

Curso Preparatório

Semana 3:

Estruturas condicionais

Admissão 2025





Estruturas de controle

- Em Python, assim como na maioria das linguagens de programação, o programa deve ser capaz de tomar decisões com base em valores e resultados gerados durante sua execução, ou seja, deve ser capaz de decidir se determinada instrução deve ou não ser executada de acordo com uma condição.
- Para atender a esse tipo de situação, podemos utilizar instruções especiais denominadas estruturas condicionais.
- As estruturas de controle servem para decidir quais blocos de código serão executados.



Estruturas de controle

No nosso dia a dia, já usamos algumas estruturas de forma involuntária, por exemplo:

Você vai sair de casa, então pensa:

Se estiver nublado:

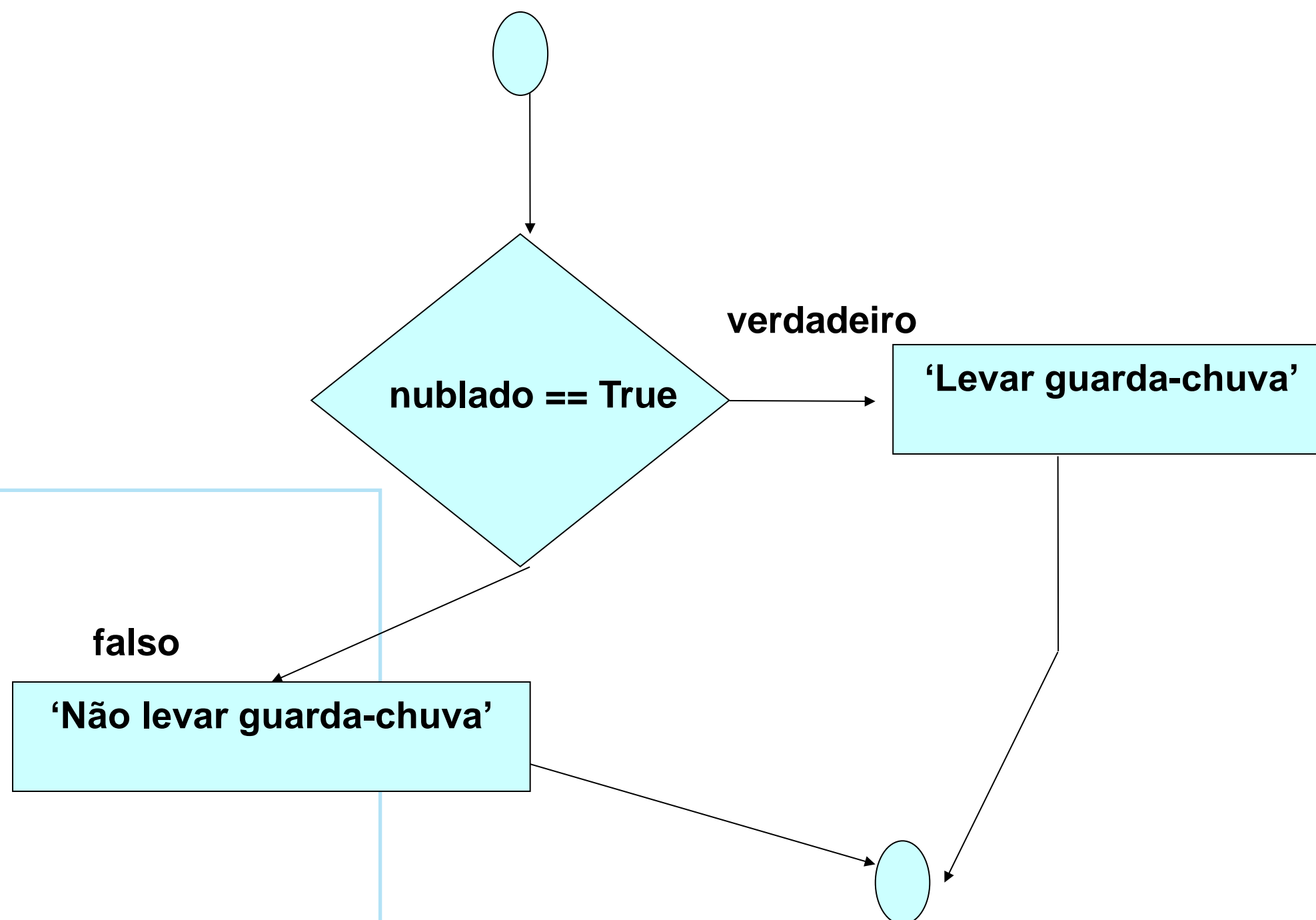
Levar guarda-chuva

Senão:

‘Não levar guarda-chuva’

Traduzindo para o computador teremos.....

if nublado==True:
 'Levar guarda-chuva'
else:
 'Não levar guarda-chuva'



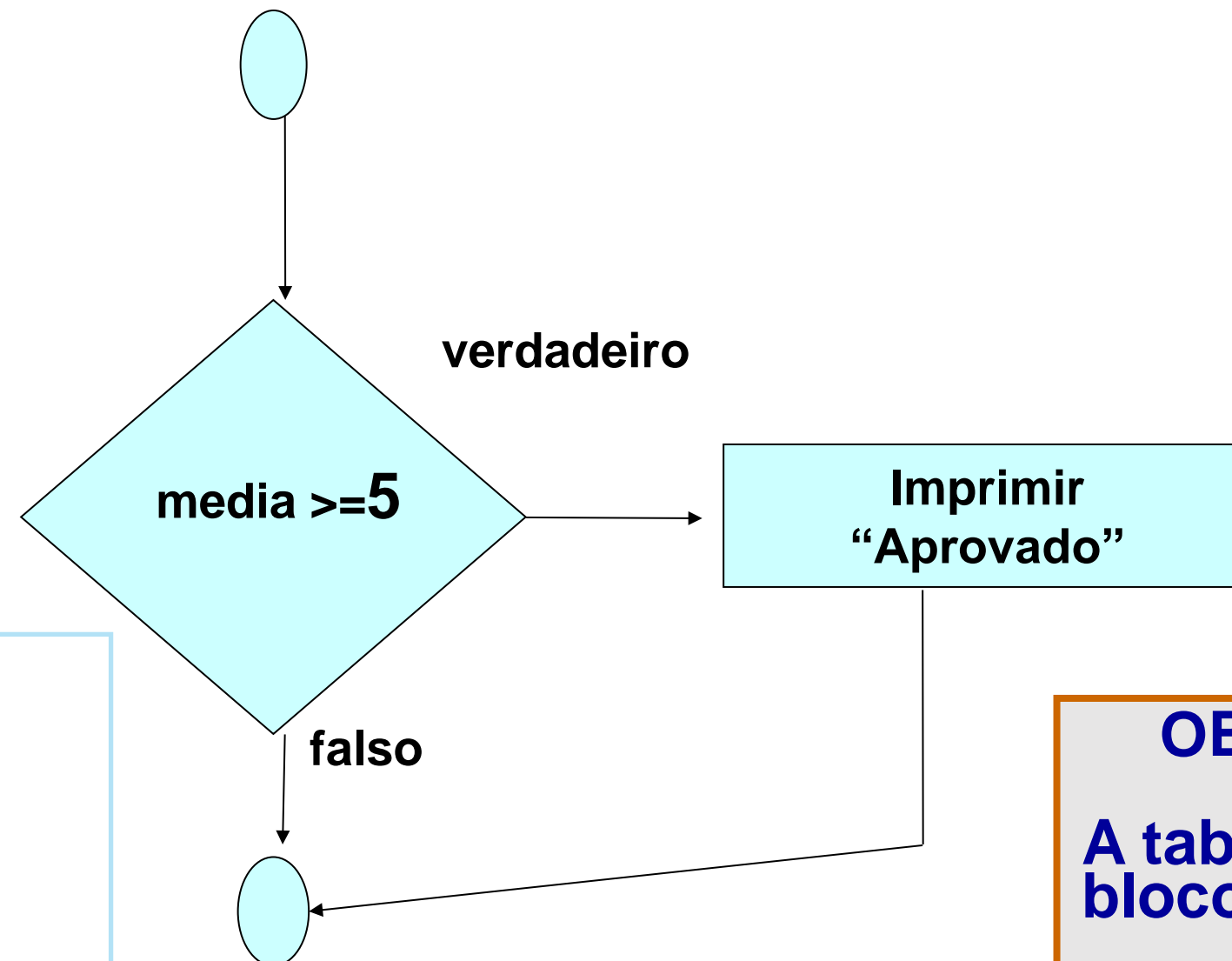
Vamos entender como devemos utilizar as estruturas de controle condicional em Python.

