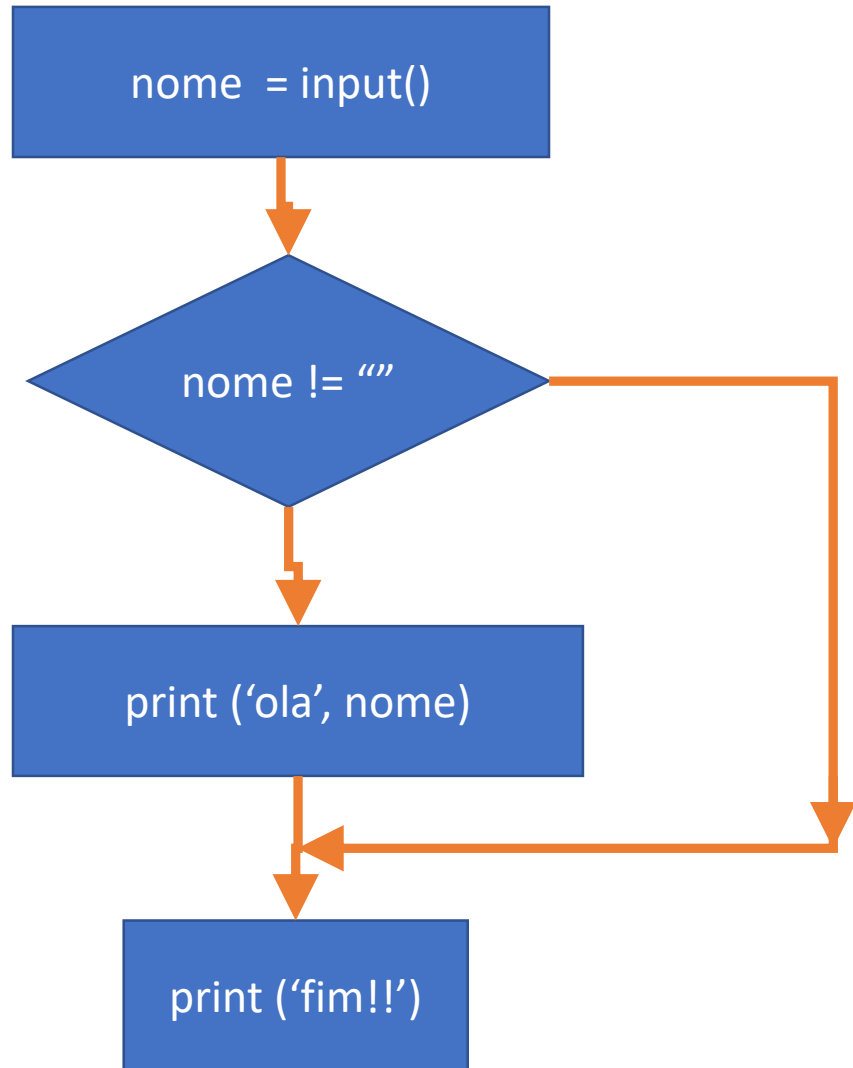


# Python: Comando condicional

**Galileu** Batista de Sousa  
Galileu.batista -at +ifrn -edu +br

# A ideia de execução condicional



```
nome = input()
if nome != '':
    print('ola', nome)
print('fim')
```

# Ifs em sequência – sem else



```
salario = int (input('Salario'))

if salario >= 10000 :
    print('Rico')
    print('Vive bem, hein?')

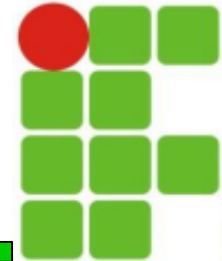
print ('E agora, Jose?')

if salario < 10000:
    print('Classe media. Lute.')

print ('Ja não pode beber.')
```

- Identação
  - Subordinação
- : finalizando o if
- Lógica
  - Implementada com negação da condição

# Identação



- Estabelece a subordinação de comandos
  - Mais de um comando subordinado, mesma identação
- Alinhamento com comando subordinador
  - Acabou a subordinação
- Use **quatro espaços** com identação (**nunca tab**)
  - Modelo mental para deixar claro quem manda.

