

SSC0304 - Introdução à Programação para Engenharias

Operadores Condicionais

Prof.: Leonardo Tórtoro Pereira

leonardop@usp.br

Baseado no material dos profs Fernando S. Osório e Claudio F.M. Toledo

Na aula passada...



Operadores Relacionais

Operador	Descrição	Exemplo	Resultado
>	Maior que	$2 > 3$	False
<	Menor que	$2 < 4$	True
>=	Maior igual que	$2 >= 2$	True
<=	Menor igual que	$4 <= 4$	True
!=	Diferente de	$\text{True} != \text{False}$	True
==	Igual a	$\text{True} == 1$	True

Operadores Lógicos

Operador	Descrição	Exemplo	Resultado
not	Negação lógica	<code>not 2 == 2</code>	False
and	E lógico	<code>2 == 2 and 4==2</code>	False
or	OU lógico	<code>2 == 2 or 4==2</code>	True

Operadores Aritméticos

Operador	Descrição	Exemplo	Resultado
+	Adição	$2 + 3$	5
-	Subtração	$2 - 2$	0
*	Multiplicação	$2 * 3$	6
/	Divisão	$5/4$	1.25
//	Divisão inteira	$4//3$	1
**	Exponenciação	$5**3$	125
%	Resto de Divisão (mod)	$10\%2$	0

Operadores bit a bit

Operador	Descrição	Exemplo	Resultado
~	Complemento bit-a-bit	~0010	1101
<<	Deslocar bits à esq.	001 << 1	010
>>	Deslocar bits à dir.	100 >> 2	001
&	E (AND) bit-a-bit	001 & 101	001
^	OU exclusivo (XOR) bit-a-bit	001 ^ 101	100
	OU (OR) bit-a-bit	001 101	101

Precedência de operadores

Operador	Precedência
()	+
**	
-	
* / % //	
+ -	
> >= < <= == !=	
not	
and	
or	
=	-

Expressões

→ Expressões são compostas por:

◆ Operandos/Variáveis: `a`, `b`, `x`, `Meu_dado`, `2`, ...

◆ Operadores: `+`, `-`, `%`, ...

◆ Precedência: `()`

◆ Funções da biblioteca: `math.sin(x)`, `math.sqrt(x)`, ...

→ Sempre retornam um valor

◆ `X = 5 + 4` -> Retorna 9

◆ `((5 + 4) == 9)` -> Retorna True

Entrada e Saída de Dados

- *input()* sempre lê dados como str (string)
 - ◆ `a = input("Valor de a")`
- Para converter em outros tipos, usar conversão explícita
 - ◆ `c = int(a) + int(b)`
 - ◆ `a = float(input("Valor de a"))`

Entrada e Saída de Dados

→ Pode-se exibir os resultados com *print()*

◆ `a = "batata"`

◆ `print(a)`

◆ `b = 132`

◆ `print(b)`

◆ `c = 123.433632`

◆ `print("{:.3f}".format(c))`

◆ `print(format(c, ".2f"))`

O que vamos aprender hoje?



