

ICC901 – Introdução à Programação de Computadores
IEC081 – Introdução à Ciência dos Computadores
IEC037 – Introdução à Programação de Computadores

Aula 03 – Estruturas Condicionais Encadeadas

Atualização: 6/mar/20

UFAM

ICOMP
Instituto de Computação

Fonte da imagem: https://www.codeschool.com/courses/forging-ahead-with-asp-net-core

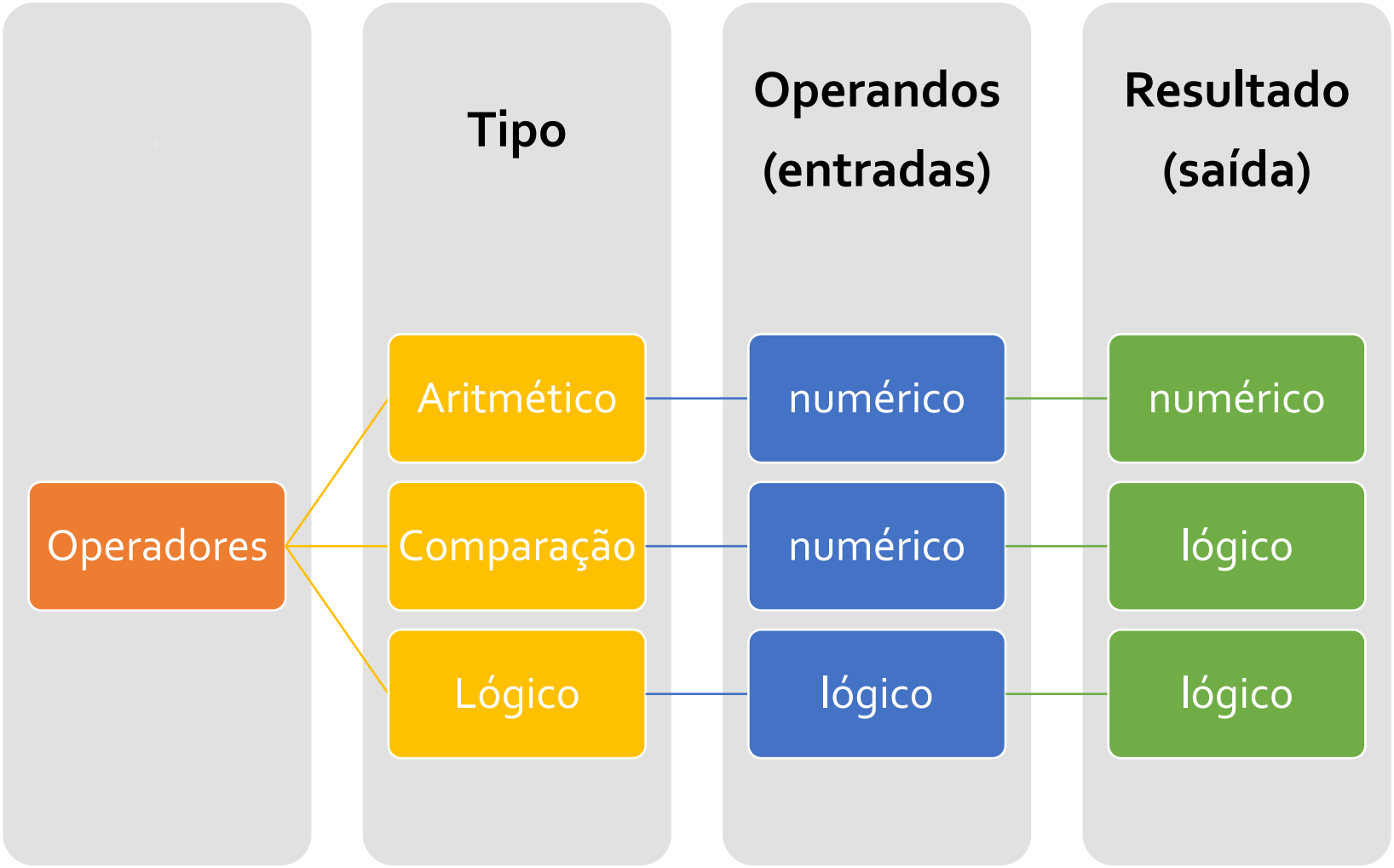
Conteúdo

- Como montar uma condição?
- Estruturas Condicionais Encadeadas
- Exemplos extras

Conteúdo

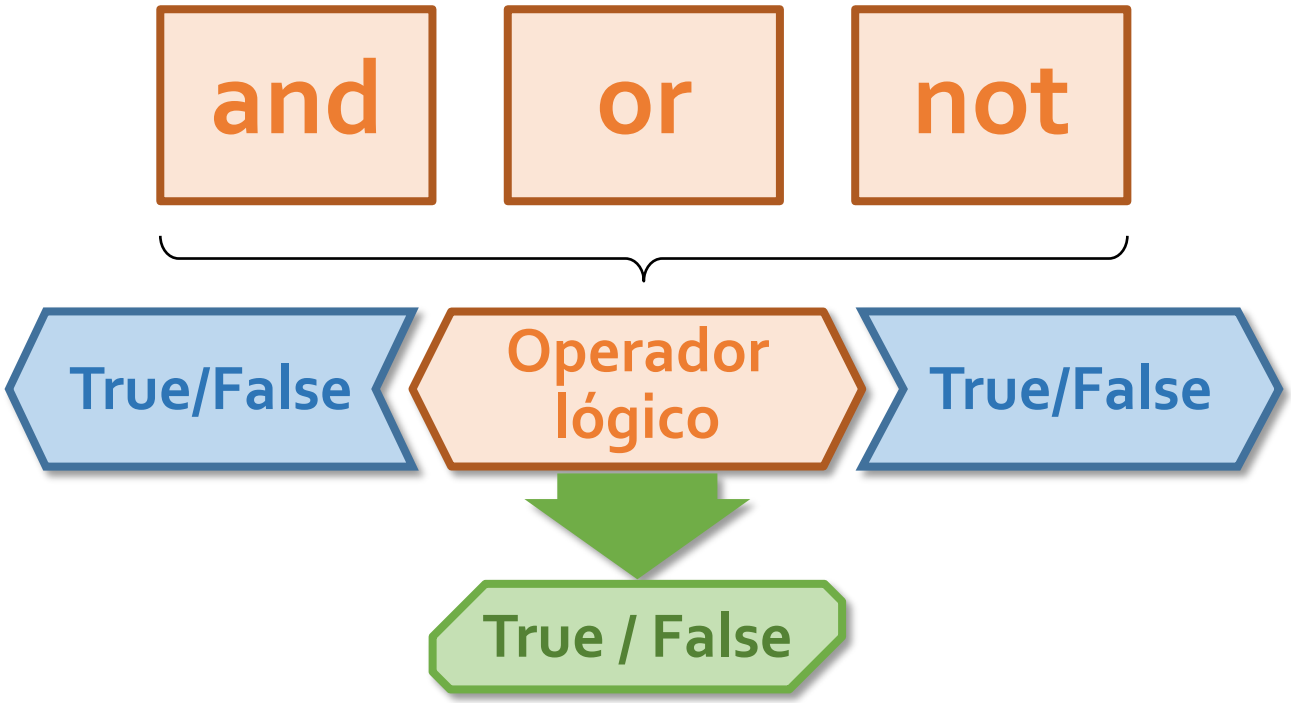
- Como montar uma condição?
- Estruturas Condicionais Encadeadas
- Exemplos extras

Tipos de operadores



Operadores Lógicos

- Os operadores lógicos (ou booleanos) são utilizados para combinar condições.



AND ("E" lógico)

- Resulta em verdade somente se todas as condições são verdadeiras (válidas).

A	B	A and B
F	F	F
F	V	F
V	F	F
V	V	V



```
ladoA == ladoB and ladoC
```

Operador and liga condições, e não números

Arquivo Editar Buscar Executar Ferramentas

Python 3 | main.py

```
# Testa se triangulo
eh equilatero

ladoA = 5
ladoB = 5
ladoC = 5

print(
(ladoA == ladoB) and
(ladoB == ladoC))
```

Console | Shell

```
True
```

OR ("Ou" lógico)

- Resulta em verdade se pelo menos uma (alguma) das condições é verdadeira.

A	B	A or B
F	F	F
F	V	V
V	F	V
V	V	V



```
ladoA == ladoB or ladoC
```

Operador or liga condições, e não números

Arquivo Editar Buscar Executar Ferramentas

Python 3 | main.py

```
# Testa se triangulo eh isosceles

ladoA = 4
ladoB = 5
ladoC = 5

print(
(ladoA == ladoB) or
(ladoB == ladoC) or
(ladoA == ladoC))
```

Console | Shell

```
True
```

NOT ("Não" lógico)

- Nega o valor de uma condição.

A	not A
F	V
V	F



```
not (a = b)
```

Comparação deve ser formada por **DOIS** sinais de =

Arquivo Editar Buscar Executar Ferramentas

Python 3 | main.py

```
# Testa se a defesa
do personagem eh
diferente de
armadura

defesa = "escudo"

print(
not (defesa ==
"armadura"))
```

Console | Shell

```
True
```

Precedência entre operadores

Operador	Significado
`()`	Grupos entre parênteses
`-`	Inversão de sinal
`**`	Potenciação
`* / % //`	Multiplicação, divisões, resto
`+ -`	Subtração, adição
`> >= < <= == !=`	Comparações
`not`	NÃO lógico
`and`	E lógico
`or`	OU lógico
`=`	Atribuição

Na dúvida, use parênteses

Operadores Lógicos :: Exemplo A

```
x <= 20 or x >= 40
```

```
salario > 1000 and
idade > 18
```

x	resultado
10	V
20	V
30	F
40	V
50	V

salario	idade	resultado
900	18	F
1000	19	F
1100	17	F
1200	22	V

Operadores Lógicos :: Exemplo B

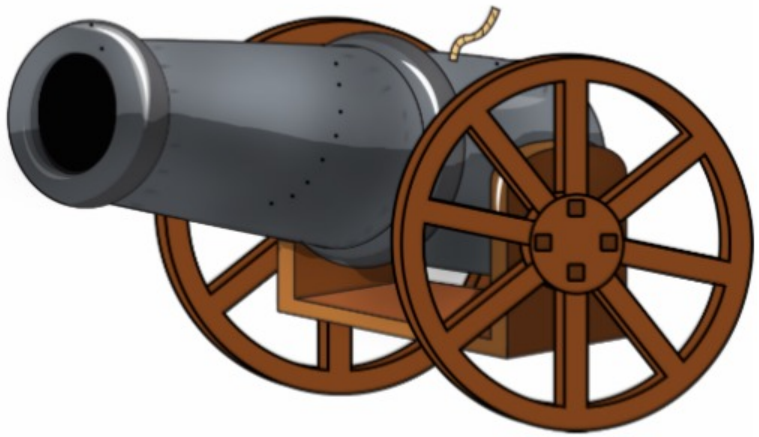
```
num % 2 != 0 and
contador < 50
```

num	contador	resultado
1231	51	F
1232	50	F
1233	49	V
1234	48	F

Problema 1

- Calcular o alcance S de um projétil.
- Entradas:
  - Velocidade inicial v_0
  - Ângulo θ entre o cano do canhão e o solo
- Considere:
  - $g = 9{,}81 m/s^2$.

