



# Introdução a Python

Dia 2 <3

# Recapitulando:

- Conhecemos Colab
- Tipos básicos
- Operadores
- Variáveis
- Entradas e saídas
- Comentários
- Split



# Estruturas condicionais

Uma Estrutura de Condição, como o próprio nome já diz, verifica a condição dos argumentos passados e, executa um comando caso a condição seja verdadeira.



# Algoritmo

[ Uma "receita" para executarmos uma tarefa ou resolver algum problema]

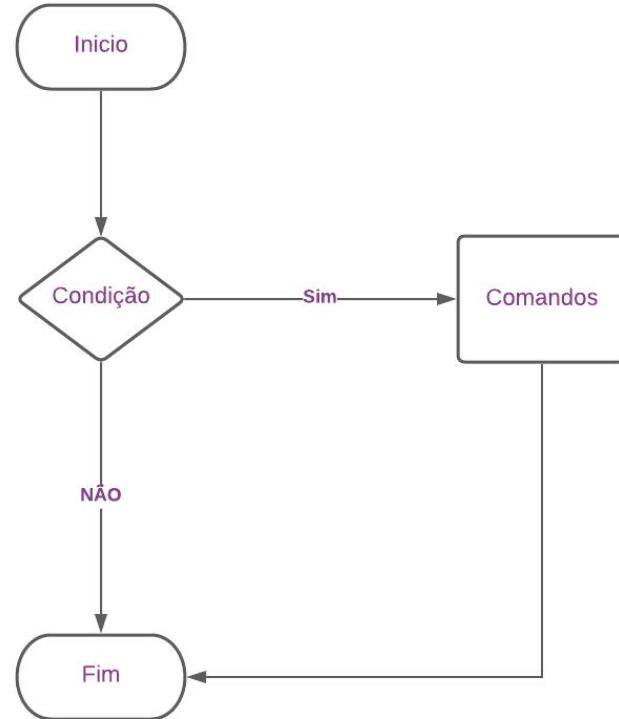


# Estruturas condicionais

Com as estruturas condicionais, podemos fazer com que algumas linhas do nosso algoritmo não sejam executadas.

Sintaxe de um **if simples**:

```
se condicao:  
    comando1  
    comando2
```



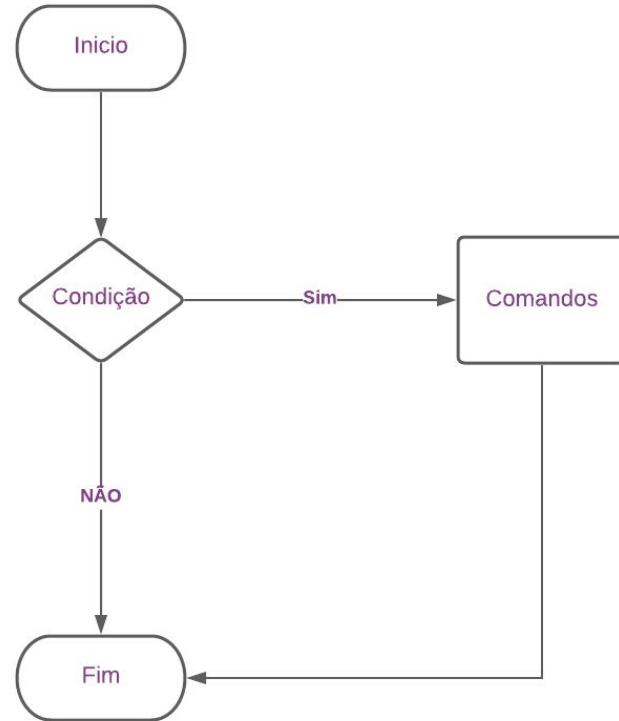
{:l

# Estruturas condicionais

Exemplo de um **if simples**:

Emitir um alerta de multa quando a velocidade de um carro estiver acima do permitido (70 km/h)

```
velocidade = float(input("Informe a velocidade: "))
if(velocidade > 70):
    print("você será multado")
print("fim")
```



