

Python para Análise de Dados: Estruturas Condicionais e de Repetição

André Grégio
Paulo Almeida



JUSTIÇA 4.0: INOVAÇÃO E EFETIVIDADE NA REALIZAÇÃO DA JUSTIÇA PARA TODOS
PROJETO DE EXECUÇÃO NACIONAL BRA/20/015

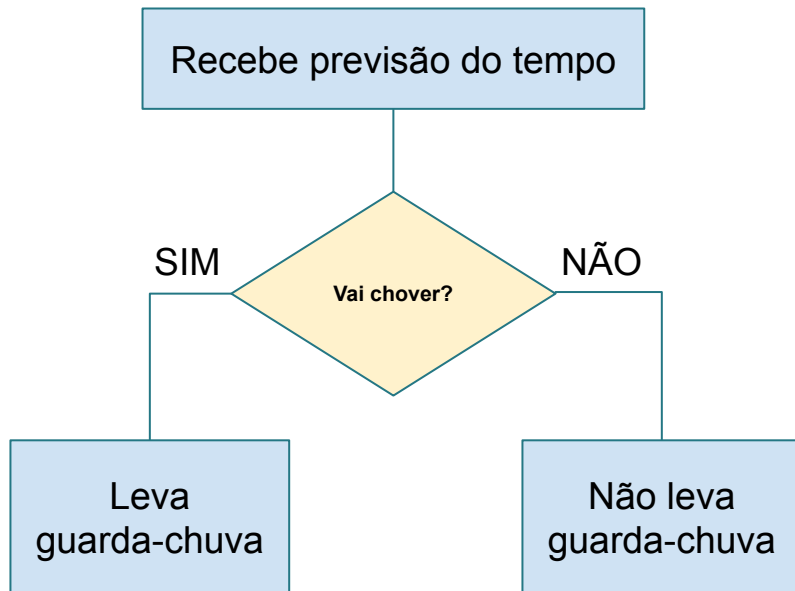
1.

Testes Condicionais



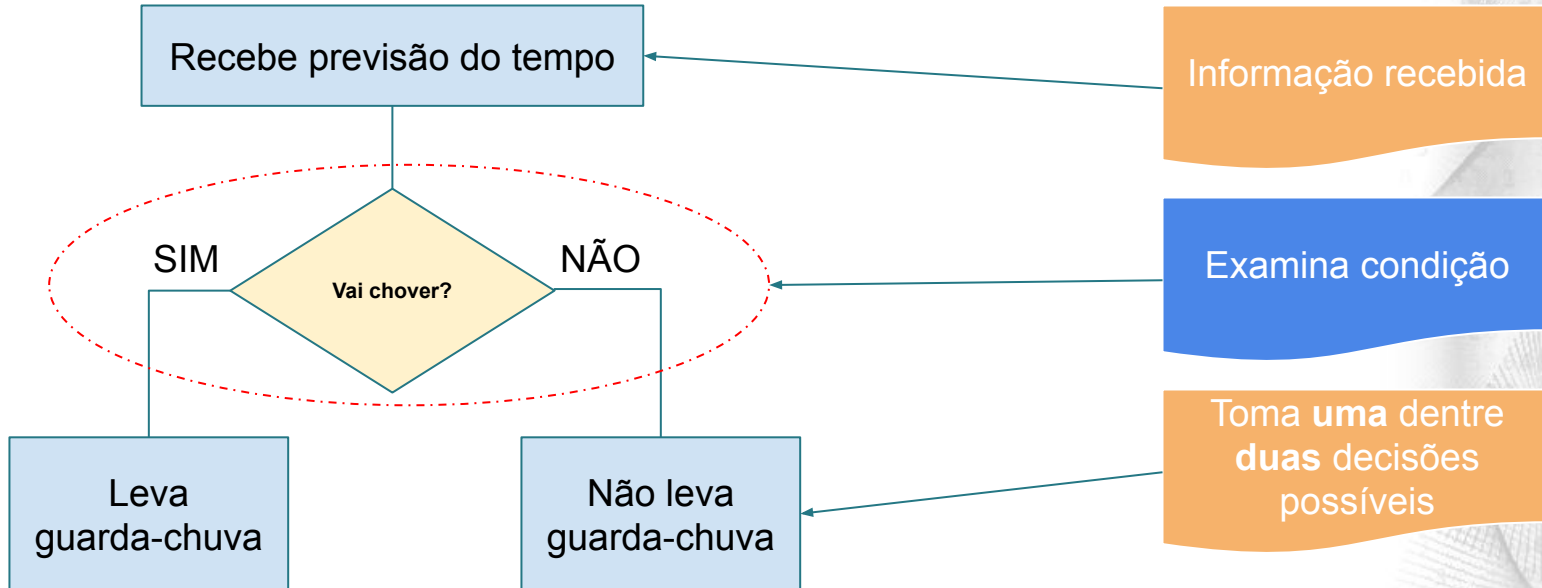
Declarações condicionais

Alguns problemas requerem examinar um conjunto de condições antes da tomada de decisão:



Declarações condicionais

Alguns problemas requerem examinar um conjunto de condições antes da tomada de decisão:



O comando “if”

O comando “if” está presente na maioria das linguagens de programação e permite:

1. Examinar o estado de um programa
2. Modificar seu fluxo de execução

É possível checar mais de uma situação a fim de se gerar declarações condicionais complexas para garantir as condições exatas requeridas!