$$S = \frac{v_0^2}{g} \cdot \operatorname{sen}(2\theta)$$

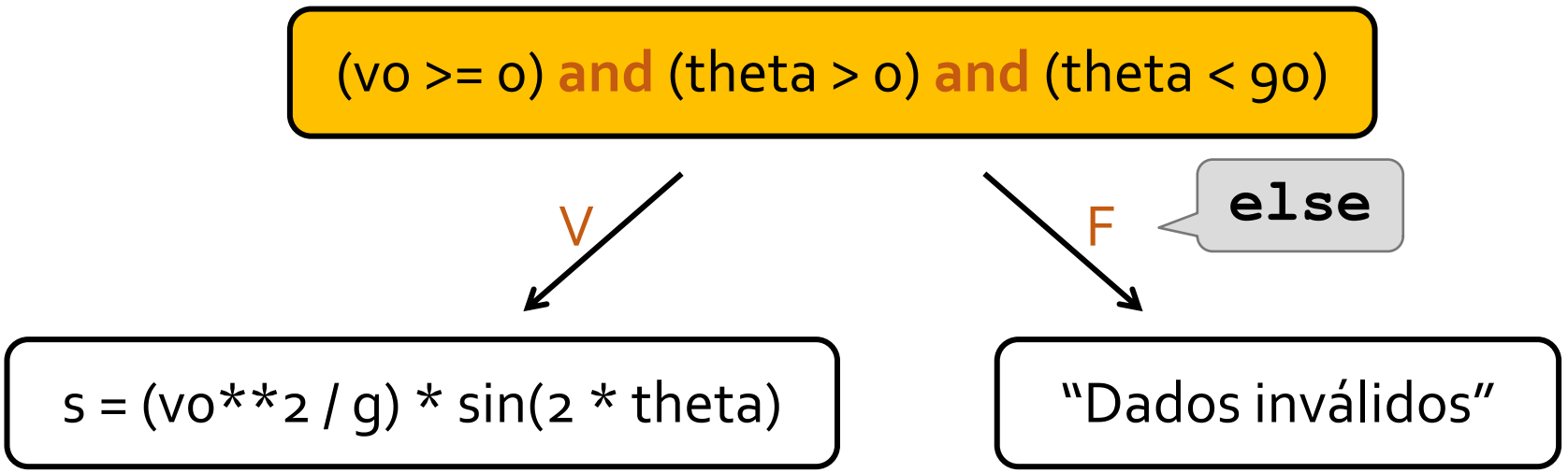


Problema 1
2 – Identifique as entradas e saídas

	Grandeza	Unidade de medida	Faixa de valores
Entradas	Velocidade inicial	m/s	>= 0
	Ângulo com solo	graus	]0; 90[
Saídas	Alcance	m	> 0

Problema 1
3 – Projete o script – versão 1

- Em que condição as entradas são **válidas**?



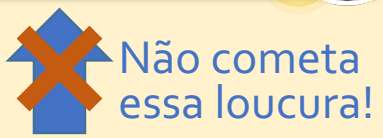
Problema 1
4 – Codifique em Python – versão 1

```
# Entrada de dados e definicao de constantes
v0    = float(input("Velocidade inicial: "))
theta = float(input("Angulo (em graus): "))
g     = 9.81

# Importar modulo matematico
from math import *

if (v0 >= 0) and (theta > 0) and (theta < 90):
    theta_rad = radians(theta)
    s = (v0**2 / g) * sin(2 * theta_rad)
    print(round(s, 3))
else:
    print("Dados invalidos.")
```





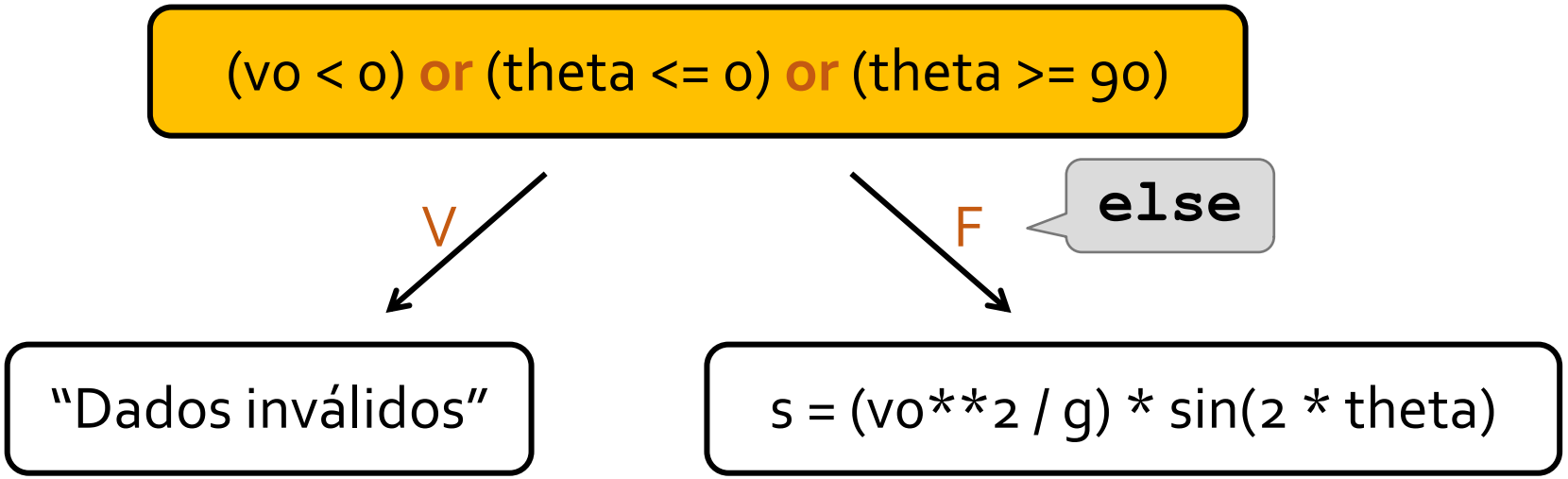
Existe outra forma de resolver?

- Para algumas pessoas, é mais fácil pensar nas condições **válidas** do problema.
- Para outras, nas condições **inválidas**.
- De qualquer maneira, ambas as soluções devem ser corretas, pois o problema é o mesmo.



Problema 1
3 – Projete o script – versão 2

- Em que condição as entradas **não são válidas**?



Como negar expressões lógicas?

not (A **and** B **and** C **and** ...)



not (A) **or not** (B) **or not** (C) **or not** (...)

not (A **or** B **or** C **or** ...)



not (A) **and not** (B) **and not** (C) **and not** (...)

Problema 1
4 – Codifique em Python – versão 2

```
# Entrada de dados e definicao de constantes
v0    = float(input("Velocidade inicial: "))
theta = float(input("Angulo (em graus): "))
g     = 9.81

# Importar modulo matematico
from math import *

if (v0 < 0) or (theta <= 0) or (theta >= 90):
    print("Dados invalidos.")
else:
    theta_rad = radians(theta)
    s = (v0**2 / g) * sin(2 * theta_rad)
    print(round(s, 3))
```

Medidas são valores **reais**

Inverteram-se as condições

Problema 1
5 – Teste o script

v_0: -1
ângulo: 45

Velocidade inválida

v_0: 2
ângulo: 99

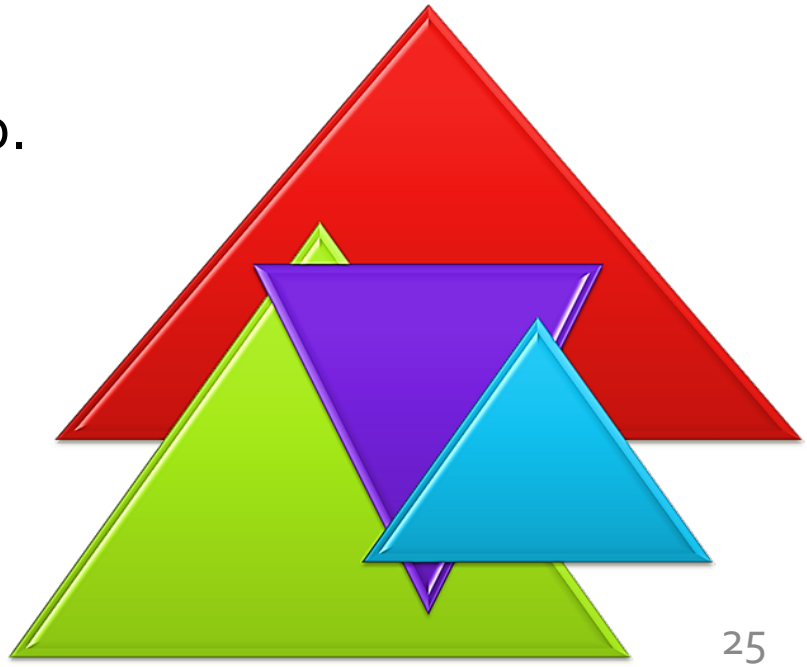
Ângulo inválido

v_0: 5
ângulo: 40

Entradas **válidas**

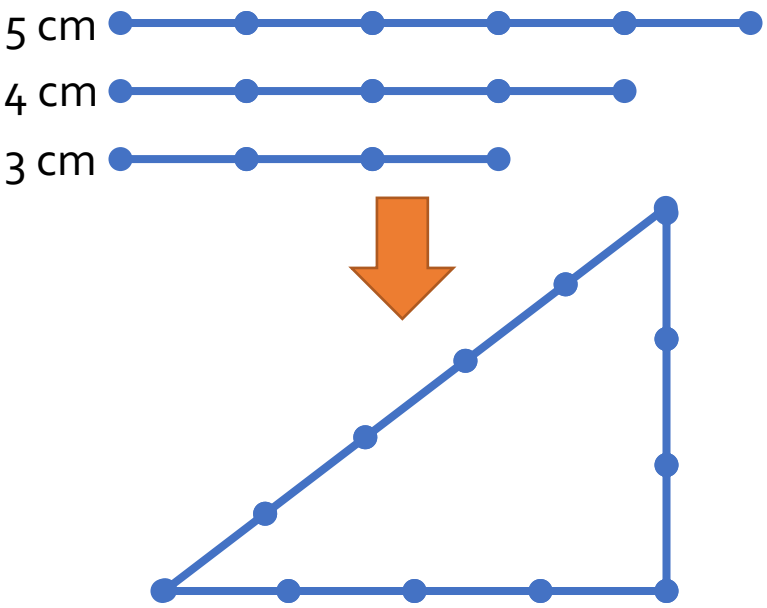
Problema 2

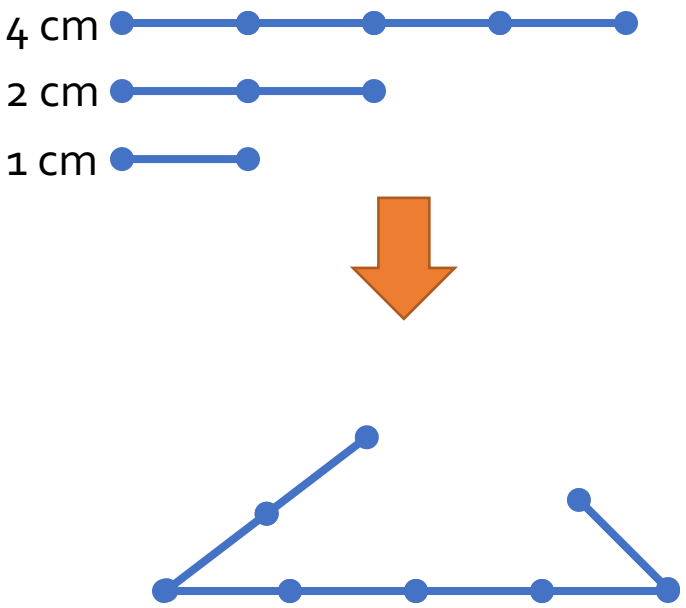
- Dados três valores x, y e z, verifique:
  1. Se eles podem ser os comprimentos dos lados de um triângulo.
  2. Caso positivo, se o triângulo é equilátero, isósceles ou escaleno.
  3. Imprimir nomes sem acento.