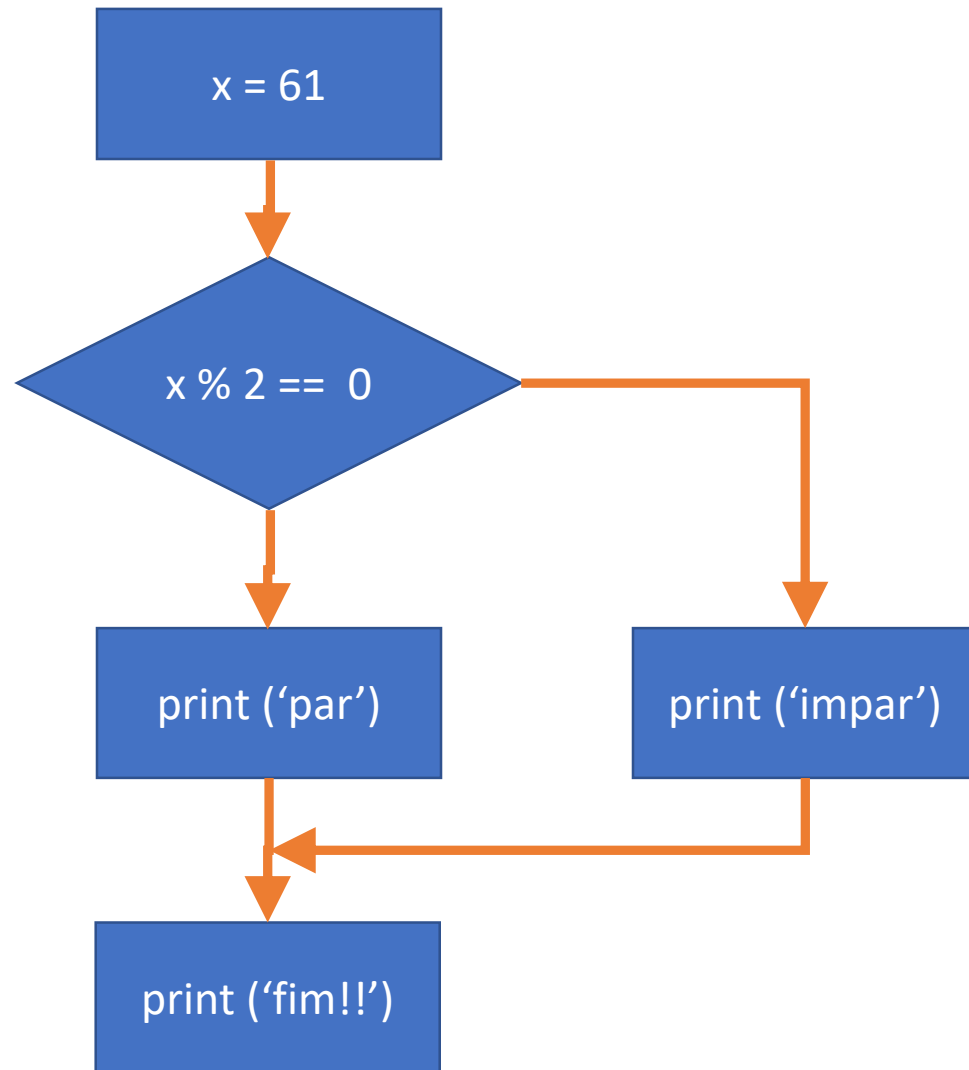
# Ifs em sequência – sem else



```
salario = int (input('Salario'))
if salario > 10000 :
    print('Rico')
    print('Vive bem, hein?')
if salario > 6000 :
    print('Classe media alta')
if salario >= 4000:
    print('Classe media')
if salario == 3000:
    print('Classe media baixa')
if salario <= 2000 :
    print('Classe baixa')
if salario < 300:
    print('Desassistido')
```