O comando “if”

Testes condicionais (vistos anteriormente com operadores de comparação) servem para verificar se uma dada situação é *verdadeira* ou *falsa*:

```
>>> carro1 = "Honda"
>>> carro2 = "Toyota"
>>> carro1 == carro2
```

False

```
>>> carro1 = "Honda"
>>> carro2 = "honda"
>>> carro1.lower() == carro2
```

True

O comando “if”

Quando se faz uma verificação usando “if”, ocorre um teste condicional

- ▶ Python usa os valores “True” e “False” para decidir que rumo tomar:
 - ▶ Se o resultado do teste é “True”, o código seguinte ao comando “if” é executado
 - ▶ Se o resultado do teste é “False”, o código que segue o comando “if” é ignorado

O comando “if”

Quando se faz uma verificação usando “if”, ocorre um teste condicional

- ▶ Python usa os valores “True” e “False” para decidir que rumo tomar:

```
carro = "suzuki"
```

- ▶ Se o resultado do teste é “True”, o código seguinte ao comando “if” é executado

```
if (carro.title() == "Suzuki"):  
    print("Acertou!")
```

- ▶ e o resultado do teste é “False”, o código que segue o comando “if” é ignorado

O comando “if”

Quando se faz uma verificação usando “if”, ocorre um teste condicional

- ▶ Python usa os valores “True” e “False” para decidir que rumo tomar:

```
carro = "suzuki"
```

- ▶ Se o resultado do teste é “True”, o código seguinte ao comando “if” é executado

```
if (carro.title() == "Suzuki"):  
    print("Acertou!")
```



Acertou!

- ▶ e o resultado do teste é “False”, o código que segue o comando “if” é ignorado

O comando “if”

Quando se faz uma verificação usando “if”, ocorre um teste condicional

- ▶ Python usa os valores “True” e “False” para decidir que rumo tomar:

```
entrou = True
```

- ▶ Se o resultado do teste é “True”, o código seguinte ao comando “if” é executado
- ▶ Se o resultado do teste é “False”, o código que segue o comando “if” é ignorado

```
if (entrou):  
    print("Dentro do if")  
print("Fora do if")
```

O comando “if”

Um exemplo de checagem de desigualdade:

```
resposta = int(input("Qual a resposta para tudo?\nR.: "))  
  
if resposta != 42:  
    print("Errou, tente novamente!")
```

O comando “if”

Um exemplo de checagem de desigualdade:

```
resposta = 17  
  
if resposta != 42:  
    print("Errou, tente novamente!")
```

Desigualdade foi
satisfeita (17 != 42)

Entra no “if” e
imprime

Próximo tópico: checagem de múltiplas condições



1.

Testes Condicionais com Operadores Lógicos



Checagem de múltiplas condições

Operadores de comparação + operadores lógicos!

```
idade = int(input("Digite sua idade: "))
if idade >= 18:
    print("Pode dirigir no Brasil...")
if idade < 18:
    print("Não pode dirigir no Brasil!")
if idade > 15:
    print("Pode dirigir nos EUA...")
if idade >= 16 and idade < 21:
    print("Pode dirigir, mas não comprar álcool nos EUA")
```

Checagem de múltiplas condições

Operadores de comparação + operadores lógicos!

Uso de parênteses
melhora legibilidade
do código!

```
if (idade >= 16) and (idade < 21):  
    print("Pode dirigir, mas não entrar na balada nos EUA")
```

```
if ((pessoa1 >= 18) or (pessoa2 >= 18)):  
    print("Menor pode entrar no cinema acompanhado por pais ou  
tutor")
```


Checagem de múltiplas condições

Operadores de comparação + operadores lógicos!

Combinação de
operadores lógicos!

```
if ((pessoa1 >= 18) or (pessoa2 >= 18)) and ((papell == 'responsável') or  
(papel2 == 'responsável')):  
    print("Menor pode entrar no cinema acompanhado por pais ou  
tutor")
```

Checagem de múltiplas condições

Operadores de comparação + operadores lógicos!

```
if ((pessoa1 >= 18) or (pessoa2 >= 18))  
  
    and  
  
    ((papell == 'responsável') or (papel2 == 'responsável'))  
  
    and not  
  
    ((pessoa1 < 12) or (pessoa2 < 12)):  
  
    print("Ambos assistem o filme no cinema...")
```

Se copiar e colar do
jeito que está não vai
funcionar
(IDENTAÇÃO!!!)

Próximo tópico: blocos de condicionais



2.

Condicionais

Mútiplos/Aninhados (parte 1)



Condicionais em bloco

Lembre-se: quando uma condição é satisfeita, a instrução “dentro” do teste é executada (em Python, o comando ou bloco indentado após o IF)

```
idade = int(input("Digite sua idade: "))
if (idade > 16 or idade >= 70):
    print("Pode votar!")
    print("Já fez seu título de eleitor?")
if (idade >= 18 and idade < 70):
    print("Na sua idade, o voto é obrigatório.")
if (idade < 16):
    print("Você ainda não pode votar")
```

Condicionais em bloco

Lembre-se: quando uma condição é satisfeita, a instrução “dentro” do teste é executada (em Python, o comando ou bloco indentado após o IF)

```
idade = int(input("Digite sua idade: "))  
if (idade > 16 or idade >= 70):  
    print("Pode votar!")  
    print("Já fez seu título de eleitor?")  
if (idade >= 18 and idade < 70):  
    print("Na sua idade, o voto é obrigatório.")  
if (idade < 16):  
    print("Você ainda não pode votar")
```

Condicionais em bloco

Lembre-se: quando uma condição é satisfeita, a instrução “dentro” do teste é executada (em Python, o comando ou bloco indentado após o IF)

```
idade = int(input("Digite sua idade: "))
if (idade > 16 or idade >= 70):
    print("Pode votar!")
    print("Já fez seu título de eleitor?")
if (idade >= 18 and idade < 70):
    print("Na sua idade, o voto é obrigatório.")
if (idade < 16):
    print("Você ainda não pode votar")
```

17

V or F = ?