Problema 2
1 – Entenda o problema (1/2)

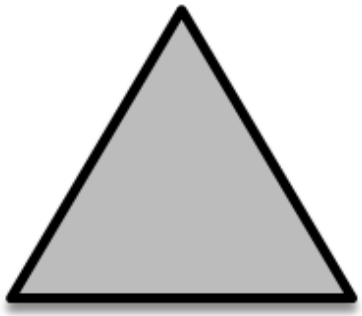
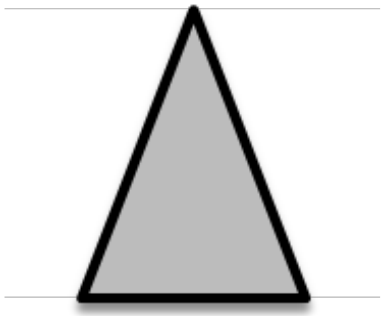
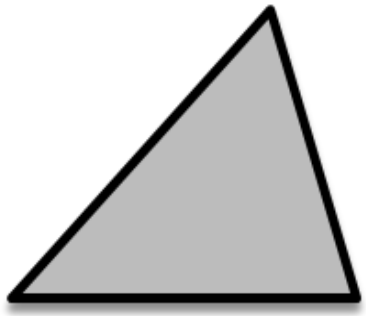
1. Verificar se três medidas formam um triângulo válido:
   - Nenhum dos lados de um triângulo deve ser maior ou igual que a soma dos outros dois.
   - Exemplos:





Problema 2
1 – Entenda o problema (2/2)

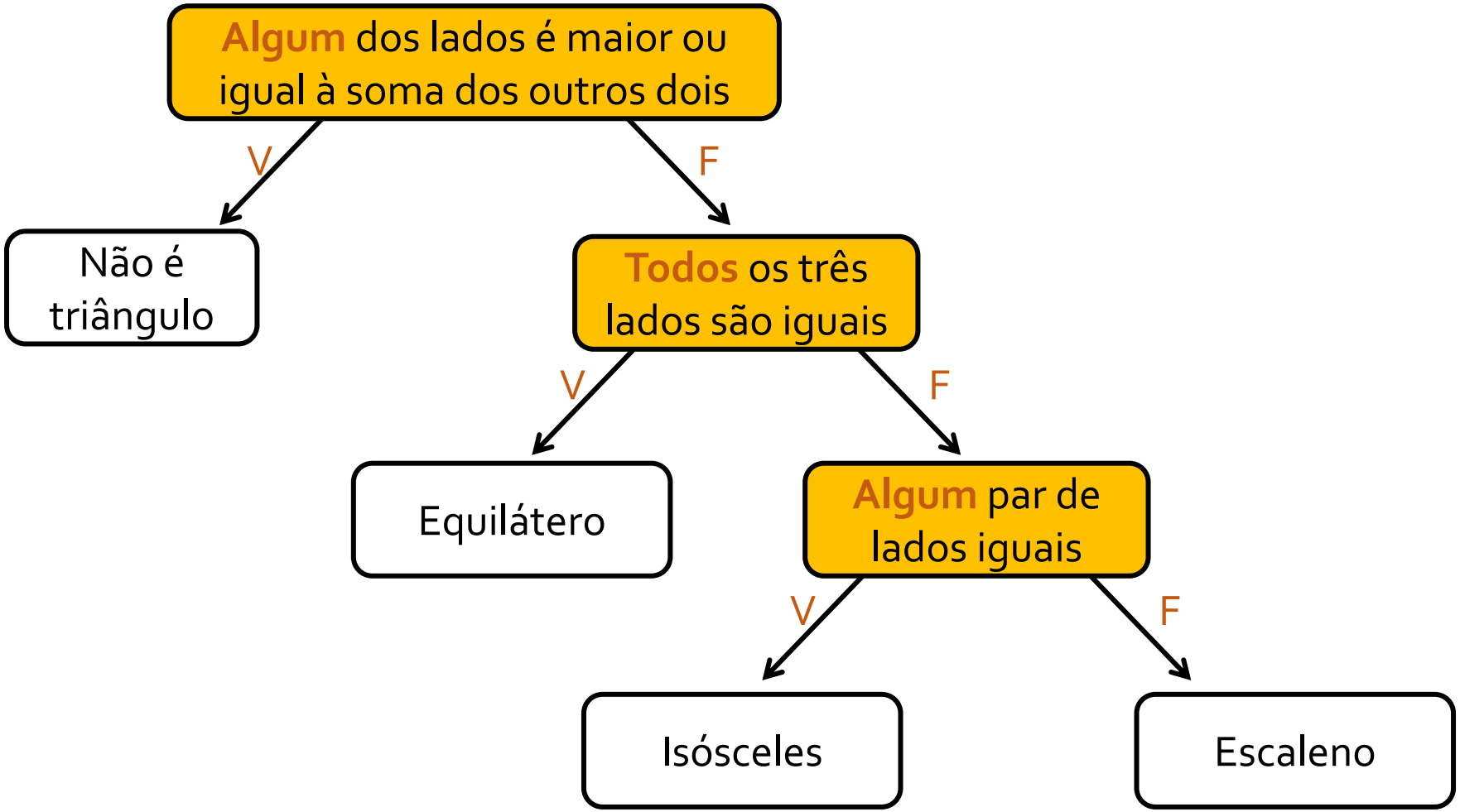
2. Verificar quantos lados são iguais entre si.
   - Exemplos:

Equilátero	Isósceles	Escaleno
		
Triângulo cujos lados têm comprimentos iguais	Triângulo que tem dois lados com comprimentos iguais	Triângulo que tem os três lados com comprimentos diferentes

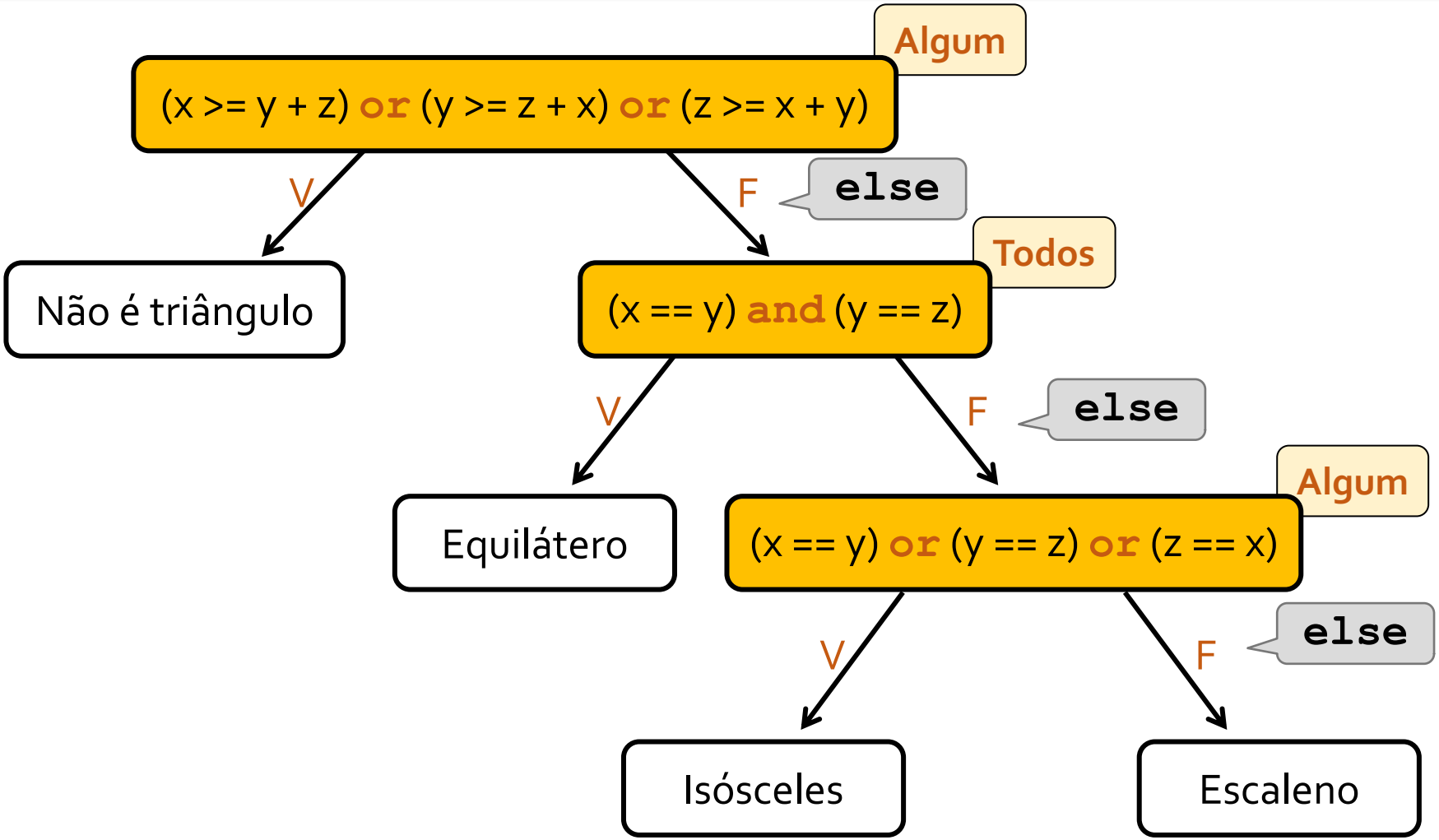
Problema 2
2 – Identifique as entradas e saídas

	Grandeza	Unidade de medida	Faixa de valores
Entradas	x	metro	> 0
	y	metro	> 0
	z	metro	> 0
Saídas	mensagem	---	"Não é triângulo", "equilátero", "isósceles", "escaleno"

Problema 2
3 – Projete o script – versão 1



Problema 2
3 – Projete o script – refinamento



Problema 2
4 – Codifique em Python – versão 1

```
# Entrada de dados
x = float(input("Lado x: "))
y = float(input("Lado y: "))
z = float(input("Lado z: "))

if ((x >= y + z) or (y >= x + z) or (z >= y + x)):
   print("Nao eh triangulo")
else:
    if ((x == y) and (y == z)):
        print("equilatero")
    else:
        if ((x == y) or (y == z) or (z == x)):
            print("isosceles")
        else:
            print("escaleno")
```

Medidas dos lados de um polígono são números **reais**

Atenção para uso de maiúsculas e minúsculas conforme manda enunciado

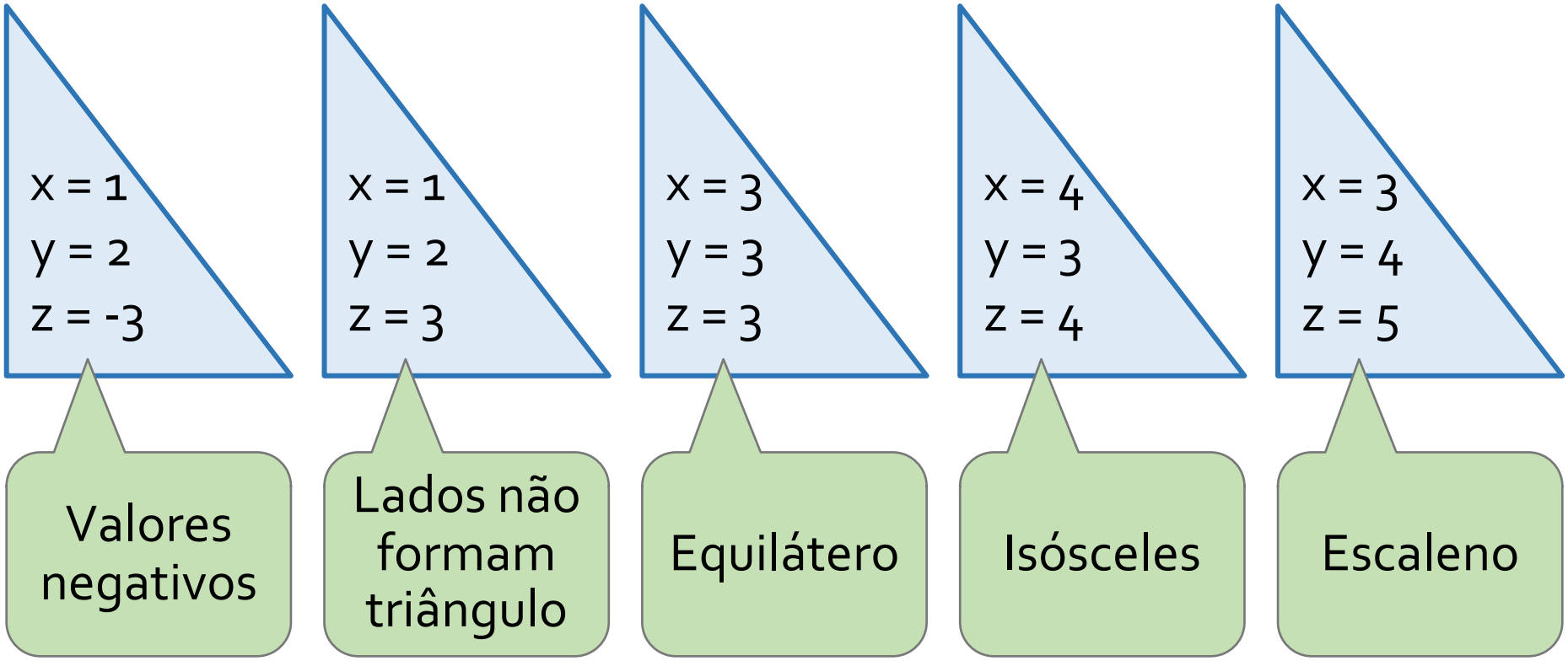
Problema 2
4 – Codifique em Python – versão 2

```
# Entrada de dados
x = float(input("Lado x: "))
y = float(input("Lado y: "))
z = float(input("Lado z: "))

if ((x < y + z) and (y < x + z) and (z < y + x)):
   if ((x != y) and (y != z) and (z != x)):
      print("escaleno")
   else:
      if ((x != y) or (y != z)):
         print("isosceles")
      else:
         print("equilatero")
else:
   print("Nao eh triangulo")
```