Objetivos

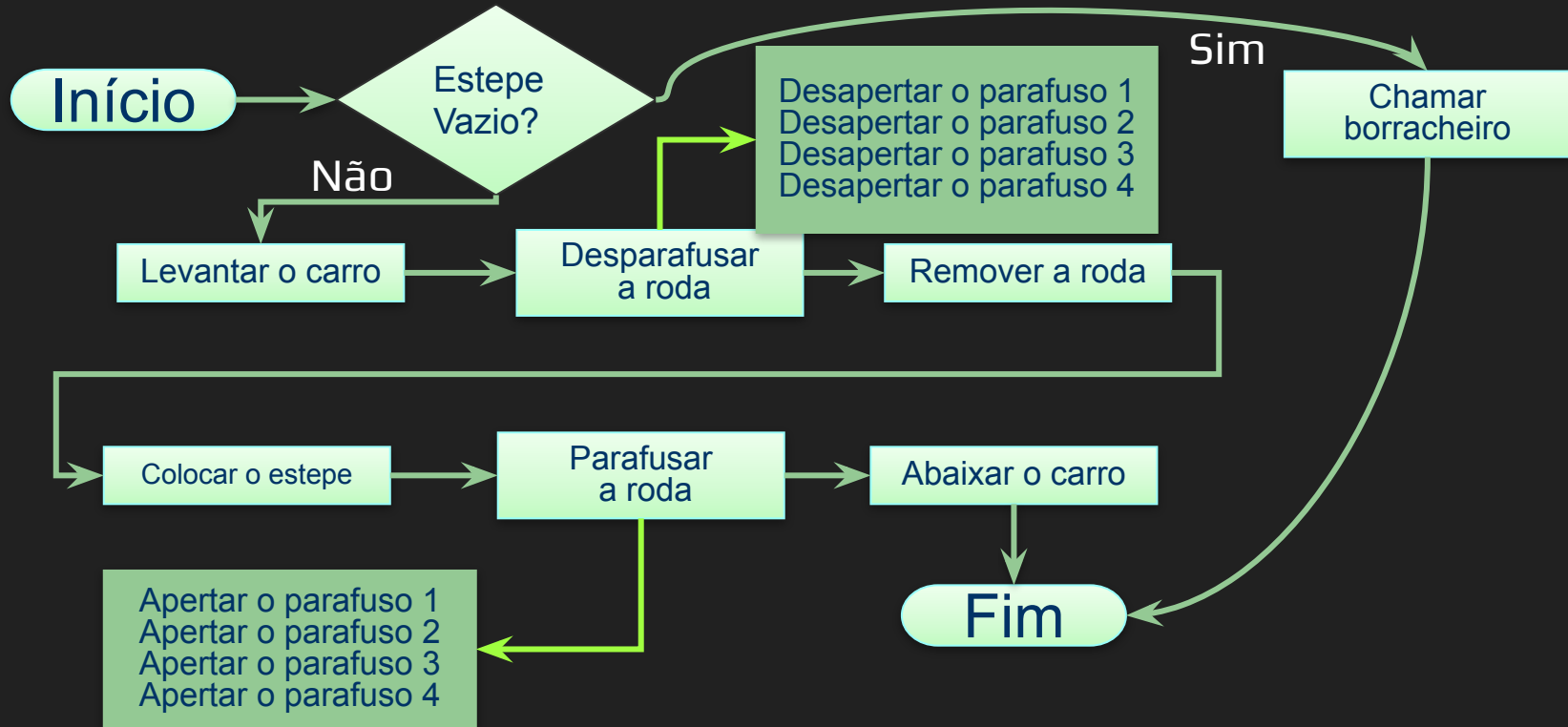
→ Aprender a realizar fluxos condicionais em Python

Tópicos da Aula

- Estrutura Condicionais
- Operadores If-Elif-Else
- Aninhamento de If-Elif-Else

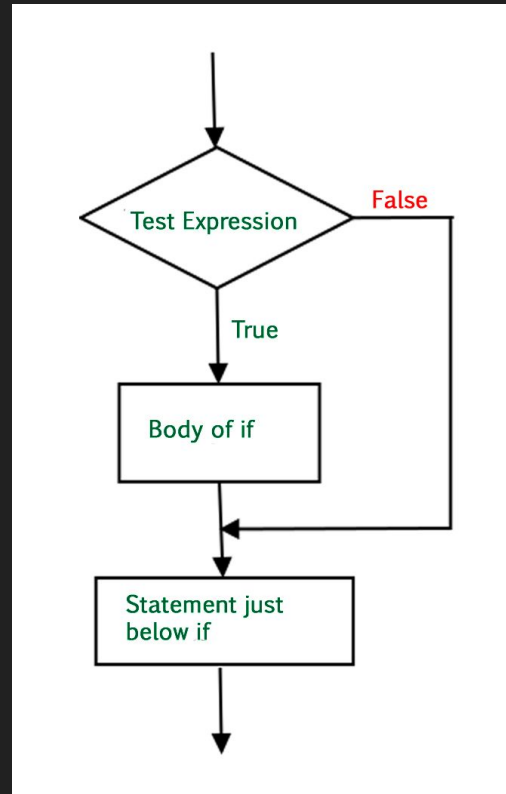
Estrutura Condicionais

Algoritmo para trocar um pneu - Relembrando



Estrutura Condicionais


- Usada para definir qual “caminho” o código deve seguir
- Muda o fluxo de execução do programa de acordo com as condições impostas
- Deve-se tomar cuidado para garantir que todas as possibilidades sejam levadas em conta
- Pode ser dada pela estrutura *if-else-else if*, por *switch-case* ou *operador ternário*



