim, logo no primeiro número digitado elas certamente serão alteradas
menor = math.inf
maior = -math.inf

k = 0

# recebe os 4 números e atualiza maior e menor quando necessário
while (k < 4):
    print('Digite o numero', k+1, ':')
    n = int(input())
    if n > maior: maior = n # nessa solução preciso de 2 if's independentes
    if n < menor: menor = n #(não dá certo com elif, como eu tinha feito em sala)
    k += 1

#imprime o resultado
print('menor:', menor)
print('maior:', maior)
```

EXERCÍCIO 11: Foi feita uma pesquisa de audiência de canal de TV aberta em várias casas de uma certa cidade, em um determinado dia. Para cada casa visitada, foi anotado o número do canal onde a TV estava ligada (existem apenas sete canais: 2, 4, 6, 7, 9, 11 e 13) e a quantidade de pessoas que estavam assistindo a esse canal (para simplificar, considere que existia apenas um televisor em cada residência). Elabore um programa que:

- Leia os dados desta pesquisa a partir do teclado (ou seja leia pares de valores CANAL e QUANTIDADE DE PESSOAS). Quando o usuário digitar o valor “0” para um canal, a leitura deve ser encerrada.
- Ao final da leitura, o programa deve imprimir o total de pessoas que assistiu a cada um dos 6 canais no dia da pesquisa. Deve imprimir também a porcentagem de audiência de cada canal (ver Figura abaixo “N” e “X” devem ser trocados, respectivamente, pelo número de pessoas e a porcentagem de audiência).

CANAL 2	N ₁ PESSOAS - X ₁ % DE AUDIÊNCIA
...	...
CANAL 13	N ₆ PESSOAS - X ₆ % DE AUDIÊNCIA

OBS: É importante estudar e entender a resolução do problema, pois ele ilustra um exemplo simples de programa para tabular de dados de uma pesquisa.

DICA PARA RESOLUÇÃO DE UM PROBLEMA DESTES TIPO: vocês podem, por exemplo, anotar no papel os passos, as etapas que vocês executariam caso fossem resolver o programa manualmente (ou seja, somando a audiência de cada emissora num caderno ou usando uma calculadora). Depois disso, vocês traduzem o método de resolução “manual” para uma sequência de comandos Python.

RESOLUÇÃO:

```
# -----
#PASSO 1: INICIALIZA AS VARIÁVEIS
# -----

# estas variáveis recebem o total de pessoas que assistiu cada canal
# a sintaxe abaixo é permitida pelo Python para atribuir o mesmo valor a todas
# (nesse caso, valor 0)
c2 = c4 = c6 = c7 = c9 = c11 = c13 = 0

N = 0 #total geral de pessoas contabilizadas pela pesquisa

# -----
#PASSO 2: LOOP QUE LÊ E CONTABILIZA OS DADOS DA PESQUISA
# -----
print('DIGITE OS DADOS DA PESQUISA')
print('*****')

num_canal = 999 #inicializa com um valor qualquer ≠ 0 só para entrar no while

while (num_canal != 0):
    print()
    num_canal = int(input('NUMERO DO CANAL: '))
    if num_canal != 0:
        qtd_pessoas_na_casa = int(input('QUANT. PESSOAS NA CASA: '))

        #atualiza a audiência do canal usando if-elif
        if num_canal == 2:
            c2 += qtd_pessoas_na_casa
            N += qtd_pessoas_na_casa
        elif num_canal == 4:
            c4 += qtd_pessoas_na_casa
            N += qtd_pessoas_na_casa
        elif num_canal == 6:
```

```

        c6 += qtd_pessoas_na_casa
        N += qtd_pessoas_na_casa
    elif num_canal == 7:
        c7 += qtd_pessoas_na_casa
        N += qtd_pessoas_na_casa
    elif num_canal == 9:
        c9 += qtd_pessoas_na_casa
        N += qtd_pessoas_na_casa
    elif num_canal == 11:
        c11 += qtd_pessoas_na_casa
        N += qtd_pessoas_na_casa
    elif num_canal == 13:
        c13 += qtd_pessoas_na_casa
        N += qtd_pessoas_na_casa

# -----
# PASSO 3: IMPRIME OS RESULTADOS
# -----

print()
print('RESULTADOS DA PESQUISA:')
print('*****')
print()

# IMPRIME CANAL POR CANAL (pois ainda não sabemos usar listas...)
# usamos end="" no print do canal,
# para que o próximo print saia do lado e não embaixo

#CANAL 2
pct_audiencia = (c2 / N) * 100
print('- CANAL 2 : ', c2, ' PESSOA(S) - ', end="")
print(round(pct_audiencia,2), '% DE AUDIENCIA')

#CANAL 4
pct_audiencia = (c4 / N) * 100
print('- CANAL 4 : ', c4, ' PESSOA(S) - ', end="")
print(round(pct_audiencia,2), '% DE AUDIENCIA')

#CANAL 6
pct_audiencia = (c6 / N) * 100
print('- CANAL 6 : ', c6, ' PESSOA(S) - ', end="")
print(round(pct_audiencia,2), '% DE AUDIENCIA')

#CANAL 7
pct_audiencia = (c7 / N) * 100
print('- CANAL 7 : ', c7, ' PESSOA(S) - ', end="")
print(round(pct_audiencia,2), '% DE AUDIENCIA')

#CANAL 9
pct_audiencia = (c9 / N) * 100
print('- CANAL 9 : ', c9, ' PESSOA(S) - ', end="")
print(round(pct_audiencia,2), '% DE AUDIENCIA')

#CANAL 11
pct_audiencia = (c11 / N) * 100
print('- CANAL 11 : ', c11, ' PESSOA(S) - ', end="")
print(round(pct_audiencia,2), '% DE AUDIENCIA')

#CANAL 13
pct_audiencia = (c13 / N) * 100
print('- CANAL 13 : ', c13, ' PESSOA(S) - ', end="")
print(round(pct_audiencia,2), '% DE AUDIENCIA')

```

OBS: Quando você aprender a trabalhar com listas (próximo capítulo), verá que é possível resolver esse problema de uma forma muito mais elegante, que utiliza bem menos linhas de código.