Inverteram-se as condições

Problema 2
5 – Teste o script



Conteúdo

- Como montar uma condição?
- **Estruturas Condicionais Encadeadas**
- Exemplos Extras

Estruturas Condicionais Encadeadas

- Estruturas condicionais encadeadas (ou aninhadas) são estruturas condicionais dentro de outras estruturas condicionais.
- Um longo encadeamento de **`if`**s e **`else`**s, em diversos níveis de indentação, pode confundir o programador.
- O comando **`elif`** substitui um par **`else`** / **`if`** sem criar um outro nível de indentação na estrutura condicional.

Estruturas Condicionais Encadeadas :: Exemplo

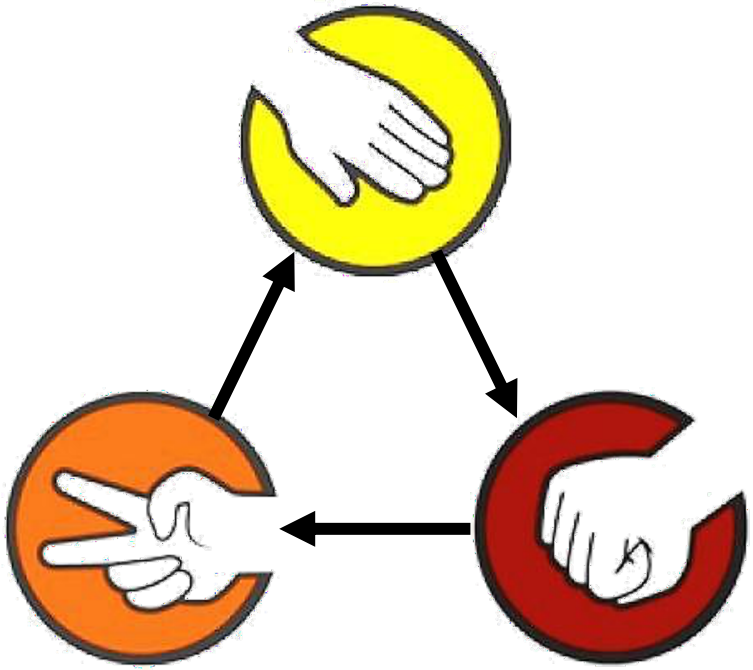
```
if (delta < 0):
    print("Nao tem raiz real")
else:
    if (delta == 0):
        r1 = -b/(2 * a)
    else:
        r1 = (-b+delta**0.5)/(2*a)
        r2 = (-b-delta**0.5)/(2*a)
```



```
if (delta < 0):
    print("Nao tem raiz real")
elif (delta == 0):
    r1 = -b/(2 * a)
else:
    r1 = (-b + delta**0.5)/(2*a)
    r2 = (-b - delta**0.5)/(2*a)
```

Problema 3

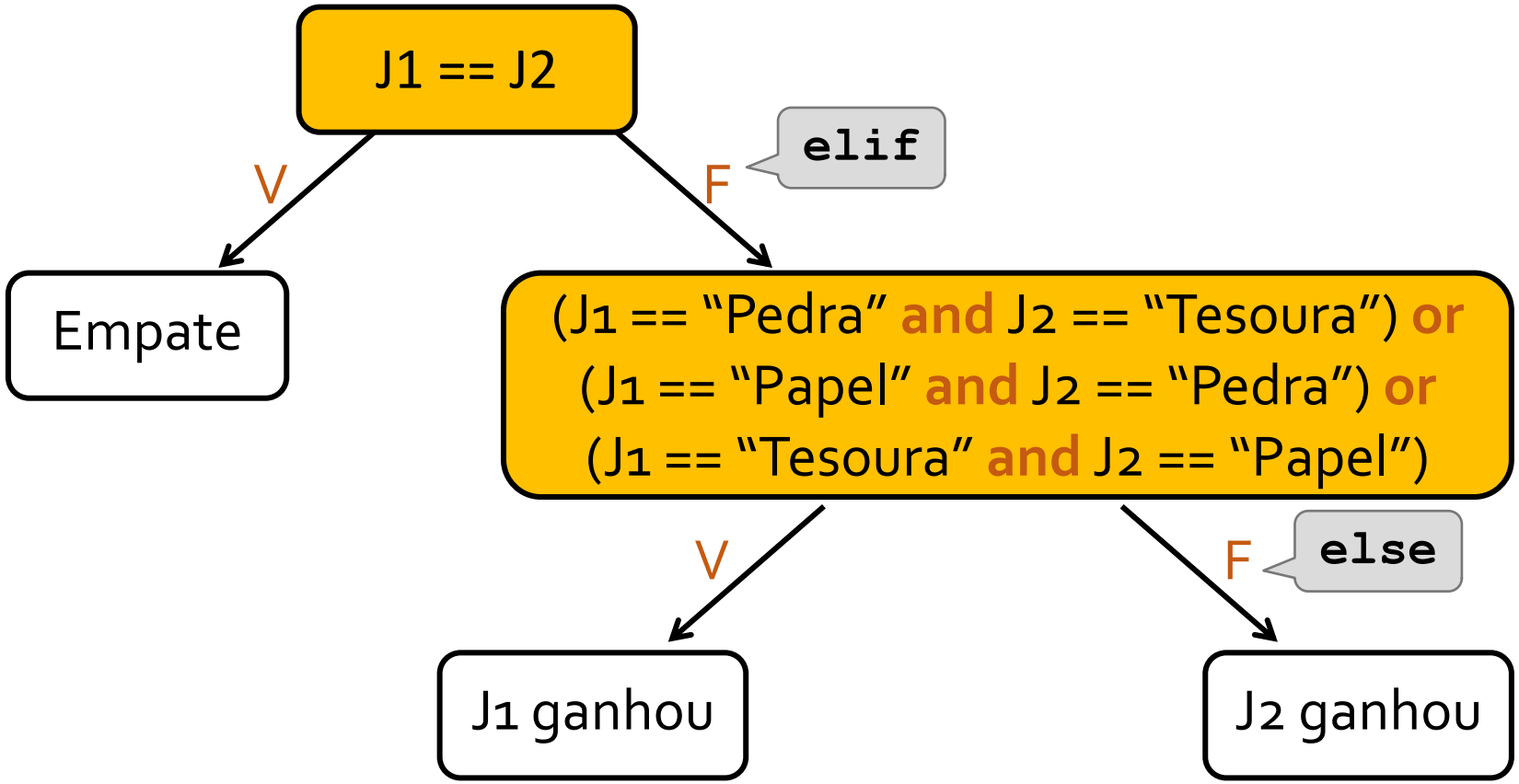
- Duas pessoas (J1 e J2) jogam pedra, papel, tesoura.
- Escreva um programa que leia a jogada de cada pessoa.
- Determine quem ganha ou se há empate.



Problema 3
2 – Identifique as entradas e saídas

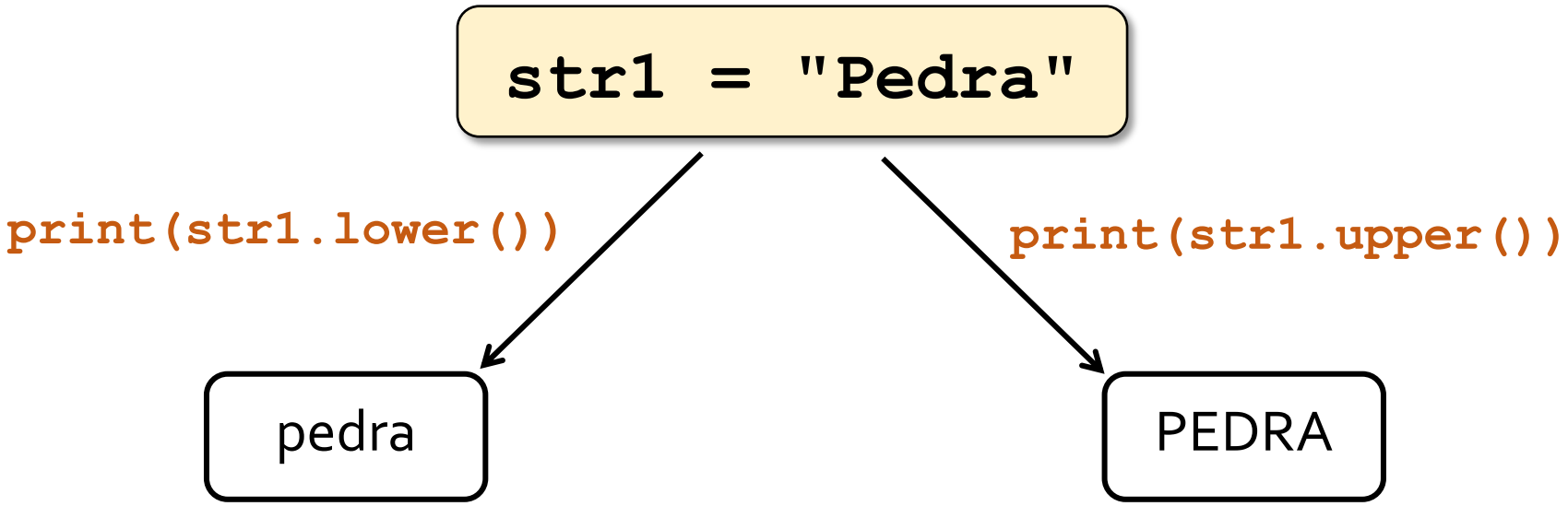
	Grandeza	Unidade de medida	Faixa de valores
Entradas	Mão do J1	---	{Pedra, Papel, Tesoura}
	Mão do J2	---	{Pedra, Papel, Tesoura}
Saídas	Vencedor	---	{J1, J2, Empate}

Problema 3
3 – Projete o script



Problema 3
Tratamento de maiúsculas e minúsculas

- É desejável garantir que a comparação das jogadas dê certo mesmo quando o usuário digita PEDRA, PaPeL ou TESOURa em padrões diferentes de caracteres maiúsculos ou minúsculos.



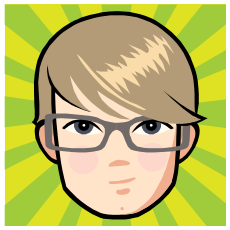
Problema 3
4 – Codifique em Python

```
# Leitura das entradas
j1 = input("Jogada do Jogador 1: ").lower()
j2 = input("Jogada do Jogador 2: ").lower()

if (j1 == j2):
    print("Empate")
elif (((j1 == "pedra") and (j2 == "tesoura")) or
((j1 == "papel") and (j2 == "pedra")) or ((j1 ==
"tesoura") and (j2 == "papel"))):
    print("Jogador 1 ganhou.")
else:
    print("Jogador 2 ganhou.")
```

Ajusta todas as letras de uma string para **minúsculas**

Problema 3
5 – Teste o script

J1 

J2 




J1 ganha




J2 ganha




Empate

Problema 4

- Anos bissextos são definidos da seguinte forma:
  1. Anos divisíveis por 400 são bissextos.
  2. Anos divisíveis por 100, mas não por 400, não são bissextos.
  3. Anos divisíveis por 4, mas não por 100, são bissextos.
  4. Todos os outros anos não são anos bissextos.