Primeiro, existem:

- Estruturas de controle simples
 - IF
- Estruturas composta
 - IF – ELSE
- Estruturas encadeadas
 - IF – ELSE – IF (ELIF)
- Estruturas aninhadas
 - IF
 - IF – ELSE
 - ELSE
 - IF – ELSE



Condicionais simples (SE) - IF



Python

```
if (<condição>) :  
    <instruções>
```

OBS: Além do `:` Python precisa da **indentação**
A tabulação a direita, assim que ela reconhece os blocos de códigos de uma estrutura de controle



Condicionais Simples

Python

```
if( media >= 5.0 ) :  
print( "Aprovado" )
```

JavaScript

```
if ( media >= 5.0 ) {  
    console.log( "O Aluno está " );  
    console.log( "Aprovado" );  
} // fim do bloco if
```



Condicional Simples

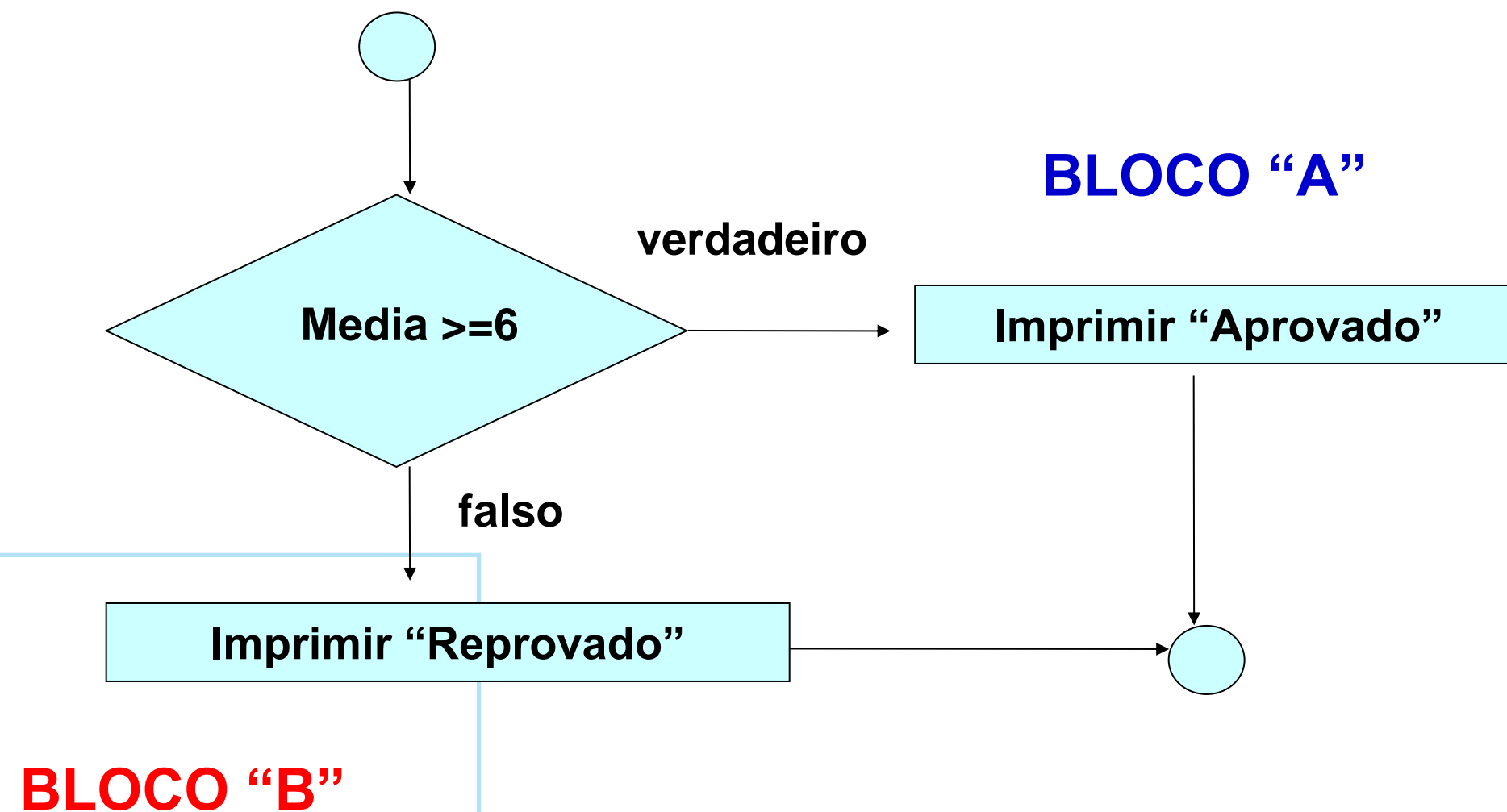
- Vamos verificar se a pessoa digitou um número positivo.

```
1  numero = int(input("Digite um numero: "))
2  if numero > 0:
3      print("Ótimo, você escolheu um numero positivo")
```