# Álgebra booleana em Python

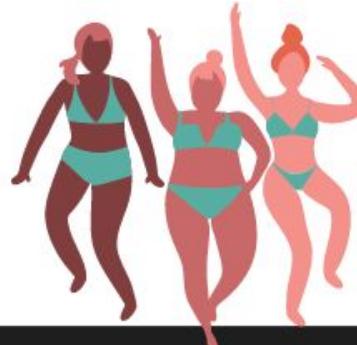
Para facilitar a estruturação das condições, podemos utilizar os operadores booleanos (Lembram da tabela da verdade?)

| Operador | Python | Exemplo | Descrição  |
|----------|--------|---------|--|
| E        | and    | P and Q | Se P e Q forem verdadeiros retorna True, se não retorna False  |
| OU       | or     | P or Q  | Se P ou Q forem verdadeiros retorna True, se não retorna False |
| Não      | not    | not Q   | Se Q é verdadeiro retorna False, se não retorna True           |



# Estruturas condicionais

Exemplos de um **if simples** em situações compostas:



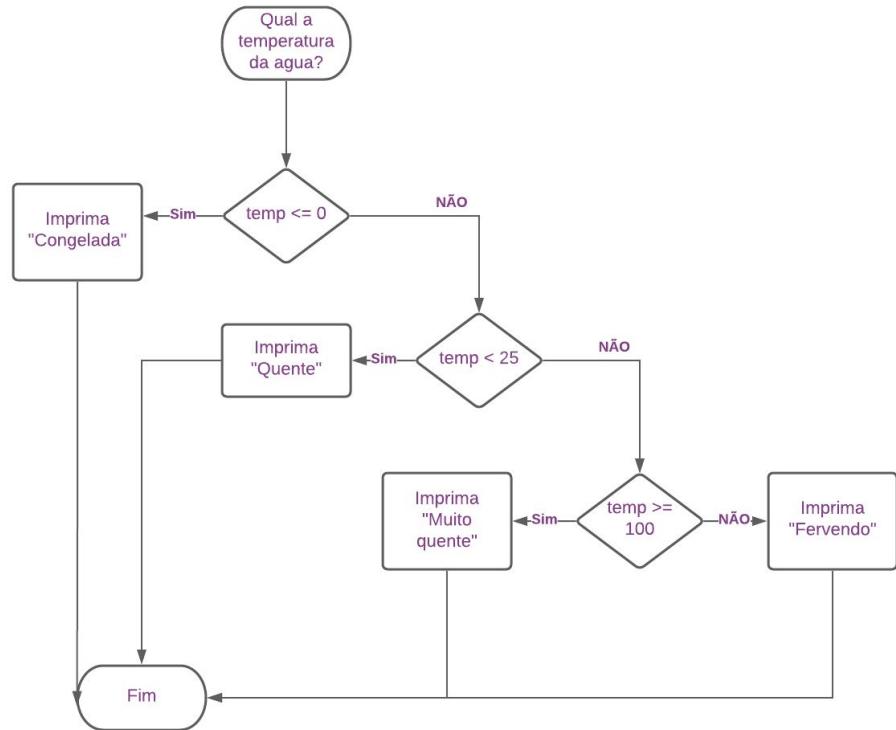
```
# Descubra a situação da massa corporal
massa = float(input("Informe o peso:"))
altura = float(input("Informe altura:"))
imc = massa / (altura*altura)
if (imc < 18):
    print("Abaixo do recomendado")
if (imc >= 18 and imc < 26):
    print ("Recomendado")
if (imc >=26):
    print("Acima do recomendado")
print("Fim!! PS: você é linda em qualquer imc!")
```



# Estruturas condicionais

Sintaxe de um if composto:

```
# Sintaxe de um if composto
se condição
    comando1
    comando2
senao
    comando3
```



# Estruturas condicionais

Exemplo de um **if composto** (que tem o else):

```
# Imprimir o maior de dois números
n1 = float(input('Informe o primeiro número:'))
n2 = float(input('Informe o segundo número:'))
if(n1 > n2):
    print(n1)
else:
    print(n2)
```



# Encadeamento de comandos condicionais

```
# Descubra a massa corporal, utilizando encadeamento de comandos
# Encadeamento de comandos seria, colocar um if dentro de um outro if

massa = float(input("Informe o peso:"))
altura = float(input("Informe altura:"))
imc = massa / (altura*altura)
if (imc < 18):
    print("Abaixo do peso!")
else:
    if(imc >= 18 and imc < 26):
        print ("Peso normal!")
    else:
        print("Acima do peso!")
print (imc)
print ("Fim")
```