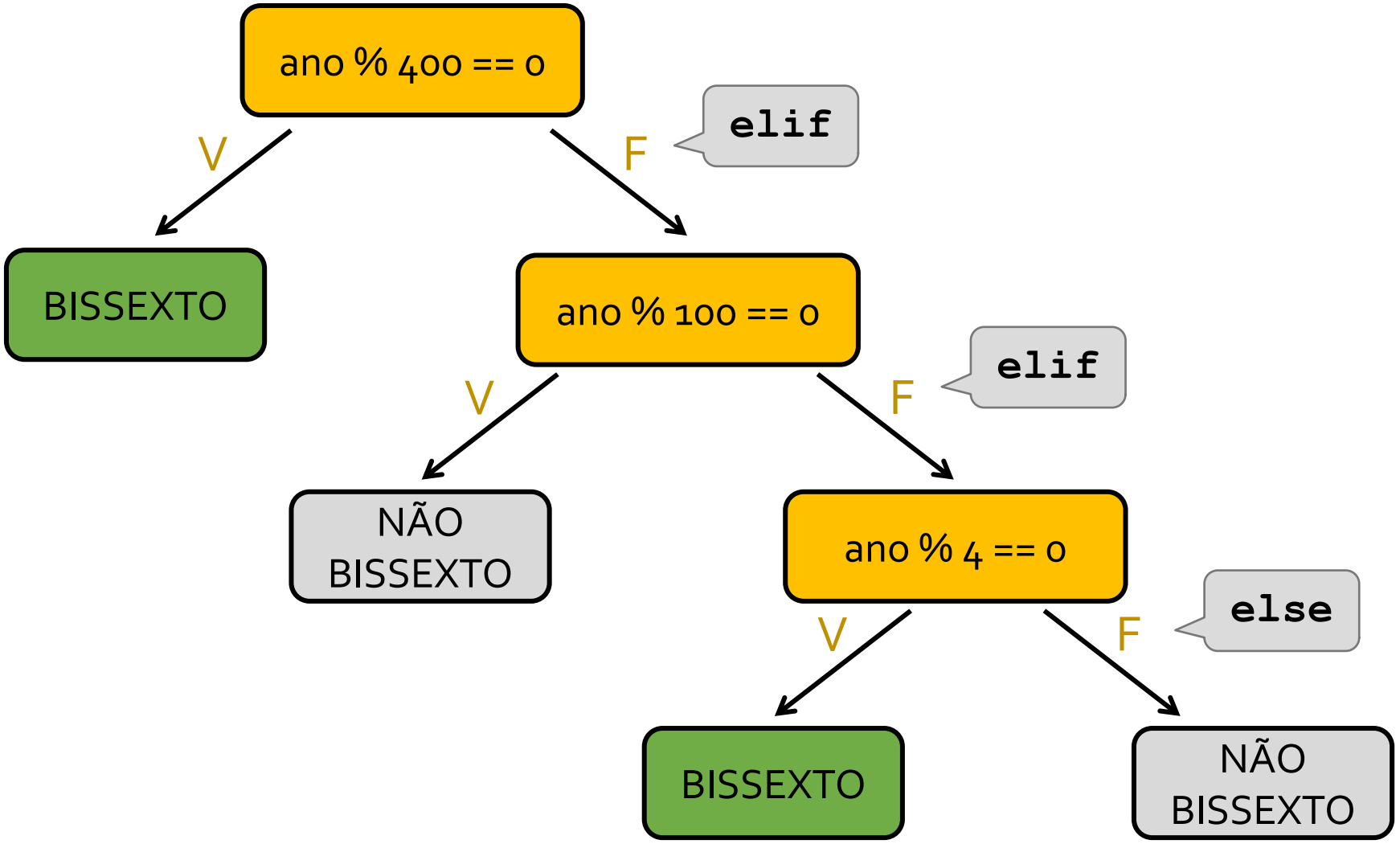
- Escreva um programa que determine se um ano é BISSEXTO ou NÃO BISSEXTO (sem acentos).

Problema 4
2 – Identifique as entradas e saídas

	Grandeza	Unidade de medida	Faixa de valores
Entradas	Ano	---	> 0
Saídas	Mensagem	---	{BISSEXTO, NÃO BISSEXTO}

Seria mais correto usar **> 1582**, quando o Calendário Gregoriano entrou em vigor

Problema 4
3 – Projete o script – versão 1



Problema 4
4 – Codifique em Python – versão 1

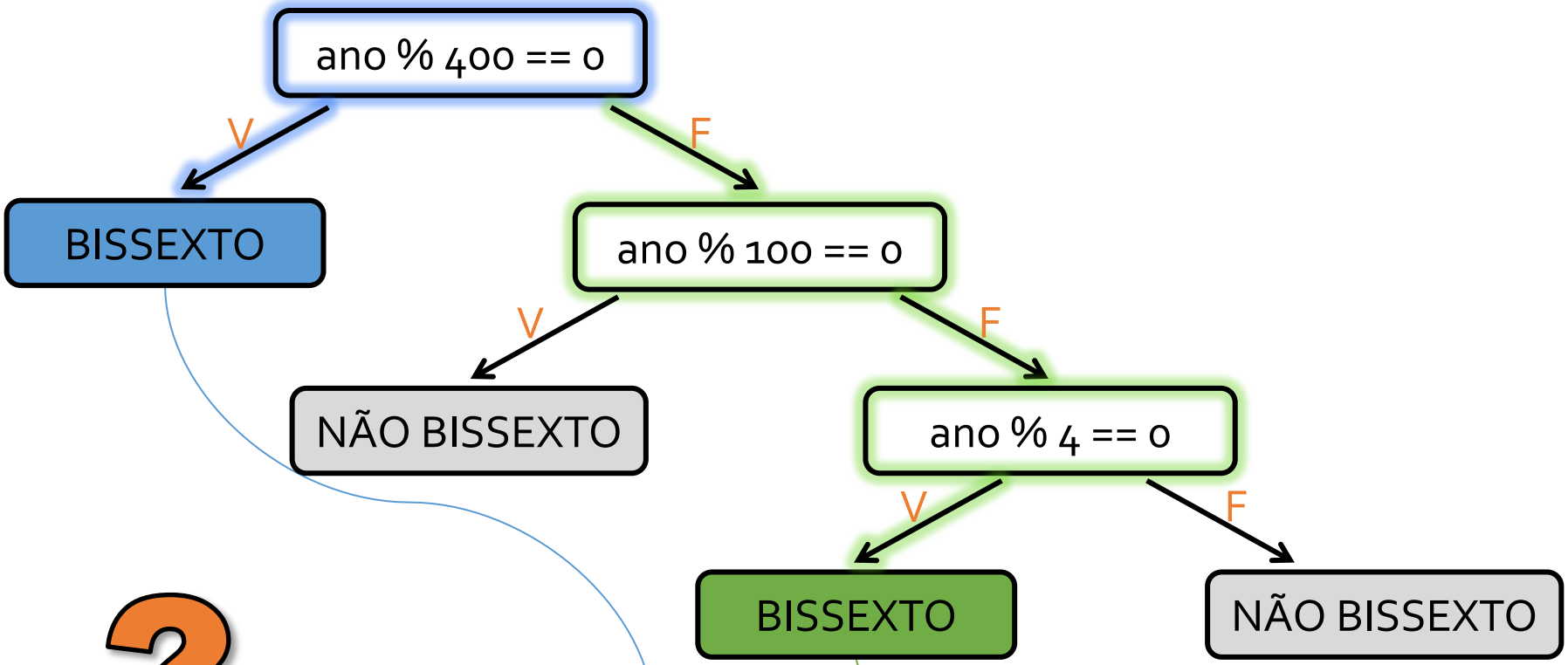
```
# Entrada de dados
ano = int(input("Ano: "))

if (ano <= 0):
  print("Entrada invalida")
elif (ano % 400 == 0):
  print("BISSEXTO")
elif (ano % 100 == 0):
  print("NAO BISSEXTO")
elif (ano % 4 == 0):
  print("BISSEXTO")
else:
  print("NAO BISSEXTO")
```

Ano é um valor **inteiro**

Se há apenas dois casos, como reduzir o código a apenas duas decisões?

Problema 4
3 – Projete o script – versão 2



Quais caminhos levam a um ano bissexto?

```
(ano % 400 == 0) or
((ano % 100 != 0) and (ano % 4 == 0))
```

Problema 4
4 – Codifique em Python – versão 2

```
# Entrada de dados
ano = int(input("Ano: "))

if (ano <= 0):
  print("Entrada invalida")
elif ((ano%400 == 0) or ((ano%100 != 0) and (ano%4 == 0))):
  print("Bissexto".upper())
else:
  print("Nao Bissexto".upper())
```

Ajusta todas as letras de uma string para **maiúsculas**

Problema 4
5 – Teste o script

-22	Entrada invalida
2018	Nao Bissexto
2019	Nao Bissexto
2020	Bissexto
2100	Nao Bissexto

Problema 5

- Escreva um programa que leia a renda mensal de uma pessoa.
- Como saída, informe o valor do imposto de renda devido.
- Importante: as alíquotas são aplicadas de forma progressiva.



Problema 5
2 – Identifique as entradas e saídas

	Grandeza	Unidade de medida	Faixa de valores
Entradas	Renda	R$	≥ 0
Saídas	Imposto	R$	≥ 0

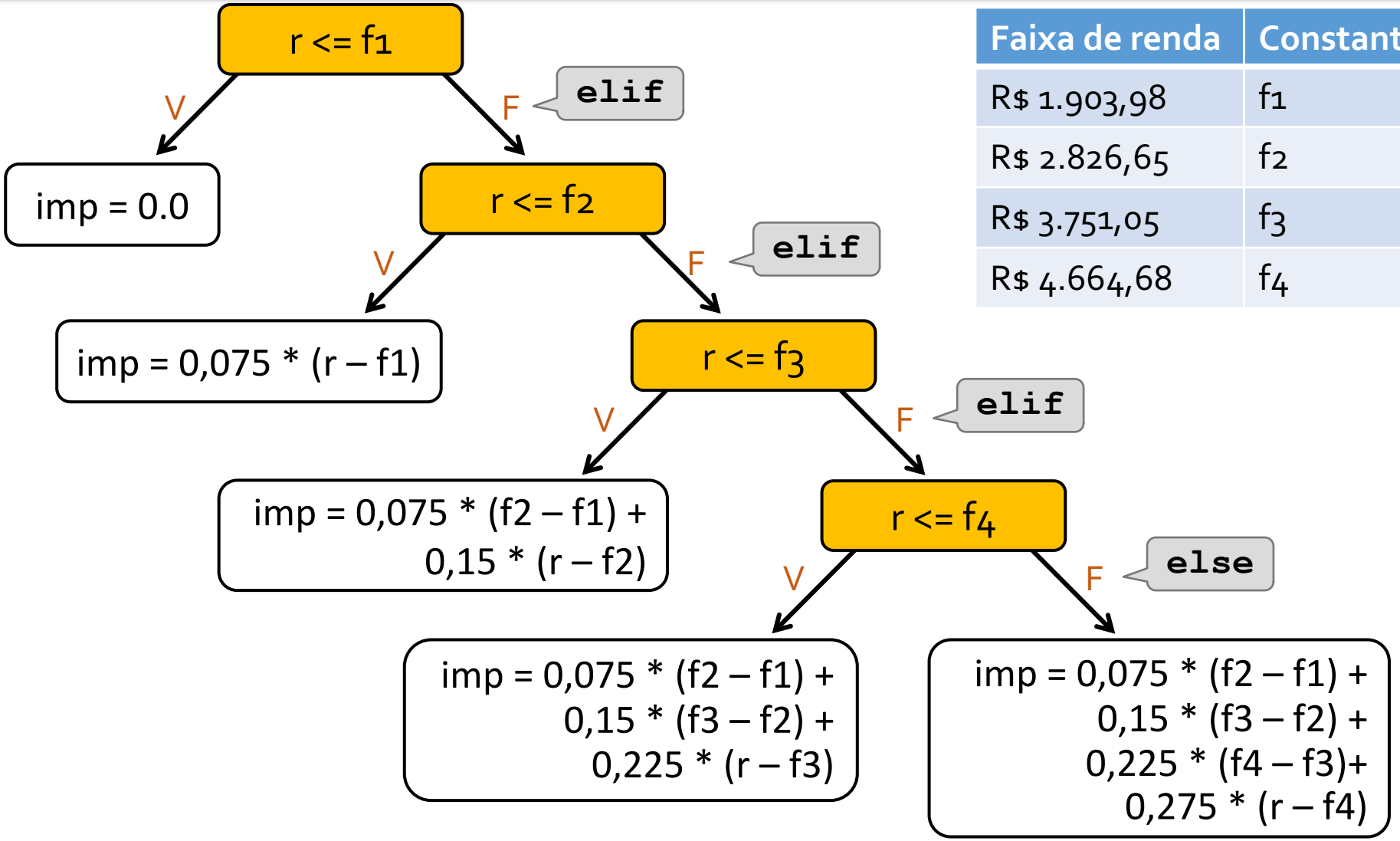
Problema 5
O que é cálculo progressivo?