- No código acima, se o número for menor do que 0 (zero), não vai aparecer nada.
- Mas, na maioria das vezes, ao invés de fazer apenas **uma** verificação (um teste lógico), com a condicional simples (IF), é recomendado ter outra condição sendo avaliada (testada), pelo menos sendo o inverso a condicional do IF.

Condicional Composta

Condicional composta (SE-SENAO) – IF-ELSE



SINTAXE JavaScript

```
if ( media >= 5) {  
    console.log("Aprovado");  
} else {  
    console.log("Reprovado");  
}
```

Python

```
if (<condição>):  
    ____<Bloco A>  
else:  
    ____<Bloco B>
```




Estruturas de controle

- Comparando textos em Python
- Em **Python** pode usar o `==` mesmo.

if (nome `==` "Marcelo"):

- Procure usar a manipulação de strings para melhorar as condições: **`lower()`, `upper()`, `find()`, `strip()`, `split()`, `count()`, `string[]`, etc...**



Estruturas de controle

- Vamos criar um programa em Python para que o usuário digite um nome e um número qualquer.
 - Vamos comparar se esse nome foi igual a “Salomão” ou “salomão”, use a manipulação de strings para isso.
 - Vamos verificar se a pessoa digitou um número maior, menor, ou se ela digitou o número 5.

```
1 nome = input("Digite um nome?")
2 numero = int(input("Digite um numero: "))
3
4 if (nome.strip().lower() == "salomão" or nome.strip().lower() == "salomao"):
5     print("Seja bem vindo! Você digitou o nome Salomão")
6 else:
7     print("POXA! Você não digitou o nome Salomão")
8
9 if (numero > 5):
10     print("Você digitou um numero maior que 5")
11 elif (numero < 5):
12     print("Você digitou um numero menor do que 5")
13 else:
14     print("Você digitou o numero 5")
```



Estruturas de controle

Crie um programa para calcular a média aritmética de um aluno a partir de 3 provas e informe se esse aluno está aprovado ou reprovado: A média para aprovação é 6.

```
1  nota1 = float(input("Primeira nota: "))
2  nota2 = float(input("Segunda nota: "))
3  nota3 = float(input("Terceira nota: "))
4  media = (nota1+nota2+nota3)/3
5  print("Média = {:.2f}".format(media))
6  if (media >= 6):
7      print("Aprovado")
8  else:
9      print("Reprovado")
```

Em algumas disciplinas, não basta que a média das provas de um aluno seja maior ou igual a 6.0. O aluno também deve tirar uma nota mínima 3.0 na primeira atividade, como você alteraria no código acima para essa regra funcionar?