- Identação
  - Subordinação
- : finalizando o if
- Lógica
  - os **ifs** propriamente não estão subordinados

# A ideia de execução condicional



```
x = 5
if x % 2 == 0 :
    print ('par')
else :
    print ('impar')
print ('fim!!')
```

# Aninhamento



```
salario = int (input('Salario'))

if salario >= 10000 :
    print('Rico')
    if salario > 39000:
        print('Mais que rico. $$$$')
else:
    print ('É possível chegar lá.')
    print ('Não desista.')

print ('Ja não pode beber.')
```

- Identação
  - else corresponde ao if a que está alinhado
- Boa prática:
  - Manter mesmo alinhamento de subordinação no if e no else

# else na parte if



```
salario = int (input('Salario'))

if salario < 10000:
    print ('É possível chegar lá.')
    print ('Não desista.')
else:
    if salario > 39000:
        print('Mais que rico. $$$$')
    else:
        print('Rico')
print ('Ja não pode beber.')
```

- É uma situação muito comum
- Python suporta:
  - `elif`
  - Como se juntasse:
    - “Se não” e “se”
- Vejamos ....



# else com if - elif



```
salario = int (input('Salario'))

if salario < 10000:
    print ('É possível chegar lá.')
    print ('Não desista.')
elif salario > 39000:
    print('Mais que rico. $$$$')
else:
    print('Rico')
print ('Ja não pode beber.')
```

- Observe o alinhamento
- O `else` só é executado quando os dois `ifs` falham

# Operadores relacionais



- Avaliadas antes do seu uso:
  - Em uma condição de if:
- Principais operadores:

==	igual
!=	diferente
<	menor
<=	menor ou igual
>	maior
>=	maior ou igual

## Precedência

()	é a maior
**	a seguinte
+ -	menor
relacionais	ainda menor

## Associatividade

**	à direita
outros	à esquerda

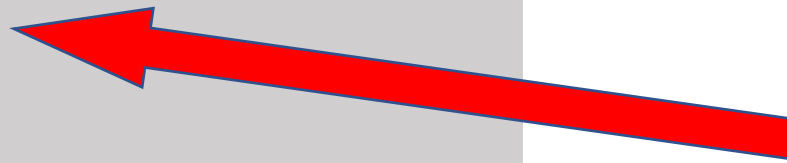
# Erros 'imprevisíveis'



- Alguns erros não são facilmente previsíveis
  - Ou melhor, seria trabalhoso testar cada detalhe....
- O que acontece se o usuário **digitar uma palavra?**

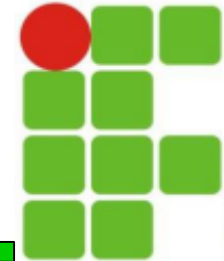
```
print ('Salario?')
salario = int (input())

if salario > 10000:
    print ('É possível chegar lá.')
    print ('Não desista.') else:
print ('Ja não pode beber.')
```



Não tem muito o que  
fazer....  
O programa vai finalizar  
com um erro 'estranho'...

# Tentativa e erro....



- A ideia é proteger um trecho de código por **try/except**

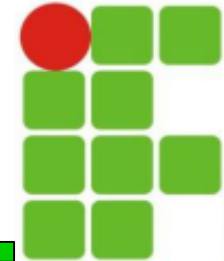
```
try:
    print ('Salario?')
    salario = int (input())

    print ('É bom ter salario')
    print (salario, 'não é pouco.')

except:
    print ("Digite um salario válido")
```

- O fluxo vai seguindo ...
- Se algum der erro ...
  - Direciona o fluxo para o estabelecido no **except**.
- Sem erro ...
  - Sem **except**

# Tentativa e erro....



```
try:
    print ('Salario?')
    salario = int (input())

    print ('É bom ter salario')
    print (salario, 'não é pouco.')

except:
    print ('Digite um salario válido')
print ('fim')
```

