

# Estruturas de repetição

---

# Sumário

- Exercício 7 - Estatísticas
- Comando **break**
- Exercício 8 - Loteria
- Estrutura de repetição **while**
- Exercício 9 - Soma com while
- Exercício 10 - Loteria aprimorada

# Exercício 7 - Estatísticas

Faça um programa que receba como entrada **n** números do usuário, um por linha, o programa imprime na tela as seguintes estatísticas sobre os números digitados: maior valor, menor valor e a média dos valores.

ENTRADA:

- O usuário digita quantos valores ele quer inserir

REPITA N VEZES:

ENTRADA:

- O usuário digita um valor

SAÍDA:

- O menor valor, o maior valor e a média dos valores

# Solução

```
n = int(input("Quantos valores: "))

soma = 0
for i in range(0, n):
    valor = float(input("Digite o valor: "))
    if i==0:
        maior = valor
        menor = valor
    else:
        if valor>maior:
            maior = valor
        if valor<menor:
            menor = valor

    soma = soma + valor

media = soma/n

print(f"Maior valor: {maior}")
print(f"Menor valor: {menor}")
print(f"Média dos valores: {media}")
```

# Commando *break*

É possível encerrar uma estrutura de repetição a qualquer momento através do comando ***break***.

No código abaixo, a estrutura **for** será encerrada se o usuário digitar o valor 3.

```
for i in range(0, 10):  
    valor = int(input("Digite um valor: "))  
    print(valor)  
    if valor==3:  
        break
```

# Exercício 8 - Loteria

Faça um programa no qual o usuário terá que adivinhar um número escondido pelo programa. O usuário possui 5 tentativas para adivinhar o número. O programa deve executar as tarefas:

1. No começo do programa, atribua um valor entre 1 e 20 a uma variável chamada `num_oculto`
2. Repita 5 vezes:
  1. Solicitar que o usuário digite um número
  2. Se o número digitado for igual a `num_oculto`, o programa imprime uma mensagem de sucesso e termina
  3. Se o número digitado não for igual a `num_oculto`, o programa informa que o número está errado

Para criar o número oculto usando um número aleatório , use o código:

```
import random  
num_oculto = random.randint(1, 20)
```

# Solução

```
import random
num_oculto = random.randint(1, 20)

print("Seja bem vindo(a)!")
print("Estou pensando em um valor entre 1 e 20. Tente adivinhar o valor.")
print("Você possui 5 tentativas!")

for i in range(0, 5):
    valor = int(input(f"Tentativa {i+1}/5: "))
    if valor==num_oculto:
        print("Parabéns! Você ganhou R$50,00!")
        break
    else:
        # \n adicionado para separar diferentes tentativas
        print("Não foi dessa vez!\n")
```

# Exercício 8 - Loteria

Como imprimir uma mensagem caso o usuário não acerte nenhum número?

Definimos uma variável para contar o número de erros. Também é útil reorganizarmos o código para imprimir a mensagem de sucesso apenas ao final. (código no próximo slide)



# Solução 1

```
import random
num_oculto = random.randint(1, 20)

TENTATIVAS = 5

print("Seja bem vindo(a)!")
print("Estou pensando em um valor entre 1 e 20. Tente adivinhar o valor.")
print("Você possui 5 tentativas!")

contador = 0
for i in range(0, TENTATIVAS):
    valor = int(input(f"Tentativa {i+1}/5: "))
    if valor==num_oculto:
        break
    else:
        # \n adicionado para separar diferentes tentativas
        print("Não foi dessa vez!\n")
        contador += 1

if contador<TENTATIVAS:
    print("Parabéns! Você ganhou R$50,00!")
else:
    print("Não há mais tentativas. Mais sorte da próxima vez!")
```

# Solução 2

Podemos também utilizar uma variável booleana ao invés de um contador:

```
...
sucesso = False
for i in range(0, TENTATIVAS):
    valor = int(input(f"Tentativa {i+1}/5: "))
    if valor==num_oculto:
        sucesso= True
        break
    else:
        # \n adicionado para separar diferentes tentativas
        print("Não foi dessa vez!\n")

if sucesso:
    print("Parabéns! Você ganhou R$50,00!")
else:
    print("Não há mais tentativas. Mais sorte da próxima vez!")
```

## Estrutura de repetição *while*

---

# Estrutura de repetição *while*

Nos exercícios anteriores, foi solicitado que o usuário digitasse o número de valores que ele pretende inserir

Como podemos fazer para permitir que o usuário digite valores até que ele deseje encerrar o programa?

# Estrutura de repetição *while*

Para casos nos quais queremos repetir uma sequência de comandos até que uma condição seja satisfeita, usamos a estrutura de repetição *while*

# Estrutura de repetição *while*

**while** condicao:

comando 1

comando 2

.

.

.

Podemos ler essa linha como: enquanto **condicao** for verdadeira, execute os comandos abaixo.