Estruturas de controle

Crie um programa para calcular a média aritmética de um aluno a partir de 3 provas e informe se esse aluno está aprovado ou reprovado: A média para aprovação é 6.

```
1  nota1 = float(input("Primeira nota: "))
2  nota2 = float(input("Segunda nota: "))
3  nota3 = float(input("Terceira nota: "))
4  media = (nota1+nota2+nota3)/3
5  print("Média = {:.2f}".format(media))
6  if (media >= 6 and nota1 > 5):
7      print("Aprovado")
8  else:
9      print("Reprovado")
```

Em algumas disciplinas, não basta que a média das provas de um aluno seja maior ou igual a 6.0. O aluno também deve tirar uma nota mínima 5.0 na primeira atividade, como você alteraria no anterior para essa regra funcionar?



Condicional Encadeada

- Suponha que, em um algoritmo, a variável X possa receber apenas quatro valores, V1, V2, V3 ou V4, e que para cada valor exista um comando a ser executado.

```
se (X=V1) {  
    C1;  
}  
se (X=V2) {  
    C2;  
}  
se (X=V3) {  
    C3;  
}  
se (X=V4) {  
    C4;  
}
```

X=V1	X=V2	X=V3	X=V4	Ação
V	F	F	F	C1
F	V	F	F	C2
F	F	V	F	C3
F	F	F	V	C4

Somente um, e apenas um comando pode ser executado.
Trata-se de uma situação excludente.

Se X é igual a V3, não é igual a V1 nem a V2 nem a V4



Condicional Encadeada

```
se (X=V1) {  
    C1;  
    senão se (X=V2) {  
        C2;  
        senão se (X=V3) {  
            C3;  
            senão se (X=V4) {  
                C4;  
            }  
        }  
    }  
}
```

X=V1	X=V2	X=V3	X=V4	Ação
V	-	-	-	C1
F	V	-	-	C2
F	F	V	-	C3
F	F	F	V	C4



Condiciona Encadeada

Condiciona encadeada (IF-ELSE IF) – IF-ELIF

JavaScript

```
if (X==V1) {  
    C1;  
}else if (X==V2) {  
    C2;  
}else if (X==V3) {  
    C3;  
}else if (X==V4) {  
    C4;  
}
```

Python

```
if (X==V1) :  
    C1  
elif (X==V2) :  
    C2  
elif (X==V3) :  
    C3  
elif (X==V4) :  
    C4
```



Vamos ver na prática?

- Para o algoritmo da média aritmética: Conceito

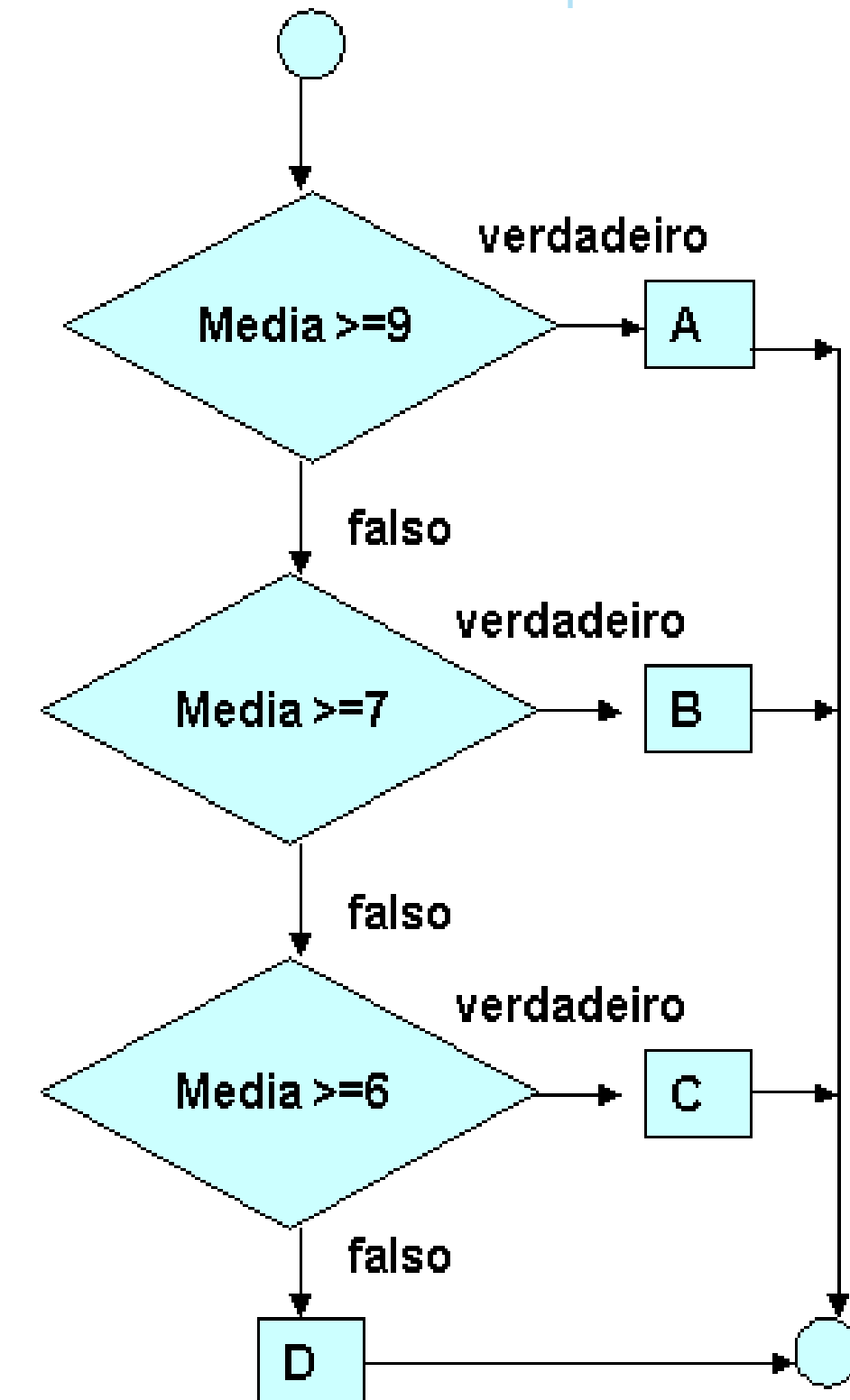
A → média ≥ 9.0 ;