Condicionais em bloco

Lembre-se: quando uma condição é satisfeita, a instrução “dentro” do teste é executada (em Python, o comando ou bloco indentado após o IF)

```
idade = int(input("Digite sua idade: "))
if (idade > 16 or idade >= 70):
    print("Pode votar!")
    print("Já fez seu título de eleitor?")
if (idade >= 18 and idade < 70):
    print("Na sua idade, o voto é obrigatório.")
if (idade < 16):
    print("Você ainda não pode votar")
```

17

V or F = V

Condicionais em bloco

Lembre-se: quando uma condição é satisfeita, a instrução “dentro” do teste é executada (em Python, o comando ou bloco indentado após o IF)

```
idade = int(input("Digite sua idade: "))
if (idade > 16 or idade >= 70):
    print("Pode votar!")
    print("Já fez seu título de eleitor?")
if (idade >= 18 and idade < 70):
    print("Na sua idade, o voto é obrigatório.")
if (idade < 16):
    print("Você ainda não pode votar")
```

17

V

F and V = ?

Condicionais em bloco

Lembre-se: quando uma condição é satisfeita, a instrução “dentro” do teste é executada (em Python, o comando ou bloco indentado após o IF)

```
idade = int(input("Digite sua idade: "))  
if (idade > 16 or idade >= 70):  
    print("Pode votar!")  
    print("Já fez seu título de eleitor?")  
if (idade >= 18 and idade < 70):  
    print("Na sua idade, o voto é obrigatório.")  
if (idade < 16):  
    print("Você ainda não pode votar")
```

17

V

F and V = F

Condicionais em bloco

Lembre-se: quando uma condição é satisfeita, a instrução “dentro” do teste é executada (em Python, o comando ou bloco indentado após o IF)

```
idade = int(input("Digite sua idade: "))  
if (idade > 16 or idade >= 70):  
    print("Pode votar!")  
    print("Já fez seu título de eleitor?")  
if (idade >= 18 and idade < 70):  
    print("Na sua idade, o voto é obrigatório.")  
if (idade < 16):  
    print("Você ainda não pode votar")
```

17

V

F

F

Condicionais em bloco

Lembre-se: quando uma condição é satisfeita, a instrução “dentro” do teste é executada (em Python, o comando ou bloco indentado após o IF)

```
idade = int(input("Digite sua idade: "))  
if (idade > 16 or idade >= 70):  
    print("Pode votar!")  
    print("Já fez seu título de eleitor?")  
if (idade >= 18 and idade < 70):  
    print("Na sua idade, o voto é obrigatório.")  
if (idade < 16):  
    print("Você ainda não pode votar")
```

17

V

F

F

Condicionais em bloco

Lembre-se: quando uma condição é satisfeita, a instrução “dentro” do teste é executada (em Python, o comando ou bloco indentado após o IF)

```
idade = int(input("Digite sua idade: "))  
if (idade > 16 or idade >= 70):  
    print("Pode votar!")  
    print("Já fez seu título de eleitor?")  
if (idade >= 18 and idade < 70):  
    print("Na sua idade, o voto é obrigatório.")  
if (idade < 16):  
    print("Você ainda não pode votar")
```

16?

Próximo tópico: o comando else



2.

Condicionalis

Mútiplos/Aninhados (parte 2)



Adicionando instruções a ambos os fluxos

Em algumas situações, é necessário tratar ambos os resultados do teste condicional, i.e., dar uma resposta (ou executar um comando) para V ou F:

```
idade = int(input("Digite sua idade: "))
if (idade >= 16):
    print("Você pode votar!")
```


Adicionando instruções a ambos os fluxos

Em algumas situações, é necessário tratar ambos os resultados do teste condicional, i.e., dar uma resposta (ou executar um comando) para V ou F:

```
idade = int(input("Digite sua idade: "))  
if (idade >= 16):  
    print("Você pode votar!")
```

15

Adicionando instruções a ambos os fluxos

Em algumas situações, é necessário tratar ambos os resultados do teste condicional, i.e., dar uma resposta (ou executar um comando) para V ou F:

```
idade = int(input("Digite sua idade: "))  
if (idade >= 16):  
    print("Você pode votar!")
```

15 >= 16?
Falso!

Adicionando instruções a ambos os fluxos

Em algumas situações, é necessário tratar ambos os resultados do teste condicional, i.e., dar uma resposta (ou executar um comando) para V ou F:

```
idade = int(input("Digite sua idade: "))  
if (idade >= 16):  
    print("Você pode votar!")
```

15 >= 16?
Falso!

Nada acontece!

Adicionando instruções a ambos os fluxos

Em algumas situações, é necessário tratar ambos os resultados do teste condicional, i.e., dar uma resposta (ou executar um comando) para V ou F:

```
idade = int(input("Digite sua idade: "))
if (idade >= 16):
    print("Você pode votar!")
else:
    print("Você ainda não pode votar!")
```

Adicionando instruções a ambos os fluxos

Em algumas situações, é necessário tratar ambos os resultados do teste condicional, i.e., dar uma resposta (ou executar um comando) para V ou F:

```
idade = int(input("Digite sua idade: "))  
if (idade >= 16):  
    print("Você pode votar!")  
else:  
    print("Você ainda não pode votar!")
```

Se condição
satisfeita

Senão

Adicionando instruções a ambos os fluxos

Portanto, os comandos do tipo **if-else** permitem:

- ▶ A criação de dois blocos distintos de execução (um para V outro para F)
- ▶ A definição de um conjunto de ações a se executar quando o teste falha

Logo, os blocos de comando **if-else**:

- ▶ Sempre executam **uma** de **duas** ações possíveis
- ▶ Limitam-se à avaliação das duas situações dadas

Próximo tópico: o comando elif



2.