# Encadeamento de comandos condicionais, com “elif”

```
# Descubra a massa corporal, utilizando encadeamento de comandos
# utilizando o elif

massa = float(input("Informe o peso:"))
altura = float(input("Informe altura:"))
imc = massa / (altura*altura)
if (imc < 18):
    print("Abaixo do peso!")
elif (imc >= 18 and imc < 26):
    print ("Peso normal!")
else:
    print("Acima do peso!")
print (imc)
print ("Fim")
```



# Indentação e blocos de código

- No Python, tudo que vem depois de ":" é um bloco de código;
- É como se existisse uma hierarquia dentro do código;
- Se os blocos não forem indentados, irá ocorrer erro no console.

Tudo dentro do while é um bloco

```
x = 0
while x < 20:
    print(x)
    x += 2
```

Indentação é o espaçoamento em relação ao primeiro nível hierárquico

```
# Exemplo de código não indentado:
y = 20
if y % 2 == 0:
print('y é par')
else:
print('y é ímpar')

# O console retorna:
#     print('y é par')
#
# IndentationError: expected an indented block
```

No Python, os Blocos precisam ser indentados



# Exercícios em Python [6]

Dez pessoas assinalaram que iriam participar do encontro do Maria vai com as Devs hoje. Faça um programa que leia o número de pessoas presentes e escreva “Sucesso!” se todas as pessoas compareceram, e “Alguém faltou :/” caso contrário. Se houverem mais de dez pessoas imprima “Temos um intruso entre nós :v”.



# Exercícios em Python [6]

```
marias = int(input("Digite o numero de mulheres presentes:"))
if marias == 10:
    print("Sucesso")
elif marias > 10:
    print("Temos um intruso entre nós :v")
else:
    print("Alguém faltou :/")
```

```
Digite o numero de mulheres presentes:2
Alguém faltou :/
```



# Exercícios em Python [7]

Receba duas notas, uma é de uma prova e outra é de um trabalho, por fim retorne:

- Se a nota da prova for maior ou igual a 5 e nota do trabalho maior que 6: “Aprovado”;
- Senão: “Reprovado”.



# Exercícios em Python [7]

```
nota_prova = float(input("Digite a nota de sua prova:"))
nota_trabalho = float(input("Digite a nota de seu trabalho:"))
if nota_prova >= 5 and nota_trabalho > 6:
    print ("Aprovado")
else:
    print ("Reprovado")
```

```
Dígitte a nota de sua prova:5
Dígitte a nota de seu trabalho:3
Reprovado
```



# Exercícios em Python [8]

A Serasa anualmente promove a ida dos desenvolvedores para a Python Brasil, mas como temos um número limitado de ingressos disponíveis, pode ser necessário decidir quem irá. Como critério de desempate sobre os ingressos, **quem nunca participou do evento terá prioridade**.

Receba o número de ingressos disponíveis, o número total de interessados e o número de interessados que já participaram do evento antes, e diga se é possível distribuir os ingressos apenas com essas regras, ou se será necessário um sorteio.



# Exercícios em python [8]

```
vagas = float(input("Número de vagas disponíveis:"))
ja_participaram = float(input("Número de pessoas que já participaram:"))
interessados = float(input("Número de pessoas interessadas:"))

if interessados == vagas:
    #há vagas o suficiente para todos os interessados
    print("Todos irão")
elif interessados - ja_participaram <= vagas:
    #De acordo com a regra de que, quem nunca participou tem prioridade, todos que nunca foram irão.
    print("Todos que nunca foram irão")
else:
    # Inicia que há mais interessados que nunca foram para a Python Brasil do que vagas.
    print("Haverá sorteio")
```

```
Número de vagas disponíveis:10
Número de pessoas que já participaram:2
Número de pessoas interessadas:10
Todos irão
```



# Exercícios em Python [9]

Nos últimos anos o Brasil está com muita variação de temperatura, e uma coisa louca acontecendo é que você pode passar por várias estações do ano num mesmo dia!

Para ajudar as pessoas a não pegar uma gripe ou uma insolação, você vai ler a temperatura e a umidade do ar da tabela do próximo slide, e criar um programa que classifica quais itens de sobrevivência são necessários para nossos dias de “verão”.