- Uma pessoa que recebe R$ 2.000 mensais não pagará imposto de 7,5% sobre os R$ 2.000, mas sim sobre a diferença (2.000,00 – 1.903,98 = 96,02).
- Da mesma maneira, quem recebe R$ 3.000 pagará:
  - 7,5% de (2.826,65 – 1.903,99) + 15% de (3.000 – 2.826,66)
- E assim por diante...

Fonte: Receita Federal, 2016

Faixa de renda mensal	Alíquota
Até R$ 1.903,98	Isento
De R$ 1.903,99 até R$ 2.826,65	7,5%
De R$ 2.826,66 até R$ 3.751,05	15%
De R$ 3.751,06 até R$ 4.664,68	22,5%
acima de R$ 4.664,68	27,5%

Problema 5
3 – Projete o script



Faixa de renda	Constante
R$ 1.903,98	f1
R$ 2.826,65	f2
R$ 3.751,05	f3
R$ 4.664,68	f4

Problema 5
4 – Codifique em Python

```
r = float(input("Digite sua renda: "))
f1 = 1903.98
f2 = 2826.65
f3 = 3751.05
f4 = 4664.68

if (r <= f1):
    imp = 0.0
elif (r <= f2):
    imp = 0.075*(r-f1)
elif (r <= f3):
    imp = 0.075*(f2-f1) + 0.15*(r-f2)
elif (r <= f4):
    imp = 0.075*(f2-f1) + 0.15*(f3-f2) + 0.225*(r-f3)
else:
    imp = 0.075*(f2-f1)+0.15*(f3-f2)+0.225*(f4-f3)+0.275*(r-f4)
print(round(imp, 2))
```

Dinheiro é valor **real**

Arredondar os centavos

Problema 5
5 – Teste o script



R$ 1000 ≤ f_1 → 0.0

R$ 2000 $f_1 \leq f_2$ → 7.2

R$ 3000 $f_2 \leq f_3$ → 95.2

R$ 4000 $f_3 \leq f_4$ → 263.87

R$ 5000 ≥ f_4 → 505.64



Como saber se as contas estão corretas?

Problema 5
5 – Teste o script

```
r = float(input("Digite sua renda: "))
f1 = 1903.98
f2 = 2826.65
f3 = 3751.05
f4 = 4664.68

if (r <= f1):
    imp = 0.0
    print("Faixa f1")
elif (r <= f2):
    imp = 0.075*(r-f1)
    print("Faixa f2")
elif (r <= f3):
    imp = 0.075*(f2-f1) + 0.15*(r-f2)
    print("Faixa f3")
elif (r <= f4):
    imp = 0.075*(f2-f1) + 0.15*(f3-f2) + 0.225*(r-f3)
    print("Faixa f4")
else:
    imp = 0.075*(f2-f1)+0.15*(f3-f2)+0.225*(f4-f3)+0.275*(r-f4)
    print("Maior que faixa f4")
print(imp)
```

Use mensagens indicativas para testar. Depois, comente usando #.

Conteúdo

- Como montar uma condição?
- Estruturas Condicionais Encadeadas
- **Exemplos Extras**

Exemplo 1

- Sejam A, B, C três números inteiros quaisquer.
- Escreva um script para arrumá-los em ordem decrescente.

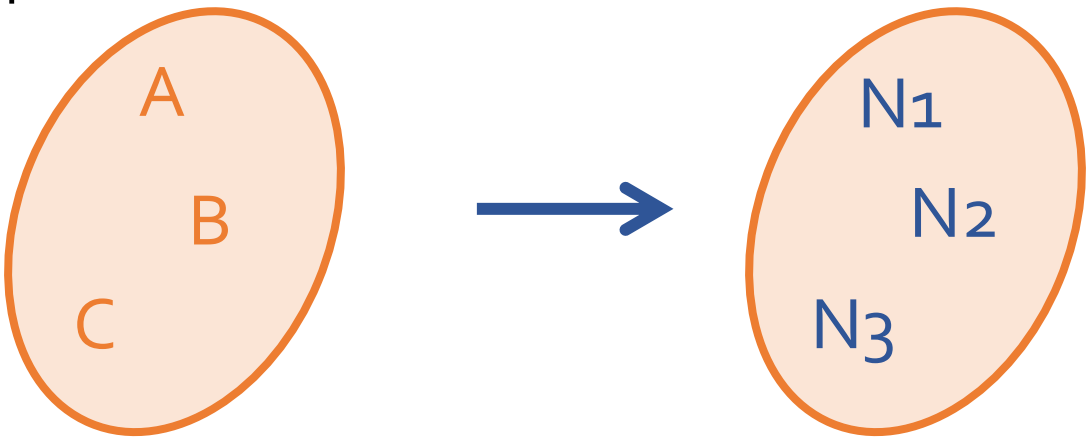


Exemplo 1
1 – Entenda o problema

- São dados três números inteiros quaisquer: A, B, C.
- Eles devem ser arrumados em ordem decrescente.
- Pode-se considerar que a saída seja

$$N_1 \geq N_2 \geq N_3$$

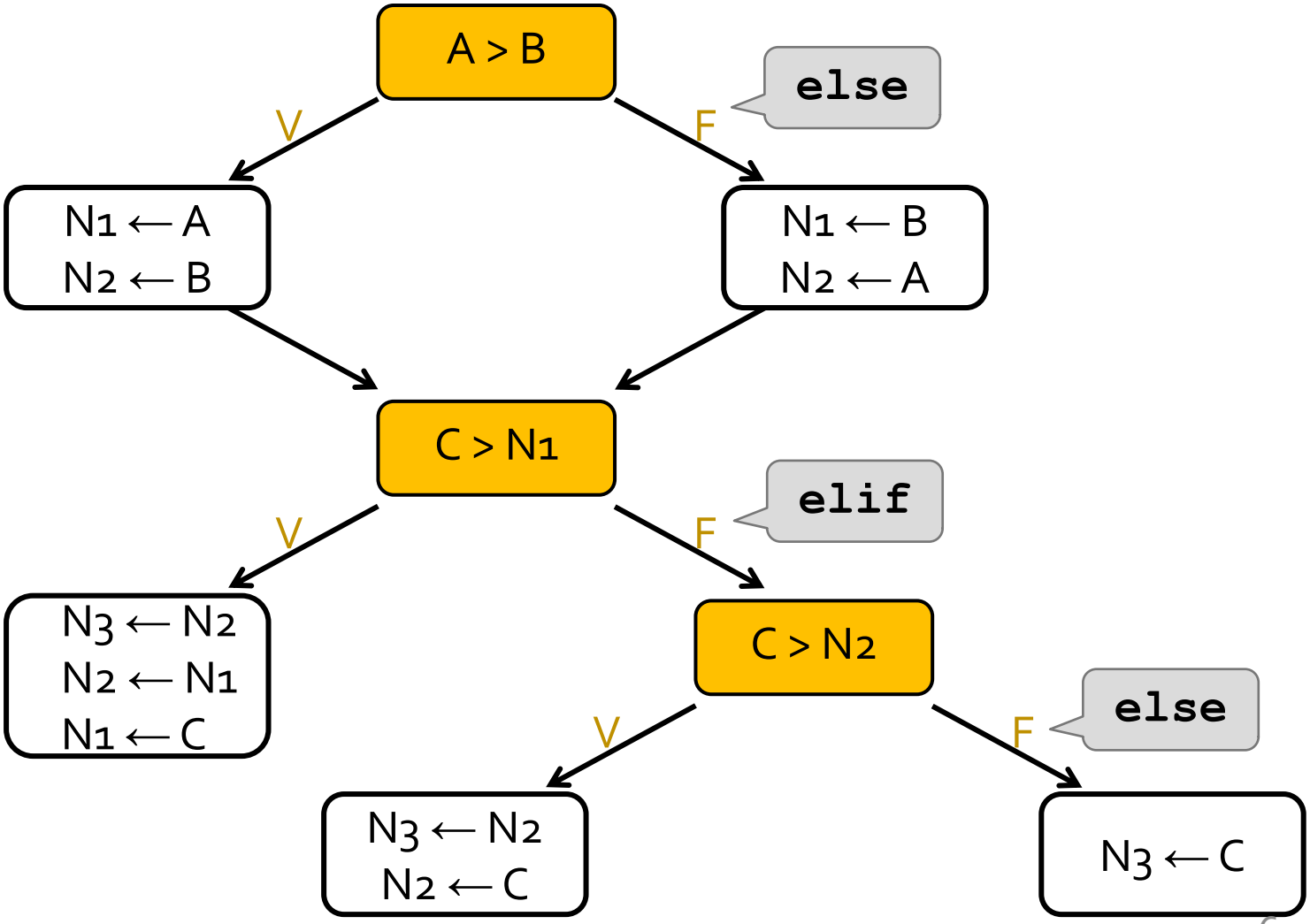
- Agora, o problema se resume a associar:



Exemplo 1
2 – Identifique as entradas e saídas

	Grandeza	Unidade de medida	Faixa de valores
Entradas	A, B, C	---	Inteiros, qualquer ordem
Saídas	N1, N2, N3	---	Inteiros, ordem decrescente

Exemplo 1
3 – Projete o script



Exemplo 1
4 – Codifique em Python

```
a = int(input("Numero a: "))
b = int(input("Numero b: "))
c = int(input("Numero c: "))

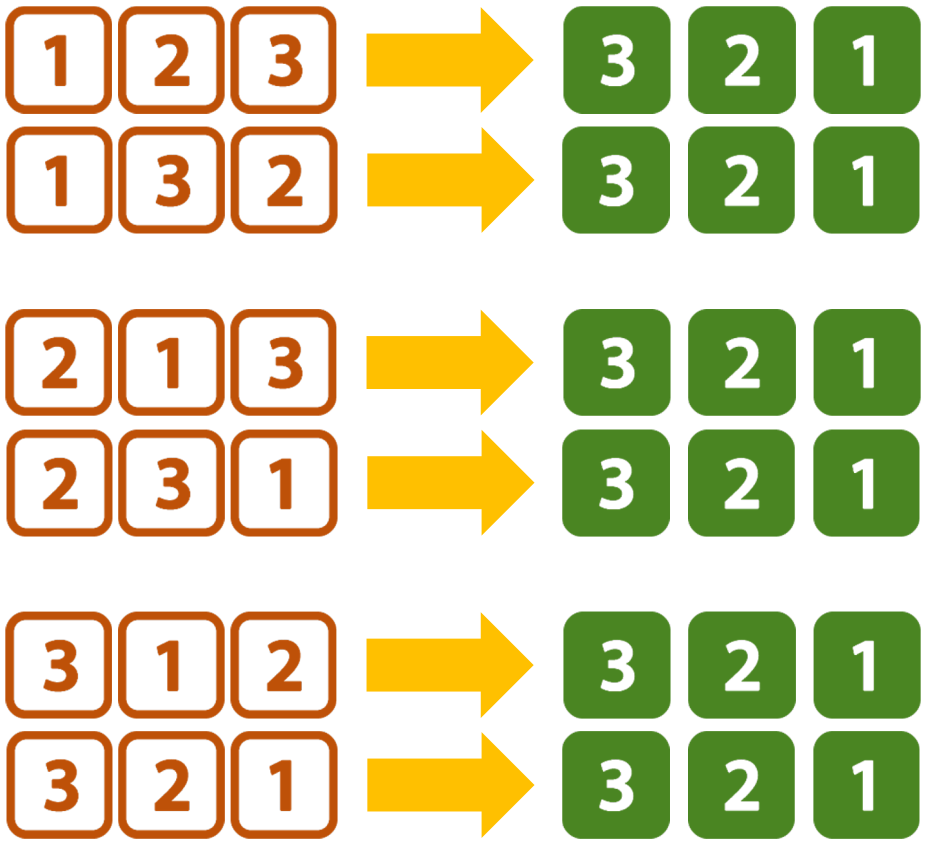
if (a > b):
    n1 = a
    n2 = b
else:
    n1 = b
    n2 = a

if (c > n1):
    n3 = n2
    n2 = n1
    n1 = c
elif (c > n2):
    n3 = n2
    n2 = c
else:
    n3 = c

print(n1, n2, n3)
```