Exemplo:

```
val = 0
while val < 10:
    print(val)
    val = val + 1
```

# *while* vs *for*

Os dois códigos abaixo realizam a mesma tarefa

```
for val in range(0, 10):  
    print(val)
```

```
val = 0  
while val < 10:  
    print(val)  
    val = val + 1
```

# *while* vs *for*

- A estrutura ***for*** é utilizada quando sabemos o número de vezes que o código será repetido
- No caso da estrutura ***while***, o número de vezes que o código será repetido é, em geral, desconhecido. O código será executado até que alguma condição seja satisfeita.
- Cuidado! Se a condição nunca for satisfeita, temos um loop infinito e o código nunca termina de executar!
- O código abaixo executará indefinidamente

```
val = 1
while val>0:
    print(val)
    val = val + 1
```



# Estrutura *while*

- A estrutura *while* é particularmente útil para requisitarmos entradas do usuário até que uma condição de parada seja atingida.
- Há três abordagens comumente utilizadas:

# Estrutura *while*

- A estrutura `while` é particularmente útil para requisitarmos entradas do usuário até que uma condição de parada seja atingida.
- Há três abordagens comumente utilizadas:
  1. Receber um valor antes de iniciar a estrutura `while`:

```
opcao = input("Digite um valor, ou 's' para sair: ")
while opcao!="s":
    valor = int(opcao)
    print(f"Valor digitado: {valor}")
    opcao = input("Digite um valor, ou 's' para sair: ")
```

- Desvantagem: Há redundância de código ao utilizar duas vezes o `input`

# Estrutura *while*

- A estrutura `while` é particularmente útil para requisitarmos entradas do usuário até que uma condição de parada seja atingida.
- Há três abordagens comumente utilizadas:
  1. Receber um valor antes de iniciar a estrutura `while`;
  2. Usar uma variável booleana:

```
continuar = True
while continuar:
    opcao = input("Digite um valor, ou 's' para sair: ")
    if opcao=="s":
        continuar = False
    else:
        valor = int(opcao)
        print(f"Valor digitado: {valor}")
```

# Estrutura *while*

- A estrutura `while` é particularmente útil para requisitarmos entradas do usuário até que uma condição de parada seja atingida.
- Há três abordagens comumente utilizadas:
  1. Receber um valor antes de iniciar a estrutura `while`;
  2. Usar uma variável booleana;
  3. Usar um laço `while` infinito em conjunto com o comando `break`:

```
while True:
    opcao = input("Digite um valor, ou 's' para sair: ")
    if opcao=="s":
        break
    valor = int(opcao)
    print(f"Valor digitado: {valor}")
```

# Exercício 9 - Soma com while

Faça um programa que receba como entrada números do usuário até que ele digite a palavra "sair". O programa imprime na tela a soma de todos os valores digitados

REPITA ATÉ QUE O USUÁRIO DIGITE "sair":

- O usuário digita um valor

SAÍDA:

- A soma dos valores

# Solução

```
soma = 0
continuar = True
while continuar:
    opcao = input("Digite um valor, ou 's' para sair: ")
    if opcao=="s":
        continuar = False
    else:
        valor = int(opcao)
        soma += valor

print(f"Soma dos valores: {soma}")
```

# Exercício 10 - Loteria Aprimorada

Faça um programa no qual o usuário terá que adivinhar um número entre 1 e 9. No início, é mostrado o seguinte menu

Menu

-----

1. Inserir créditos (você possui R\$0.0)
2. Jogar
3. Sair

Digite a opção:

Na opção 1, o programa solicita que o usuário digite um valor para inserir créditos. Na opção 2, o programa solicita que o usuário digite valores até que ele acerte ou os créditos acabem. Cada tentativa custa R\$5,00.

Se o valor for adivinhado, o usuário ganha R\$50,00, um novo número é sorteado e o menu principal é mostrado.

# Solução

```
import random
num_oculto = random.randint(1, 9)

print("Seja bem vindo(a) ao DCBet! ")
print("Estou pensando em um valor entre 1 e 9. Tente adivinhar o valor.")
print("A cada R$5.00 inseridos você ganha uma tentativa.")
print("A prêmio é de R$50.00 se você acertar o valor.")

creditos = 0.0
encerrar = False
while encerrar!=True:

    print("\nMenu")
    print("-----")
    print(f"1. Inserir créditos (você possui R${creditos})")
    print("2. Jogar")
    print("3. Sair")

    opcao = input("Digite a opção: ")

    if opcao=="1":
        ...
    elif opcao=="2":
        ...
    elif opcao=="3":
        ...
    else:
        print("Opção desconhecida!")
```

\* O elif completo é  
mostrado no próximo slide



# Solução

```
...
if opcao=="1":
    inseridos = float(input("Digite quanto quer inserir: "))
    creditos += inseridos
    print(f"R${creditos} inseridos com sucesso!")
elif opcao=="2":
    sucesso = False
    while creditos>=5.0 and sucesso==False:
        valor = int(input("Digite o valor: "))
        creditos -= 5.0
        if valor==num_oculto:
            sucesso = True
        else:
            print(f"Não foi dessa vez! (R${creditos} restantes)\n")

    if sucesso:
        print("Parabéns! Você ganhou R$50.00")
        creditos += 50.0
        # Sorteia novamente um valor
        num_oculto = random.randint(1, 9)
    else:
        print("Seus créditos acabaram!")
elif opcao=="3":
    print("Até logo!")
    encerrar = True
else:
    print("Opção desconhecida!")
```