B → média entre 7.0 e 8.99;

C → média entre 6.00 e 6.99

D → média abaixo de 5.99.

```
if (media >= 9):  
    print("Conceito A")  
elif (media >= 7):  
    print("Conceito B")  
elif (media >= 5):  
    print("Conceito C")  
else:  
    print("Conceito D")
```





Operadores

- Sabendo a sintaxe das estruturas condicionais em Python, como você criaria uma linha válida em Python para representar a seguinte frase na forma de teste condicional:
- **“Um aluno só será aprovado se ele tiver participado de, no mínimo 50% aulas, tiver feito pelo menos 2 atividades e tiver tirado na média das 3 notas da disciplina pelo menos 5.0.”**

```
if (faltas < (aulas/2)) and (atividades >= 2) and (media >=5):  
    print("Aprovado")
```

Sabendo que de 10 aulas, o aluno Marcelo faltou 4. Ele fez 2 das 3 atividades e ficou com as notas de 8 e 7 nas atividades que entregou. Substituindo na condição acima, Marcelo foi APROVADO ou REPROVADO?:

```
if (4 < (10/2)) and (2 >= 2) and ((8+7)/3 >=5):  
    (4<5) and (2 >=2) and (5.0>=5)  
    V and V and V → V que vai aparecer: Aprovado
```



Condicional Aninhada

Condicional aninhada (IF -> IF - ELSE IF) – IF-> IF-ELIF

- Uma condicional também pode ser aninhada dentro de outra. Poderíamos ter escrito o exemplo na seção anterior desta forma:

```
if condição1:           # Primeiro IF
    [codigo]

    if condição2:       # Segundo IF
        [codigo]
    else:                # ELSE é do segundo IF
        [codigo]

else:                    # ELSE do primeiro IF
    if condição3:
        [codigo]
    else:                # ELSE do terceiro IF
        [codigo]
```



Condicional Encadeada

- Crie um programa que o usuário vai digitar 3 números inteiros quaisquer e o programa deve dizer qual o **maior** e o **menor** dos 3 números:

Condicional E

- Crie um programa que o usuário vai deve dizer qual o **maior** e o **menor** dos

```
1  A = int(input())
2  B = int(input())
3  C = int(input())
4  if (A > B and A > C):
5      maior = A
6      if (B > C):
7          menor = C
8      else:
9          menor = B
10 elif (B > A and B > C):
11     maior = B
12     if (A > C):
13         menor = C
14     else:
15         menor = A
16 else:
17     maior = C
18     if (B > A):
19         menor = A
20     else:
21         menor = B
22
23 print("Maior =",maior)
24 print("Menor =",menor)
```

quer e o programa





Estruturas Condicionais

- Em Python você pode usar as duas sintaxes abaixo para comparar (relacionar) mais de um resultado, ou intervalo em uma variável:

- Se tiver uma variável chamada **media**, se:

Média maior ou igual 7: "APROVADO" \leftarrow `media >= 7`

Média entre 5 e 6.9: "RECUPERAÇÃO" \leftarrow `5 <= media < 7`

Média abaixo de 5: "REPROVADO" \leftarrow `media < 5`

Perceba que aqui não precisa do lógico 'E' (**and**). A variável fica no meio.

Recomendo que continue usando a **opção 1**, já que as outras linguagem usam assim.

Opção 1

```
if (media >= 7) :  
    print("Aprovado")  
elif (media >= 5 and media < 7) :  
    print("Recuperação")  
elif (media < 5) :  
    print("Reprovado")
```

Opção 2

```
if (media >= 7) :  
    print("Aprovado")  
elif ( 5 <= media < 7) :  
    print("Recuperação")  
elif (media < 5) :  
    print("Reprovado")
```



- | Valor do IMC | Classificação |
|--------------------|---------------------|
| $IMC < 19$ | Abaixo do Peso |
| $19 \leq IMC < 25$ | Peso Normal |
| $25 \leq IMC < 30$ | Sobrepeso |
| $30 \leq IMC < 40$ | Obesidade do tipo I |
| $IMC \geq 40$ | Obesidade Mórbida |

22



Exercício

- Crie um programa que receba o peso e a altura de uma pessoa, a partir dessas informações calcule o IMC dessa pessoa e a partir da tabela abaixo, o programa deve imprimir uma das opções:

```
1  peso = float(input("Qual seu peso (em KG): "))
2  altura = float(input("Qual sua altura (em metros): "))
3  imc = peso / (altura**2)
4  print("+-*15)
5  print("Seu IMC é {:.1f}".format(imc))
6  print("+-*15)
7  if (imc < 19):
8      print("Você está ABAIXO DO PESO")
9  elif (19 <= imc < 25):
10     print("Você está com PESO NORMAL")
11 elif (imc >= 25 and imc < 30):
12     print("Você está com SOBREPESO")
13 elif (imc >= 30 and imc < 40):
14     print("Você está com OBESIDADE DO TIPO 1")
15 else:
16     print("Você está com OBESIDADE MÓRBIDA")
```



SWITCH-CASE

- Não existe em **PYTHON**

Pseudocódigo

```
escolha <variável>  
  caso Valor1:  
    <instrução1>;  
  caso Valor2:  
    <instrução2>;  
    .  
    :  
fimescolha
```

JavaScript

```
switch <variável> {  
  case valor1:  
    <instruções>;  
    break;  
  case valor2:  
    <instruções>;  
    break;  
  .  
  default:  
    .  
}
```

Tem que ter o **break;** ao final de cada **case**:

Caso nenhuma condição seja verdadeira (**default:**)

Operador **in**

- O operador **in** verifica se o operando a sua esquerda, está contido em uma lista à sua direita, da mesma forma que o operador **not in** que verifica o contrário.
- Estes, são 2 operadores nativo para verificar se um determinado objeto está contido numa lista. A palavra **in**, do Inglês, significa, "contido em". Essa é uma maneira para simplificar a verificação se o elemento X está contido na lista Y.
- **EX: verificar se um item está na lista:**

```
1 minhas_sobremesas = ["bolo de chocolate", "sorvete", "pudim", "paçoca", "brigadeiro"]
2 x = input("Digite sua sobremesa favorita: ")
3 if x in minhas_sobremesas:
4     print("Essa também é uma das minhas sobremesas favoritas!")
5 else:
6     print("Hmm, eu não havia pensado nessa!!")
```