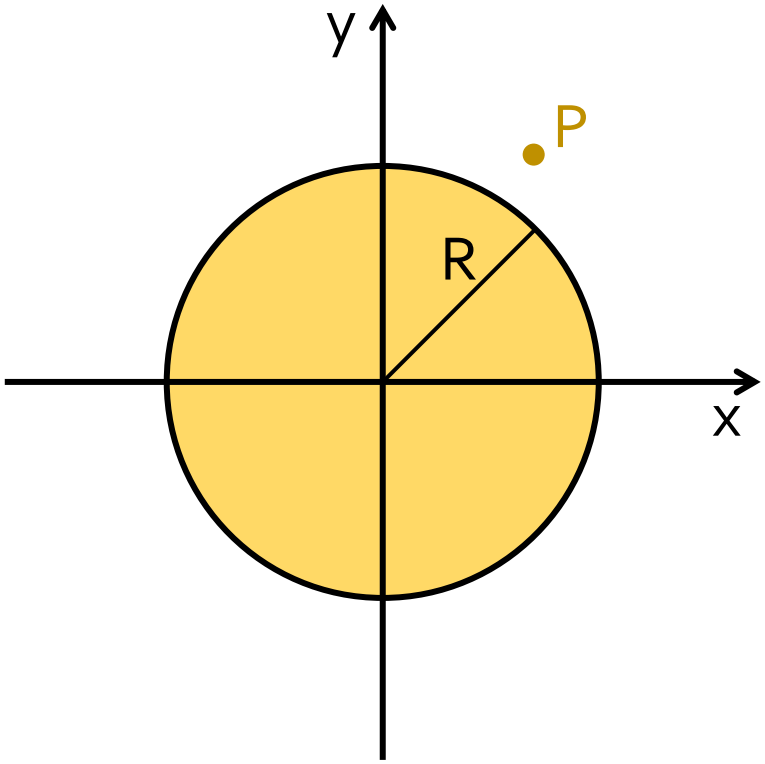
Exemplo 1
5 – Teste o script

- As 3 entradas podem ser permutadas de 3! = 6 maneiras:



Exemplo 2

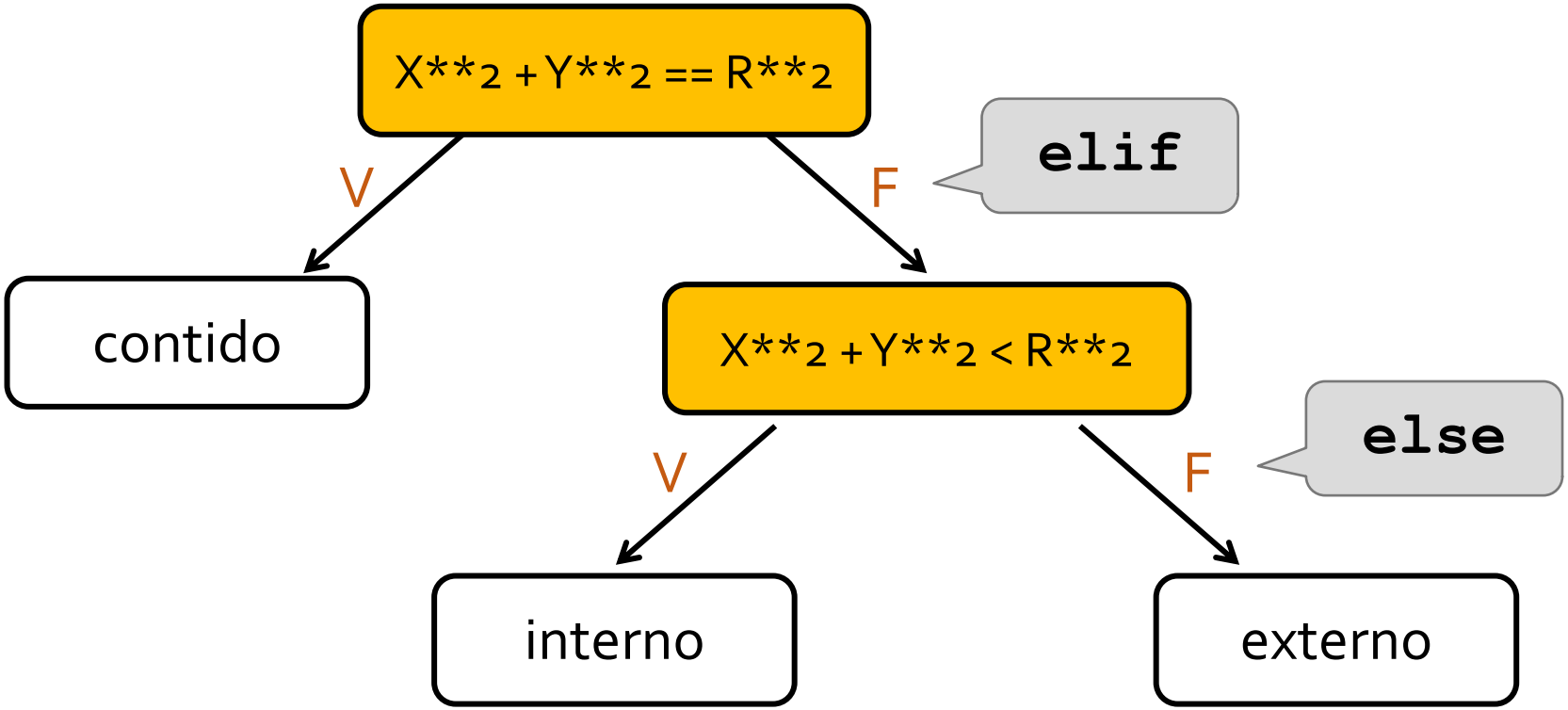
- A equação de uma circunferência de raio R é $x^2 + y^2 = R^2$.
- Escreva um algoritmo que, dado um ponto P qualquer, verifique se ele se encontra:
  - Contido na circunferência
  - No interior
  - No exterior



Exemplo 2
2 – Identifique as entradas e saídas

	Grandeza	Unidade de medida	Faixa de valores
Entradas	Raio	---	>= 0
	Coordenada X de P	---	]-∞; +∞[
	Coordenada Y de P	---	]-∞; +∞[
Saídas	Localização de P	---	{interior, exterior, contido}

Exemplo 2
3 – Projete o script



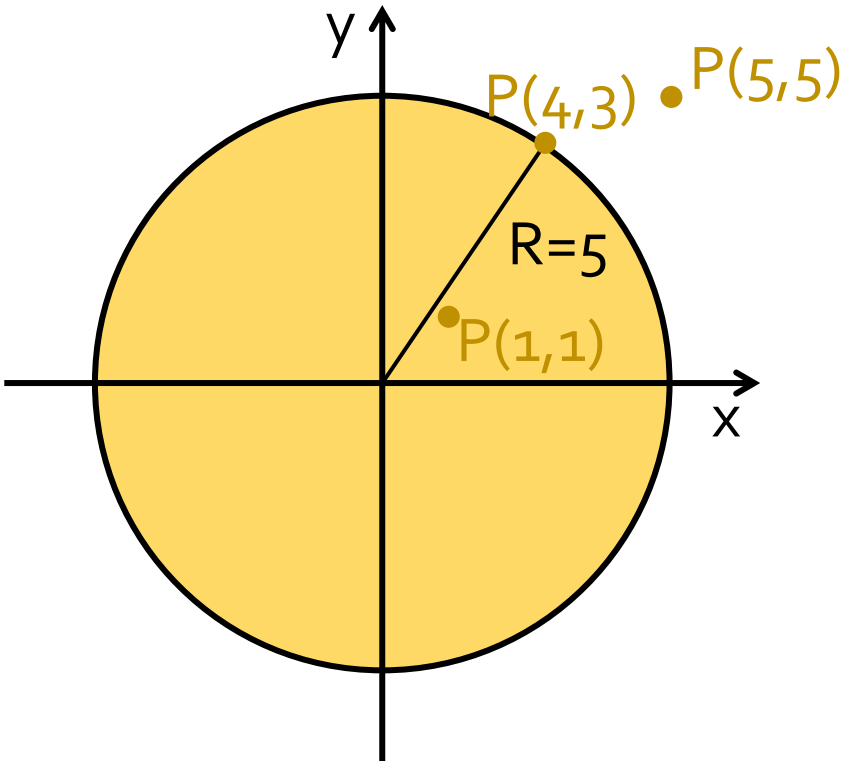
Exemplo 2
4 – Codifique em Python

Coordenadas são valores **reais**

```
# Entrada de dados
r = float(input("Raio: "))
x = float(input("Coord. X do ponto: "))
y = float(input("Coord. Y do ponto: "))

if (x**2 + y**2 == r**2):
    print("contido")
elif (x**2 + y**2 < r**2):
    print("interno")
else:
    print("externo")
```

Exemplo 2
5 – Teste o script



R = 5
X = 5
Y = 5
→ Externo

R = 5
X = 4
Y = 3
→ Contido

R = 5
X = 1
Y = 1
→ Interno

Exemplo 3

- Uma quantia inicial q é aplicada a uma taxa de juros t. O saldo s desse investimento após m meses é dado por:

$$s = q(1 + t)^m$$

- Escreva um programa que leia uma taxa t de juros ao mês.
- Determine quanto tempo (em anos e meses) é necessário para que o saldo dobre em relação ao valor inicial.