Condicionais

Mútiplos/Aninhados (parte 3)

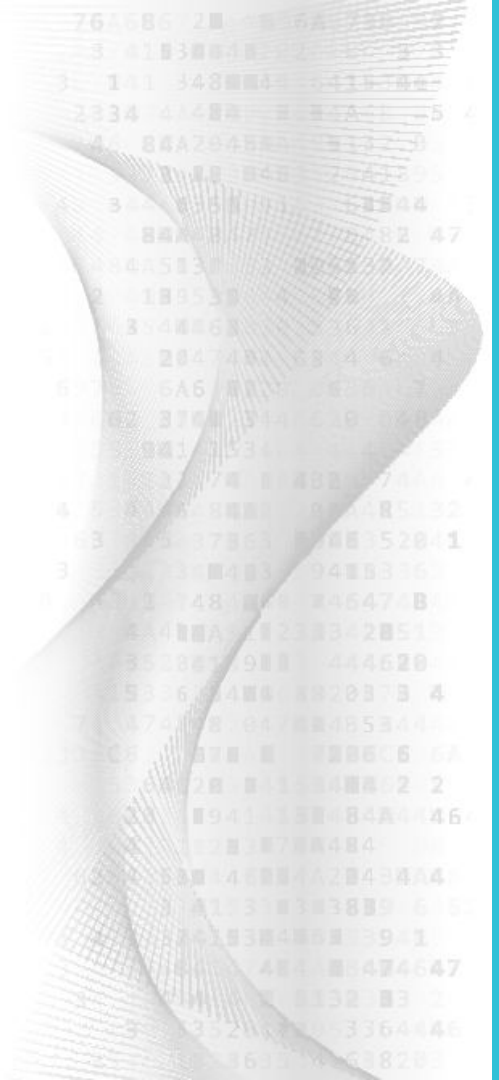


Controlando mais do que duas situações

Podemos criar novos fluxos de execução em um programa simplesmente adicionando mais condições (permite melhor controle do estado do programa).

Exemplo:

Como saber qual o maior dentre dois números?



Controlando mais do que duas situações

Exemplo: imprimir o maior dentre dois números.

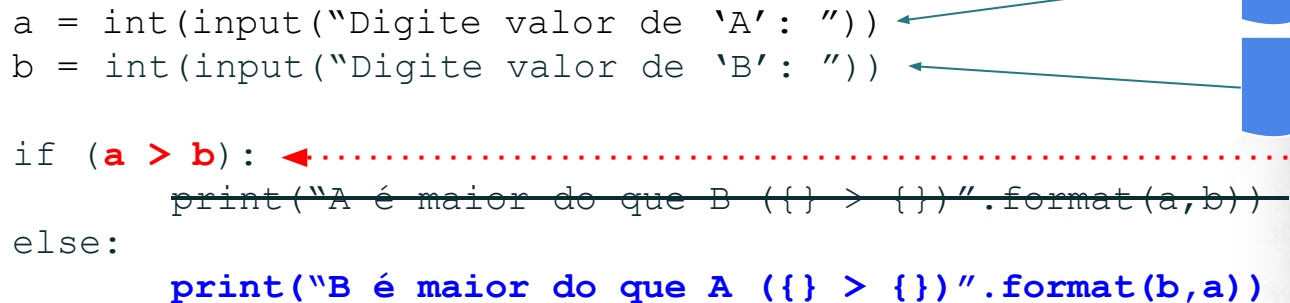
```
a = int(input("Digite valor de 'A': "))
b = int(input("Digite valor de 'B': "))

if (a > b):
    print("A é maior do que B ({} > {})".format(a,b))
else:
    print("B é maior do que A ({} > {})".format(b,a))
```

Controlando mais do que duas situações

Exemplo: imprimir o maior dentre dois números.

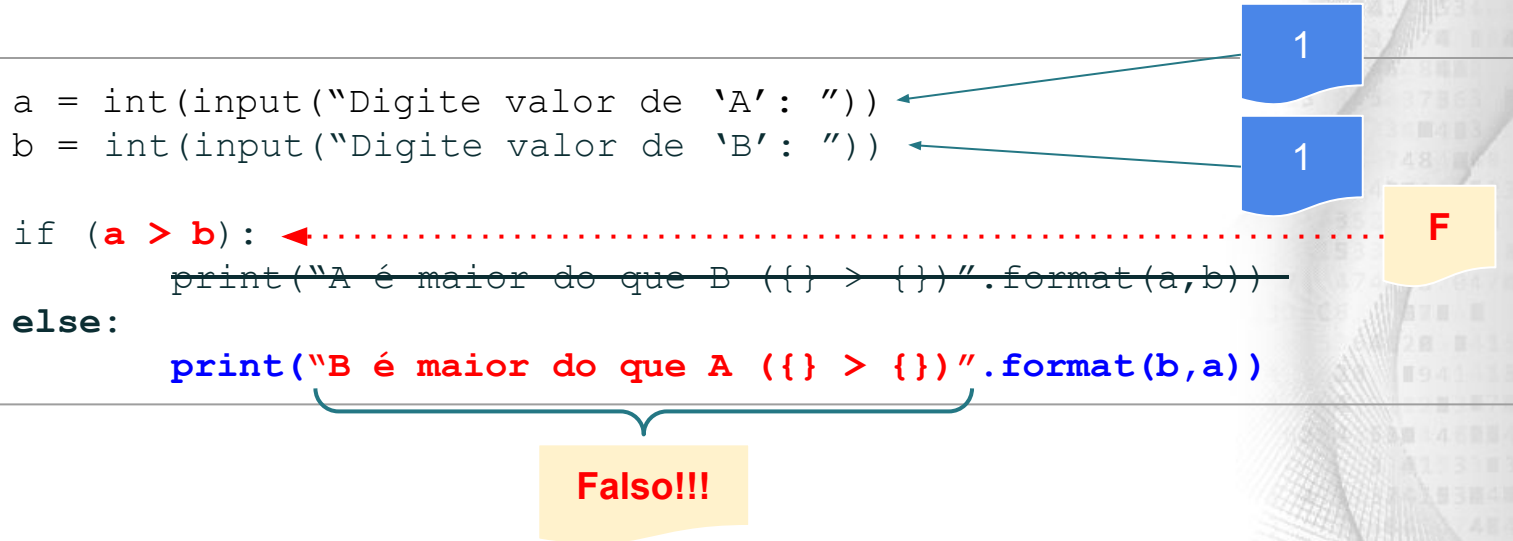
```
a = int(input("Digite valor de 'A': "))  
b = int(input("Digite valor de 'B': "))  
  
if (a > b):  
    print("A é maior do que B ({} > {})".format(a,b))  
else:  
    print("B é maior do que A ({} > {})".format(b,a))
```