Operador **in**

- Considere uma progressão aritmética (P.A) de 10 termos cujo primeiro termo é ***a1*** e a razão é ***r***. Faça um programa que recebe um número ***n*** e verifique se ele é um termo da P.A.
- **Para saber o n^o termo de uma P.A:** $a_n = a_1 + (n - 1) * r$
- Lembrando que você pode usar o **list(range())** para gerar as P.As.
- **Sintaxe: list(range(INICIO, FIM-1, PASSO DE ADAPTAÇÃO))**

```
1 a1 = int(input("Primeiro termo da P.A: "))
2 r = int(input("Razão: "))
3 n = int(input("Qual numero quer verificar se está na P.A? "))
4 decimo = a1 + (10-1) * r
5 pa = list(range(a1, decimo, r))
6 if n in pa:
7     print(f"{n} está na P.A",pa)
8 else:
9     print(f"{n} não está na P.A",pa)
```

CONDICIONAIS

REVISANDO:

Seleção simples:

```
if (condição) :  
    C1
```

Seleção encadeada:

```
if (condição1) :  
    C1  
elif (condicao2) :  
    C2  
else:  
    C3
```

Seleção composta:

```
if (condição) :  
    C1  
else:  
    C2
```

Seleção aninhada:

```
if (condição1) :  
    C1  
    if (condicao1.1) :  
        C1.1  
    else:  
        C1.2  
elif (condicao2) :  
    C2  
    if (condicao2.1) :  
        C2.1  
    else:  
        C1.2
```





Condicionais - Prática:

- Vamos criar uma calculadora, onde:
- **Entrada:** O programa deve receber um valor inteiro qualquer, seguido de um operador, que pode ser:
 - "+" para soma;
 - "-" para subtração;
 - "*" para multiplicação;
 - "/" para divisão inteira;
 - "%" para resto de divisão;
 - "^" para exponenciação.
- Depois de escolher a operação o usuário deve escolher outro número inteiro qualquer.
- **Saída:** Na resposta deverá ser mostrado a escolha dos números e da operação escolhidos na entrada como equação, além do resultado da operação, ou a mensagem "erro" se não for possível realizar a conta (divisões por zero).
- EX: $7 \% 3 = 1$
 $5 * 2 = 10$



Condicionais - Prática:

- Vamos criar uma calculadora, onde:

```
1 num1 = int(input("Digite um numero: "))
2 print("Símbolo | Operação")
3 print("1: '+' | Somar")
4 print("2: '-' | Subtrair")
5 print("3: '*' | Multiplicar")
6 print("4: '/' | Dividir")
7 print("5: '%' | Resto da divisão")
8 print("6: '//' | Divisão inteira")
9 op = input("Qual operação? ")
10 num2 = int(input("Digite outro numero: "))
11 msg = f"{num1} {op} {num2} ="
```

```
12 if (op == "+"):
13     print(msg,num1+num2)
14 elif (op == "-"):
15     print(msg,num1-num2)
16 elif (op == "*"):
17     print(msg,num1*num2)
18 elif (op == "/" and num2 != 0):
19     print(msg,num1/num2)
20 elif (op == "%"):
21     print(msg,num1%num2)
22 elif (op == "//" and num2 != 0):
23     print(msg,num1//num2)
24 else:
25     print("Não pode dividir por 0")
```



Exercícios para praticar, mas não precisa entregar!

1. Suponha que queremos descobrir se três números (A, B e C) podem ser usados como lados de um triângulo. Obviamente os três valores têm que ser positivos e não nulos mas, além disso, em um triângulo vale a propriedade de que cada um dos seus lados é menor do que a soma dos outros dois lados.

$$A < (B+C) \text{ e } B < (A+C) \text{ e } C < (B+A)$$

2. Crie um programa que faça com que o usuário digite um número qualquer e verifique se esse número digitado é **par** ou **impar**:

3. Crie um programa que gere um número inteiro aleatório entre 1-10 e você deve adivinhar que número foi escolhido pelo computador. O seu programa deve ser capaz de dizer se o seu “chute” foi certo (acertou o número), se foi maior ou menor.

DICA: importe a função **randint()**, faça o comando **from random import randint** na primeira linha do código, olhe abaixo como usar:

```
1 from random import randint
2
3 numero_aleatorio = randint(1,10)
```