Exemplo 3
2 – Identifique as entradas e saídas

	Grandeza	Unidade de medida	Faixa de valores
Entradas	Taxa de juros (t)	---	[0; 1]
Saídas	m (tempo)	anos	[0, +∞[
	m (tempo)	meses	[0, 11]

Exemplo 3
3 – Projete o script

- A saída do problema é a quantidade de meses e anos.
- Mas a saída da equação é o saldo, informação que já conhecemos.
- Portanto, temos de reescrever a equação, isolando o m no lado esquerdo.

$$s = q(1+t)^m$$

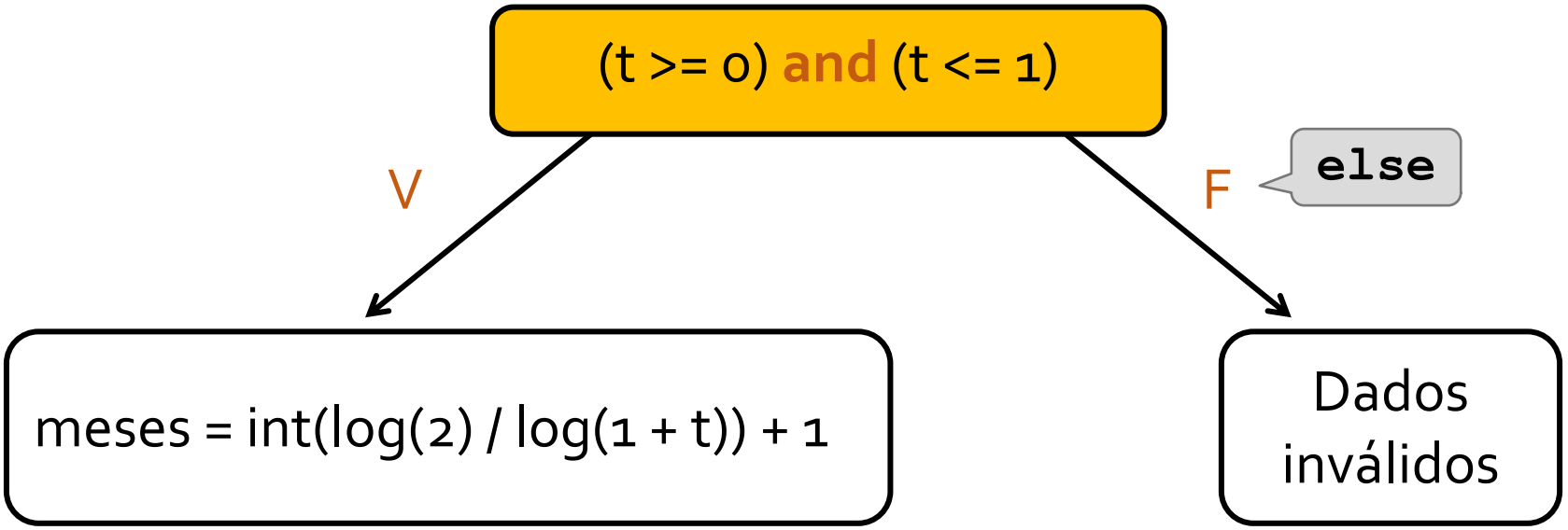
$$\frac{2q}{q} = (1+t)^m$$

$$\log 2 = \log(1+t)^m$$

$$\log 2 = m \cdot \log(1+t)$$

$$\boxed{m = \frac{\log 2}{\log(1+t)}}$$

Exemplo 3
3 – Projete o script



Exemplo 3
4 – Codifique em Python

```
# Entrada de dados
t = float(input("Informe a taxa de aplicacao: "))

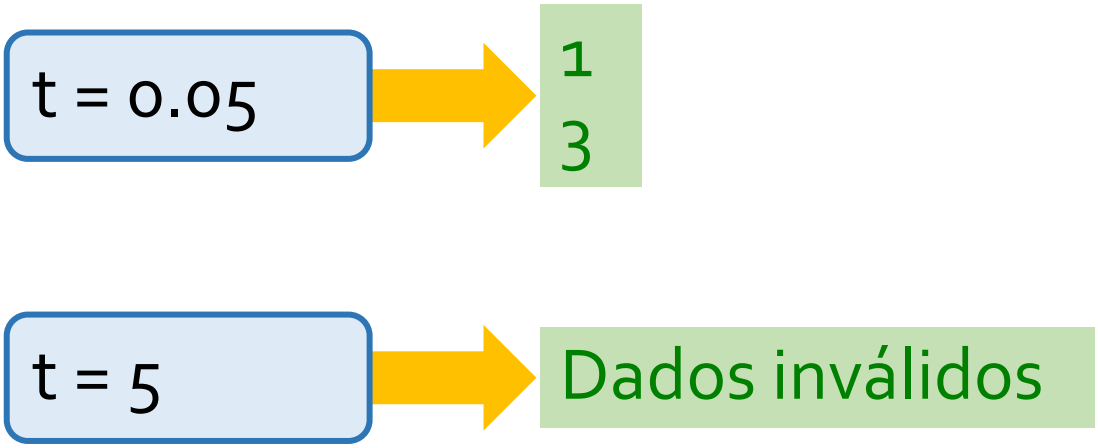
if ((t >= 0) and (t <= 1)):
    # Importar modulo matematico
    from math import *

    m = int(log(2) / log(1 + t)) + 1

    print(m // 12)    # no. de anos
    print(m % 12)     # no. de meses (0 a 11)
else:
    print("Dados invalidos")
```

Validação da entrada

Exemplo 3
5 – Teste o script



Exemplo 4

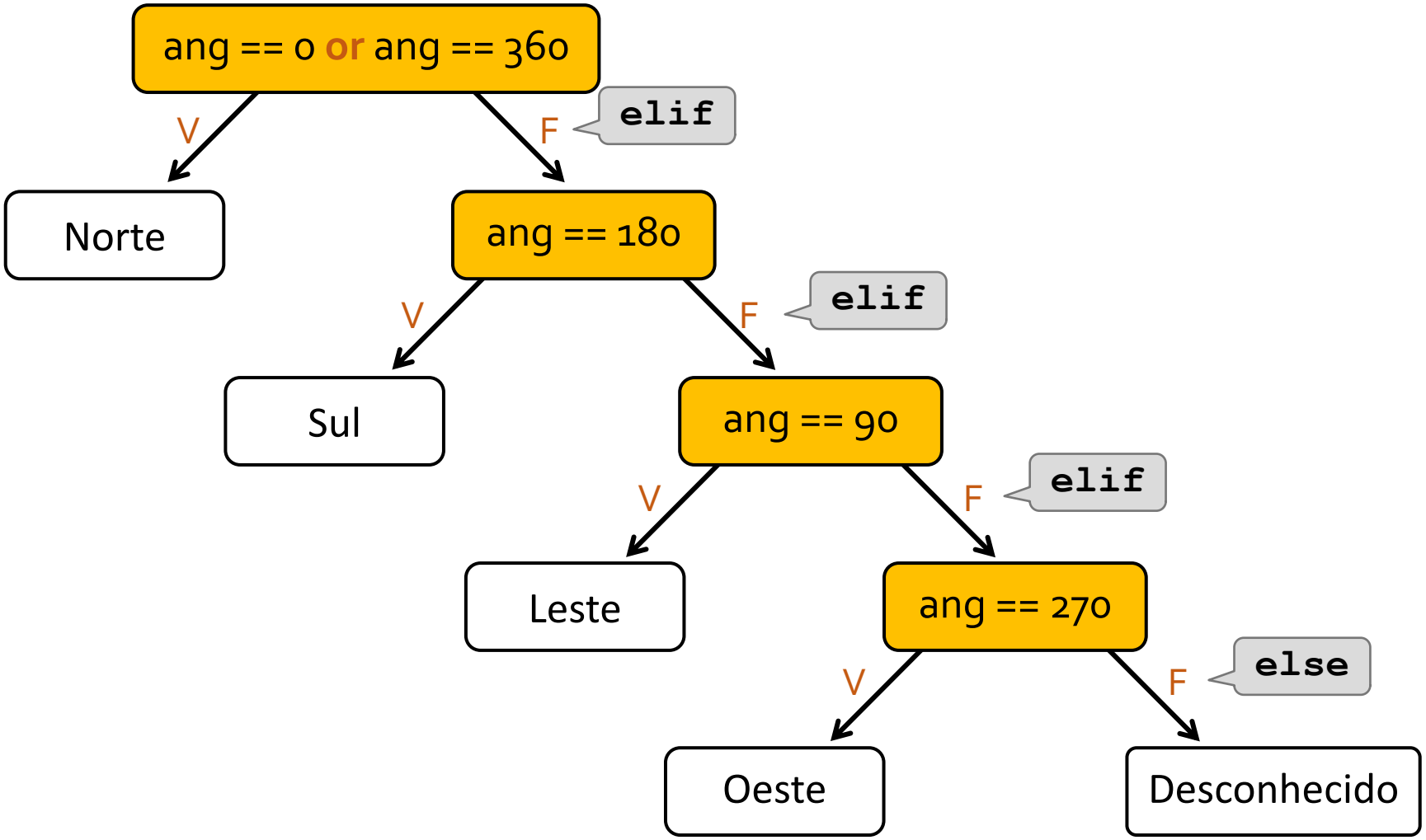
- Escrever um script em que leia um ângulo entre zero e 360° e informe o ponto cardeal correspondente.



Exemplo 4
2 – Identifique as entradas e saídas

	Grandeza	Unidade de medida	Faixa de valores
Entradas	Ângulo	graus	[0; 360]
Saídas	Ponto Cardeal	---	{N, S, L, O}

Exemplo 4
3 – Projete o script



Exemplo 4
4 – Codifique em Python

```
# Entrada de dados
ang = int(input("Angulo: "))

if ((ang == 0) or (ang == 360)):
    print("Norte")
elif (ang == 180):
    print("Sul")
elif (ang == 90):
    print("Leste")
elif (ang == 270):
    print("Oeste")
else:
    print("Desconhecido")
```



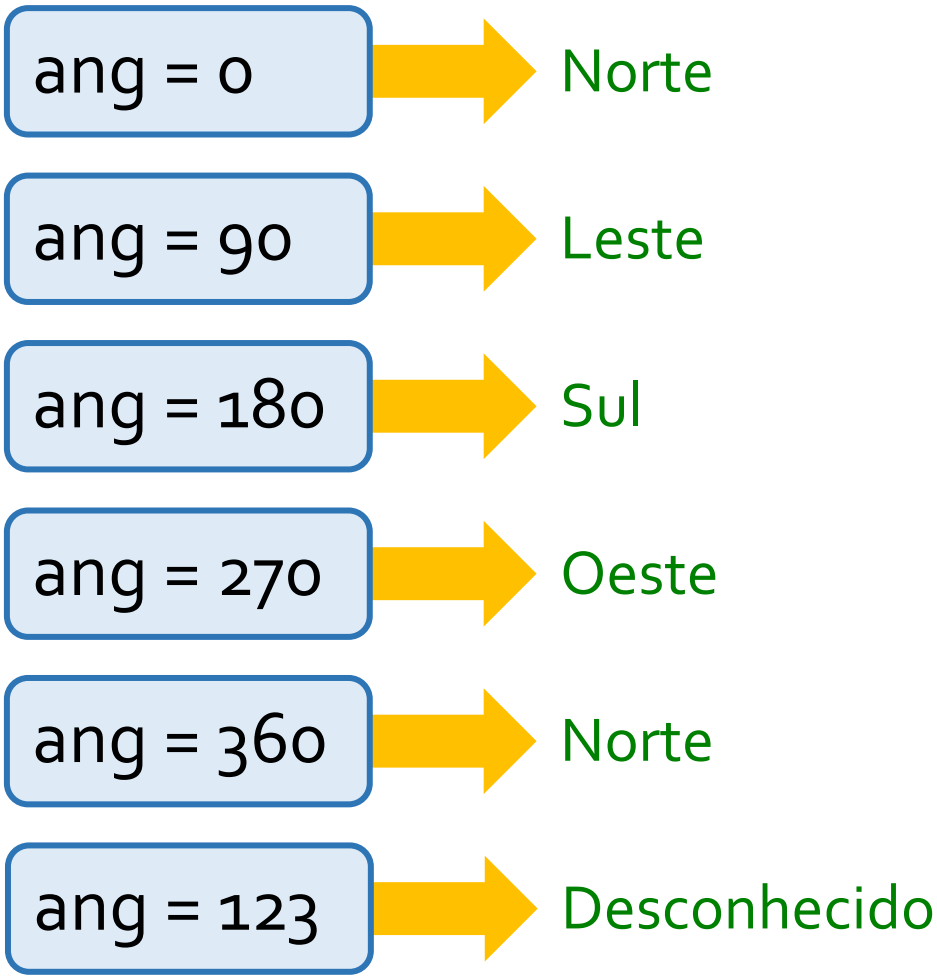
Não faça essa loucura!

```
ang == 0 or 360
```



Atenção para iniciais **maiúsculas**

Exemplo 4
5 – Teste o script



Referências bibliográficas



MENEZES, Nilo Ney Coutinho (2014). Introdução à Programação com Python, 2ed. Editora Novatec.



HETLAND, Magnus Lie (2008). Beginning Python: From Novice to Professional. Springer eBooks, 2ª edição. Disponível em: http://dx.doi.org/10.1007/978-1-4302-0634-7.

- GADDIS, Tony (2012). Starting out with Python, 2ª edição. Editora Addison-Wesley.
- DIERBACH, Charles. Introduction to Computer Science using Python: a computational problem-solving approach. John Wiley & Sons, 2012.

Dúvidas?