Controlando mais do que duas situações

Exemplo: imprimir o maior dentre dois números.

```
a = int(input("Digite valor de 'A': "))  
b = int(input("Digite valor de 'B': "))  
  
if (a > b):  
    print("A é maior do que B ({} > {})".format(a,b))  
else:  
    print("B é maior do que A ({} > {})".format(b,a))
```



Falso!!!

Controlando mais do que duas situações

Exemplo: imprimir o maior dentre dois números.

```
a = int(input("Digite valor de 'A': "))
b = int(input("Digite valor de 'B': "))

if (a > b):
    print("A é maior do que B ({} > {})".format(a,b))
elif (a < b):
    print("B é maior do que A ({} > {})".format(b,a))
else:
    print("A e B são iguais!")
```

A cadeia if-elif-else

A sintaxe vista anteriormente, também conhecida como cadeia “if-elif-else” em Python, pode ser utilizada quando se quer **testar mais de duas condições**

- ▶ Pode-se adicionar quantos *elif* forem necessários
- ▶ Só pode haver um único *if* e um único *else*
- ▶ Em outras linguagens, a sintaxe pode mudar para *elsif* ou *else if* nos testes condicionais intermediários

A cadeia if-elif-else

Imagine uma loja que diferencia preços de acordo com a forma de pagamento:

- ▶ O preço da etiqueta é válido para pagamento no cartão de crédito
- ▶ Pagamento no cartão de débito tem 3% de desconto
- ▶ Pagamento à vista em dinheiro tem 10% de desconto
- ▶ Pagamento via boleto tem 5% de desconto

A cadeia if-elif-else

Exemplo: opções [c]rédito, [d]ébito, [b]oleto ou Dinheir[o] (0, 3, 5, 10 % de desc.)

```
opcao = input("Forma de pagamento [c|d|b|o]: ")

if (opcao == 'c'):
    print("Pagamento no crédito sem desconto.")
elif (opcao == 'd'):
    print("Pagamento no débito com 3% de desconto.")
elif (opcao == 'b'):
    print("Pagamento no boleto com 5% de desconto.")
elif (opcao == 'o'):
    print("Pagamento em dinheiro com 10% de desconto.")
else:
    print("Opção '{}' não cadastrada".format(opcao))
```


A cadeia if-elif-else

Exemplo: opções [c]rédito, [d]ébito, [b]oleto ou Dinheir[o] (0, 3, 5, 10 % de desc.)

```
opcao = input("Forma de pagamento [c|d|b|o]: ")

if (opcao == 'c'):
    print("Pagamento no crédito sem desconto.")
elif (opcao == 'd'):
    print("Pagamento no débito com 3% de desconto.")
elif (opcao == 'b'):
    print("Pagamento no boleto com 5% de desconto.")
elif (opcao == 'o'):
    print("Pagamento em dinheiro com 10% de desconto.")
else:
    print("Opção '{}' não cadastrada".format(opcao))
```

'd'

A cadeia if-elif-else

Exemplo: opções [c]rédito, [d]ébito, [b]oleto ou Dinheir[o] (0, 3, 5, 10 % de desc.)

```
opcao = input("Forma de pagamento [c|d|b|o]: ")

if (opcao == 'c'):
    print("Pagamento no crédito sem desconto.")
elif (opcao == 'd'):
    print("Pagamento no débito com 3% de desconto.")
elif (opcao == 'b'):
    print("Pagamento no boleto com 5% de desconto.")
elif (opcao == 'o'):
    print("Pagamento em dinheiro com 10% de desconto.")
else:
    print("Opção '{}' não cadastrada".format(opcao))
```

'd'