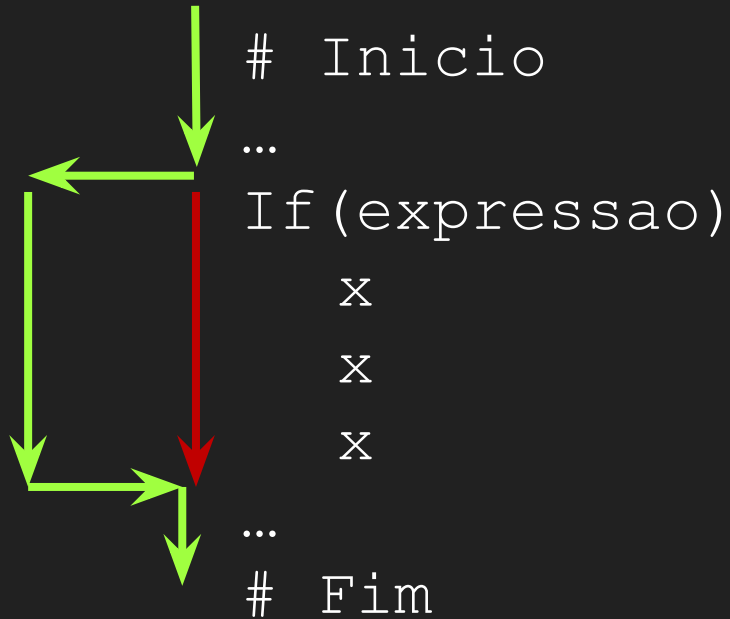
Fluxograma de uma estrutura condicional. Fonte [5]

Fluxo Sequencial Linear

```
# Inicio  
A=1  
B=2  
C=3  
D=4  
E=5  
F= A + B +  
C + D + E  
print (F)  
# Fim
```



Desvios Condicionais



Desvios Condicionais

```
# Inicio  
B=C=10  
A = int(input())  
if (A < 5) :  
    A=1  
    B=1  
    C=1  
F= ( A + B + C) / 5.0  
print (F)  
# Fim
```

Cuidado com indentação

- Em Python, não usamos chaves para separar “blocos”, como outras linguagens
- Tudo é orientado à indentação
 - ◆ Tudo que está dentro do if, tem que estar com uma “tab” a mais que o “pai” da hierarquia (comando “if”)
- Isso se repete para outros comandos em “bloco”

Operador if-else

If-Else

- A estrutura mínima é com o bloco de código
 - ◆ *if(condição): {código a ser executado}*
- Na qual a condição pode ser qualquer expressão que retorne um booleano
- Se a condição for verdadeira (valor diferente de zero) o bloco de código abaixo do *if* será executado
- Caso seja falsa (valor zero), o bloco de código é ignorado
- “{}” indica código indentado, no caso de Python

If-Else

→ Condições usam os operadores condicionais

◆ Igual a "=="

◆ Não igual a "!="

◆ Maior que ">" ou Maior ou Igual a ">="

◆ Menor que "<" ou Menor ou Igual a "<="