# Exercícios em Python [9]

| Temperatura às 8h - °C | Umidade do ar - % | Itens de sobrevivência               |
|------------------------|-------------------|--------------------------------------|
| 10°C < 15°C            | 0 < 40            | Blusa de frio e regata               |
| 10°C < 15°C            | 40 <= 100         | Blusa de frio, regata e guarda-chuva |
| 15°C < 20°C            | 0 < 40            | Blusa de frio e regata               |
| 15°C < 20°C            | 40 <= 100         | Blusa de frio, regata e guarda-chuva |
| 20°C < 25°C            | 0 < 40            | Regata                               |
| 20°C < 25°C            | 40 <= 100         | Regata e guarda-chuva                |



# Estruturas de Repetição

As estruturas padrões para repetição em Python são:

```
while (condição):
    # código

for item in iterável:
    # código
```

Observações:

- **Lembre-se:** Python utiliza indentação para definir blocos de código.
- O laço while executa um bloco de código quando determinada condição é atendida.
- O laço for, nos permite percorrer os itens de um iterável e para cada um deles, executar um bloco de código.



# Estruturas de Repetição

Exemplo:

```
▶ # Aqui repetimos o print 3 vezes
for n in range(0, 3):
    print(n)
# Output:
# 0
# 1
# 2

# Aqui iniciamos o n em 0, e repetimos o print até que
# seu valor seja maior ou igual a 3

n = 0
while n < 3:
    print(n)
    n += 1
# Output:
# 0
# 1
# 2
```



# Estruturas de Repetição

O loop for em Python itera sobre os itens de um conjunto, sendo assim, o range(0, 3) precisa ser um conjunto de elementos. E na verdade ele é:

```
list(range(0, 3))  
[0, 1, 2]
```



Iterando uma lista:

```
lista = [1, 2, 3, 4, 10]
for numero in lista:
    print(numero ** 2)
```

```
1
4
9
16
100
```

Se aplica a strings  
também:

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

```
c
a
s
a
```



# Auxiliares dos repetidores

- **break:** É usado para sair de um loop, não importando o estado em que se encontra.
- **continue:** Funciona de maneira parecida com a do break, porém no lugar de encerrar o loop, ele faz com que todo o código que esteja abaixo (porém ainda dentro do loop) seja ignorado e avança para a próxima iteração.

```
"""
Esse código deve rodar até que a palavra "sair" seja digitada.
* Caso uma palavra com 2 ou menos caracteres seja digitada, um aviso
deve ser exibido e o loop será executado do início (devido ao
continue), pedindo uma nova palavra ao usuário.
* Caso qualquer outra palavra diferente de "sair" seja digitada, um
aviso deve ser exibido.
* Por fim, caso a palavra seja "sair", uma mensagem deve ser exibida e o
loop deve ser encerrado (break).
```

Ou seja: `continue`, vai para a proxima iteração e o `break` para tudo.

```
"""
while True:
    string_digitada = input("Digite uma palavra: ")
    if string_digitada.lower() == "sair":
        print("Fim!")
        break
    if len(string_digitada) < 2:
        print("String muito pequena")
        continue
    print("Tente digitar \"sair\"")
```

```
Digite uma palavra: a
String muito pequena
Digite uma palavra: b
String muito pequena
Digite uma palavra: slaodkd
Tente digitar "sair"
Digite uma palavra: sair
Fim!
```

# Exercícios em Python [11]

Leia números enquanto a soma dos números lidos seja menor que 20.



# Exercícios em Python [11]

```
soma = 0

while soma < 20:
    soma += int(input("Digite o próximo número:"))

print(soma)
```

```
Digite o próximo número:3
Digite o próximo número:5
Digite o próximo número:3
Digite o próximo número:12
23
```





Link do material de  
estruturas  
condicionais



Maria  
vai com as  
**Devs**

# Instalando o Python e a nossa IDE (Pycharm Community)

[Link para o passo-a-passo](#)