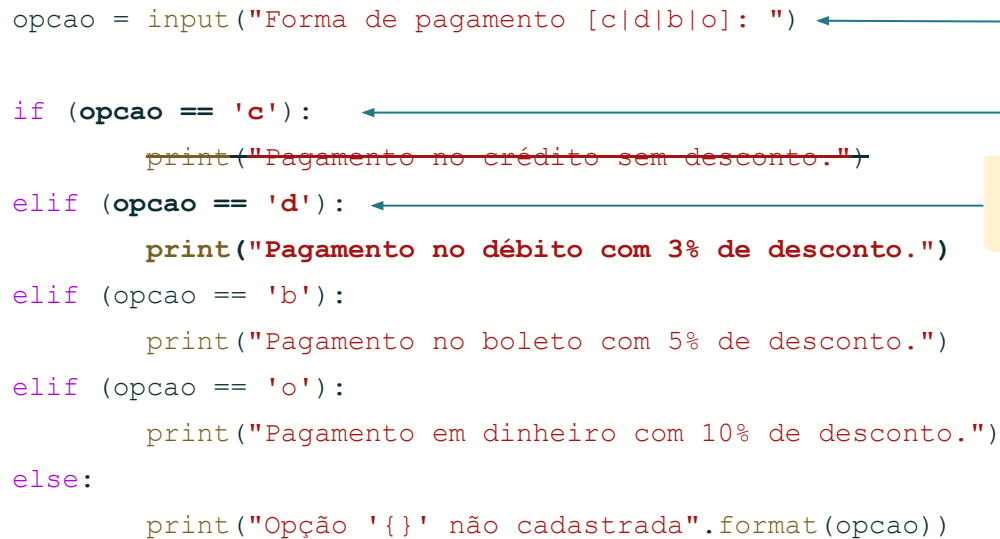
F

A cadeia if-elif-else

Exemplo: opções [c]rédito, [d]ébito, [b]oleto ou Dinheir[o] (0, 3, 5, 10 % de desc.)

```
opcao = input("Forma de pagamento [c|d|b|o]: ")

if (opcao == 'c'):
    print("Pagamento no crédito sem desconto.")
elif (opcao == 'd'):
    print("Pagamento no débito com 3% de desconto.")
elif (opcao == 'b'):
    print("Pagamento no boleto com 5% de desconto.")
elif (opcao == 'o'):
    print("Pagamento em dinheiro com 10% de desconto.")
else:
    print("Opção '{}' não cadastrada".format(opcao))
```



A cadeia if-elif-else

Exemplo: opções [c]rédito, [d]ébito, [b]oleto ou Dinheir[o] (0, 3, 5, 10 % de desc.)

```
opcao = input("Forma de pagamento [c|d|b|o]: ")

if (opcao == 'c'):
    print("Pagamento no crédito sem desconto.")
elif (opcao == 'd'):
    print("Pagamento no débito com 3% de desconto.")
elif (opcao == 'b'):
    print("Pagamento no boleto com 5% de desconto.")
elif (opcao == 'o'):
    print("Pagamento em dinheiro com 10% de desconto.")
else:
    print("Opção '{}' não cadastrada".format(opcao))
```

'd'

Após executar o bloco referente à opção "d", nenhum teste adicional é feito e o programa *pula* para a instrução logo após o ELSE (fora do bloco dele)

Sem mais instruções, o programa termina sua execução e sai...

Próximo tópico: o comando condicionais aninhados



2.

Condicionais

Mútiplos/Aninhados (parte 4)



Aninhamento de condicionais

É possível fazer testes condicionais dentro de testes preexistentes:

```
idade = int(input("Digite sua idade: "))

if (idade >= 16):
    print("Você já pode votar se tiver título de eleitor.")
    if (idade >= 18 and idade <= 70):
        print("Se você é alfabetizado, seu voto é obrigatório!")
    else:
        print("Seu voto é facultativo.")
else:
    print("Você ainda não pode votar...")
```

Aninhamento de condicionais

É possível fazer testes condicionais dentro de testes preexistentes:

```
idade = int(input("Digite sua idade: "))  
  
if (idade >= 16):  
    print("Você já pode votar se tiver título de eleitor.")  
    if (idade >= 18 and idade <= 70):  
        print("Se você é alfabetizado, seu voto é obrigatório!")  
    else:  
        print("Seu voto é facultativo.")  
else:  
    print("Você ainda não pode votar...")
```


Aninhamento de condicionais

É possível fazer testes condicionais dentro de testes preexistentes:

```
idade = int(input("Digite sua idade: "))  
  
if (idade >= 16):  
    print("Você já pode votar se tiver título de eleitor.")  
    if (idade >= 18 and idade <= 70):  
        print("Se você é alfabetizado, seu voto é obrigatório!")  
    else:  
        print("Seu voto é facultativo.")  
else:  
    print("Você ainda não pode votar...")
```

75

V

Aninhamento de condicionais

É possível fazer testes condicionais dentro de testes preexistentes:

```
idade = int(input("Digite sua idade: "))  
  
if (idade >= 16):  
    print("Você já pode votar se tiver título de eleitor.")  
    if (idade >= 18 and idade <= 70):  
        print("Se você é alfabetizado, seu voto é obrigatório!")  
    else:  
        print("Seu voto é facultativo.")  
else:  
    print("Você ainda não pode votar...")
```

75

**Este bloco
inteiro será
executado**

Aninhamento de condicionais

É possível fazer testes condicionais dentro de testes preexistentes:

```
idade = int(input("Digite sua idade: "))

if (idade >= 16):
    print("Você já pode votar se tiver título de eleitor.")
    if (idade >= 18 and idade <= 70):
        print("Se você é alfabetizado, seu voto é obrigatório!")
    else:
        print("Seu voto é facultativo.")
else:
    print("Você ainda não pode votar...")
```

75

O bloco do
else não...

Aninhamento de condicionais

É possível fazer testes condicionais dentro de testes preexistentes:

```
idade = int(input("Digite sua idade: "))
```

```
if (idade >= 16):
```

```
    print("Você já pode votar se tiver título de eleitor.")
```

```
    if (idade >= 18 and idade <= 70):
```

```
        print("Se você é alfabetizado, seu voto é obrigatório!")
```

```
    else:
```

```
        print("Seu voto é facultativo.")
```

```
else:
```

```
    print("Você ainda não pode votar...")
```

75

Imprime, depois
verifica se condição
é satisfeita

Aninhamento de condicionais

É possível fazer testes condicionais dentro de testes preexistentes:

```
idade = int(input("Digite sua idade: "))
```

75

```
if (idade >= 16):
```

```
    print("Você já pode votar se tiver título de eleitor.")
```

```
    if (idade >= 18 and idade <= 70):
```

```
        print("Se você é alfabetizado, seu voto é obrigatório!")
```

```
    else:
```

```
        print("Seu voto é facultativo.")
```

```
else:
```

```
    print("Você ainda não pode votar...")
```

75 >= 18? V
75 <= 70? F
F!

Aninhamento de condicionais

É possível fazer testes condicionais dentro de testes preexistentes:

```
idade = int(input("Digite sua idade: "))
```

75

```
if (idade >= 16):
```

```
    print("Você já pode votar se tiver título de eleitor.")
```

```
    if (idade >= 18 and idade <= 70):
```

```
        print("Se você é alfabetizado, seu voto é obrigatório!")
```

```
    else:
```

```
        print("Seu voto é facultativo.")
```

```
else:
```

```
    print("Você ainda não pode votar...")
```

75 >= 18? V
75 <= 70? F
F!

Aninhamento de condicionais

É possível fazer testes condicionais dentro de testes preexistentes:

```
idade = int(input("Digite sua idade: "))
```

```
if (idade >= 16):
```

```
    print("Você já pode votar se tiver título de eleitor.")
```

```
    if (idade >= 18 and idade <= 70):
```

```
        print("Se você é alfabetizado, seu voto é obrigatório!")
```

```
    else:
```

```
        print("Seu voto é facultativo.")
```

```
else:
```

```
    print("Você ainda não pode votar...")
```

75

Entra no bloco deste
else, imprime e termina
a execução

Próximo tópico: o comando for



3.

Estruturas de Repetição



Estruturas de repetição

Em alguns casos, pode-se precisar executar um bloco de comandos repetidamente: **ITERAÇÃO!**

Existem repetições **simples**, que são simplesmente fazer algo N vezes

Outras repetições são dependentes de certas **condições** serem satisfeitas para continuar a iterar...

Estruturas de repetição - for

Repetição simples:

- O comando “for” permite que se repita algo de acordo com um **intervalo**

```
for i in range(n):  
    print(i)
```

Estruturas de repetição - for

Repetição simples:

- O comando “for” permite que se repita algo de acordo com um **intervalo**

```
for i in range(n):  
    print(i)
```

```
0  
1  
...  
n-2  
n-1
```

Estruturas de repetição - for

Repetição simples:

- O comando “for” permite que se repita algo de acordo com um **intervalo**

```
for i in range(6):  
    print(i)
```

Para “i” de 0 até n-1, faça...

Estruturas de repetição - for

Repetição simples:

- O comando “for” permite que se repita algo de acordo com um **intervalo**

```
for i in range(6):  
    print(i)
```

Intervalo ABERTO!
[0, 6)

Estruturas de repetição - for

Repetição simples:

- O comando “for” permite que se repita algo de acordo com um **intervalo**

```
for i in range(6):  
    print(i)
```

Comando feito em cada
iteração

Estruturas de repetição - for

Repetição simples:

- O comando “for” permite que se repita algo de acordo com um **intervalo**

```
for i in range(6):  
    print(i)
```

Para “i” no intervalo [0, n-1)

Imprima o valor de “i”

0
1
2
3
4
5

Saída!

Estruturas de repetição - for

- O comando “for” permite repetir de acordo com um tamanho:

```
palavra = "Python"  
n = len(palavra)  
for i in range(n):  
    print(i)
```

0
1
2
3
4
5

Estruturas de repetição - for

- O comando “for” permite repetir de acordo com um tamanho:

```
palavra = "Python"  
for i in range(len(palavra)):  
    print(palavra)
```

Python

Python

Python

Python

Python

Python

Estruturas de repetição - for

Imagine um algoritmo que permita “riscar N cm” e depois “girar X graus à direita”. Para desenhar um quadrado de lado 1 cm, faríamos o seguinte programa:

```
for i in range(4):  
    risca(1)  
    gira(90)
```

Estruturas de repetição - for

O comando “for” também é chamado de **laço (ou loop)**:

- ▶ o fluxo de execução faz cada instrução dentro do bloco do “for”
- ▶ depois, volta-se ao início do laço para re-executar as instruções do bloco
- ▶ não há limite no número de instruções dentro do bloco do laço

Lembrar dos “:” para delimitar o comando
e de

identar para demarcar o bloco!

Estruturas de repetição - for

Pode-se utilizar o comando “for” para *iterar strings*:

```
palavra = "Python"
```

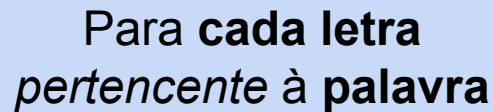
```
for letra in palavra:
```

```
    print(letra)
```

Estruturas de repetição - for

Pode-se utilizar o comando “for” para *iterar strings*:

```
palavra = "Python"  
  
for letra in palavra:  
    print(letra)
```



**Para cada letra
pertencente à palavra**

Estruturas de repetição - for

Pode-se utilizar o comando “for” para *iterar strings*:

```
palavra = "Python"  
  
for letra in palavra:  
    print(letra)
```

P
y
t
h
o
n

Estruturas de repetição

Com o comando “for”, pode-se realizar tarefas repetitivas de forma automatizada, tirando vantagem dos computadores (eles não costumam ficar cansados ou entediados, nem errar esse tipo de tarefa).

Outro comando para realizar **iterações** é o **while**...

Próximo tópico: o comando while



Estruturas de repetição

Com o comando “for”, pode-se realizar tarefas repetitivas de forma automatizada, tirando vantagem dos computadores (eles não costumam ficar cansados ou entediados, nem errar esse tipo de tarefa).