# Manipulação de Strings

# Exercícios em Python [9]

```
temperatura = float(input("Temperatura:"))
umidade = float(input("Umidade:"))

if temperatura <= 10 or temperatura > 25:
    #Como não há instruções do que vestir nestas temperaturas, fizemos um tratamento para retornar algo para o usuário.
    print("Não saia de casa, você ficará doente")
else:
    if temperatura < 15:
        # Entra no loop se a temperatura for menor que 15
        if umidade <= 40:
            print("Blusa de frio e regata")
        else:
            print("Blusa de frio, regata e guarda-chuva")
    elif temperatura < 20:
        # Senão, entra no loop se a temperatura for menor que 20
        if umidade <= 40:
            print("Blusa de frio e regata")
        else:
            print("Blusa de frio, regata e guarda-chuva")
    elif temperatura < 25:
        # Senão, entra no loop se a temperatura for menor que 25
        if umidade <= 40:
            print("Regata")
        else:
            print("Regata e guarda-chuva")

Temperatura:20
Umidade:40
Regata
```



# Strings

| Símbolo | Significado  | Exemplo                         | Resultado |
|---------|--------------|---------------------------------|-----------|
| +       | Concatenação | “bolinho de” + “chuva”+ “quero” | ?         |
| *       | Repetição    | “Maria vai com as devs”*4       | ?         |
| []      | Indexação    | “chuva”[1]                      | ?         |
| [:]     | Fatiamento   | “chuva”[0:1]                    | ?         |
| {}      | Substituição | “chuv{}”.format(“inha”)         | ?         |
| in      | Verificação  | “chuv” in “chuvinha”            | ?         |
| not in  | Verificação  | “chuva” in “chuvinha”           | ?         |



# Strings - Símbolos

| Símbolo | Significado  | Exemplo                           | Resultado                |
|---------|--------------|-----------------------------------|--------------------------|
| +       | Concatenação | “bolinho de ” + “chuva ”+ “quero” | “bolinho de chuva quero” |
| *       | Repetição    | “Maria”*4                         | “MariaMariaMariaMaria”   |
| []      | Indexação    | “chuva”[1]                        | “h”                      |
| [:]     | Fatiamento   | “chuva”[0:1]                      | “c”                      |
| {}      | Substituição | “chuv{}”.format(“inha”)           | “chuvinha”               |
| in      | Verificação  | “chuv” in “chuvinha”              | True                     |
| not in  | Verificação  | “chuva” not in “chuvinha”         | True                     |



# Strings - Funções

| Função     | Parâmetro  | Aplicação                               | Resultado             |
|------------|------------|---|-----------------------|
| upper      | nenhum     | “bolinho de chuva”.upper()              | BOLINHO DE CHUVA      |
| lower      | nenhum     | “BOLINHO DE CHUVA”.lower()              | “bolinho de chuva”    |
| capitalize | nenhum     | “bolinho de chuva”.capitalize()         | “Bolinho de chuva”    |
| count      | item       | “bolinho de chuva”.count(“h”)           | 2                     |
| replace    | (old, new) | “bolinho de chuva”.replace(“a”, “inha”) | “bolinho de chuvinha” |
| find       | item       | “bolinho de chuva”.find(“chu”)          | 1                     |
| len        | item       | len(“bolinho de chuva”)                 | 16                    |



# Exercícios em Python [10]

Você é o responsável por gerar os certificados dos da trilha financeira da serasa, porém, sabemos que eventualmente os participantes não cooperam e escrevem seus nomes sem seguir a norma padrão, apenas com a primeira letra maiúscula. Para evitar certificados despadronizados, você resolveu escrever um programinha em Python para normalizar os nomes para você!



# Exercícios em Python [10]

```
nome = input("Digite o nome do participante:")
print(f"{nome.capitalize()}")
```

```
Digite o nome do participante:aliniiiii
Aliniiiii
```



# Exercícios em Python [12]

Possuímos uma lista de cinco desenvolvedoras que participam do programa Maria vai com as devs.

Construa um programa que descubra quantas delas possuem o nome começando com a letra “A”.

As desenvolvedoras se chamam: Andorinha, Azaléia, Amélia, Margarida e Rosa.



# Exercícios em Python [12]

```
devs = ['amelia', 'azaleia', 'andorinha', 'margarida', 'rosa']
nome_com_a = 0

for nome in devs:
    if nome[0] == 'a':
        nome_com_a += 1

print(f"Foram encontradas {nome_com_a} desenvolvedoras que começam com a letra A")
```

```
Foram encontradas 3 desenvolvedoras que começam com a letra A
```