Outro comando para realizar **iterações** é o **while**:

- ▶ Estruturação do comando e suas instruções são “claras”
- ▶ Permite determinar se uma condição é *Verdadeira* ou *Falsa*
 - ▶ Se V, continua a iterar o laço até o final e volta à instrução 1 do bloco
 - ▶ Se F, sai do laço (quebra o fluxo de execução) e executa a próxima instrução após o bloco

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))
while (n > 0):
    print(n)
    n -= 1 #equivalente a n = n-1
print("Tempo esgotado!")
```

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))  
while (n > 0):  
    print(n)  
    n -= 1  
print("Tempo esgotado!")
```

3

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))  
while (n > 0):  
    print(n)  
    n -= 1  
print("Tempo esgotado!")
```

3 > 0?

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))  
while (n > 0):  
    print(n)  
    n -= 1  
print("Tempo esgotado!")
```

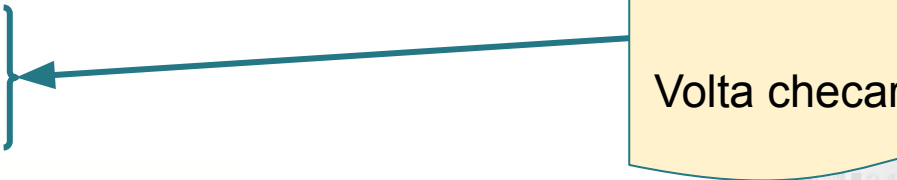
3 > 0? V

Enquanto a condição
for satisfeita, executa
o bloco...

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))  
while (n > 0):  
    print(n)  
    n -= 1  
print("Tempo esgotado!")
```



Imprime 3
 $n = 3 - 1$

Volta checar condição

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))  
while (n > 0):  
    print(n)  
    n -= 1  
print("Tempo esgotado!")
```

2 > 0? V

Enquanto a condição
for satisfeita, executa
o bloco...

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))  
while (n > 0):  
    print(n)  
    n -= 1  
print("Tempo esgotado!")
```

Imprime 2
 $n = 2 - 1$

Volta checar condição

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))  
while (n > 0):  
    print(n)  
    n -= 1  
print("Tempo esgotado!")
```

1 > 0? V

Enquanto a condição
for satisfeita, executa
o bloco...

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))  
while (n > 0):  
    print(n)  
    n -= 1  
print("Tempo esgotado!")
```

Imprime 1
 $n = 1 - 1$

Volta checar condição

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))  
while (n > 0):  
    print(n)  
    n = 1  
print("Tempo esgotado!")
```

0 > 0? F

Quando a condição não é satisfeita, pula o bloco...

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))
while (n > 0):
    print(n)
    n -= 1
print("Tempo esgotado!")
```

Imprime a próxima
instrução fora do
bloco do *while*.

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))  
while (n > 0):  
    print(n)  
    n -= 1  
print("Tempo esgotado!")
```

Sempre lembrar de
modificar a **variável
de controle!**

Estruturas de repetição - while

Exemplo de uso do comando “while” (contagem regressiva):

```
n = int(input("Digite um número inteiro: "))  
while (n > 0):  
    print(n)  
    n = n - 1  
print("Tempo esgotado!")
```

-1

Programa nunca sai
do laço para n positivo
(*loop infinito*)

Forçando a saída de um laço

```
while(True):  
    letra = input("Digite uma letra diferente  
de 'q': ")  
    if (letra == 'q'):  
        break  
print("Você digitou a letra 'q'!!!")
```

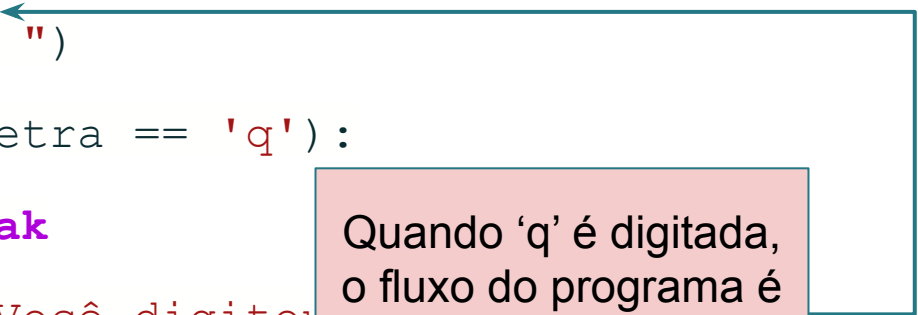

Forçando a saída de um laço

```
while(True):  
    letra = input("Digite uma letra diferente  
de 'q': ")  
    if (letra == 'q'):  
        break  
print("Você digitou")
```

Programa nunca sai
do laço até que
alguém digite 'q'
(*loop infinito*)

Forçando a saída de um laço

```
while(True):  
    letra = input("Digite uma letra diferente  
de 'q': ")  
    if (letra == 'q'):  
        break  
print("Você digitou")
```



Quando 'q' é digitada,
o fluxo do programa é
interrompido!

Forçando a saída de um laço

```
while(True):  
    letra = input("Digite uma letra diferente  
de 'q': ")  
  
    if (letra == 'q'):  
        break  
    print("Você digitou")
```

Para de iterar e vai
para a próxima
instrução...

Forçando a saída de um laço

Outro modo de sair de um laço é criar um modo de trocar a condição para Falso...

```
cond = True

while(cond):

    opcao = input("Digite 'sair' para terminar o laço")

    if opcao == 'sair':

        cond = False

    else:

        print("Ainda no laço...")
```

Teste seus conhecimentos



Exercício

Escreva um programa que imprima o preço do ingresso para um museu, dadas as condições abaixo:

- ▶ O valor-base é R\$ 10,00
- ▶ Menores de 5 anos não pagam
- ▶ Crianças entre 5 e 12 anos pagam meia-entrada
- ▶ Adolescentes entre 13 e 17 anos têm 20% de desconto
- ▶ Adultos entre 18 e 70 anos pagam o valor-base
- ▶ Maiores de 70 anos não pagam

O programa deve executar e, ao mostrar o preço, retornar ao início e perguntar se o usuário quer saber o preço de um ingresso ou sair . Para sair, o usuário deve digitar “q”.

Requisitos:

1. **O valor-base deve ser armazenado em uma variável**
2. **A idade deve ser digitada pelo usuário**
3. **Seu programa deve calcular o preço do ingresso dentro de cada condição satisfeita e mostrá-lo ao usuário**



Obrigado!

Bons Estudos!!!