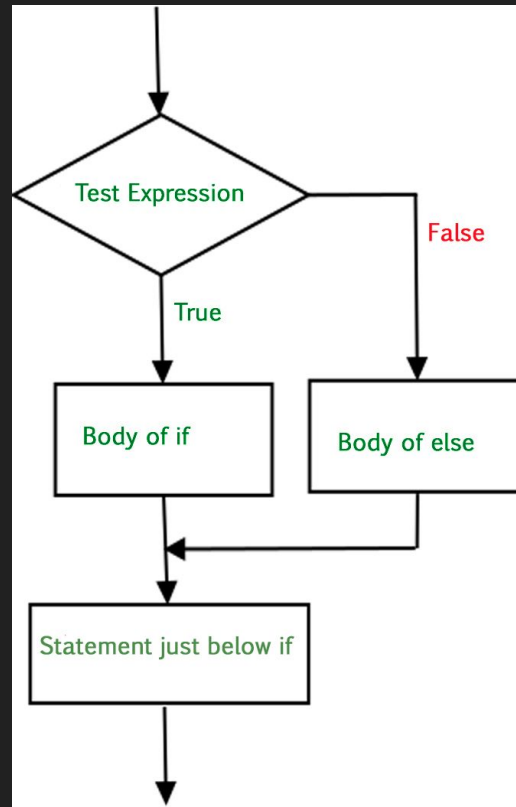
→ Os operadores lógicos também podem ser usados para conectar expressões

◆ and, or e not "!"

Exemplos

If-Else

- Você pode executar um outro bloco de código caso a condição retorne falso com o comando *else {código}*
- Só pode ser chamados DEPOIS de um *if()*

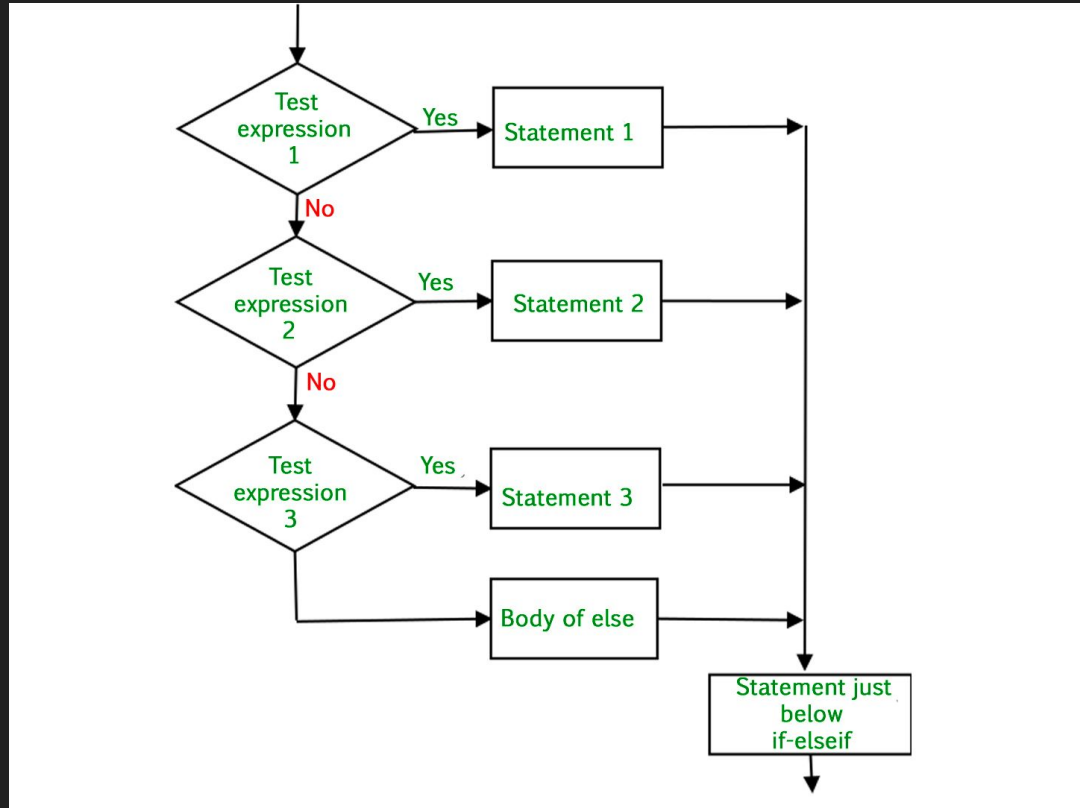


Fluxograma do If-Else. Fonte [5]

Exemplos

If-Elif-Else

- Você também pode colocar mais de uma condição com o comando *elif(condição) {código}*
- Você pode colocar quantos *elif* quiser depois de um *if()*
- Só podem ser chamados DEPOIS de um *if()*



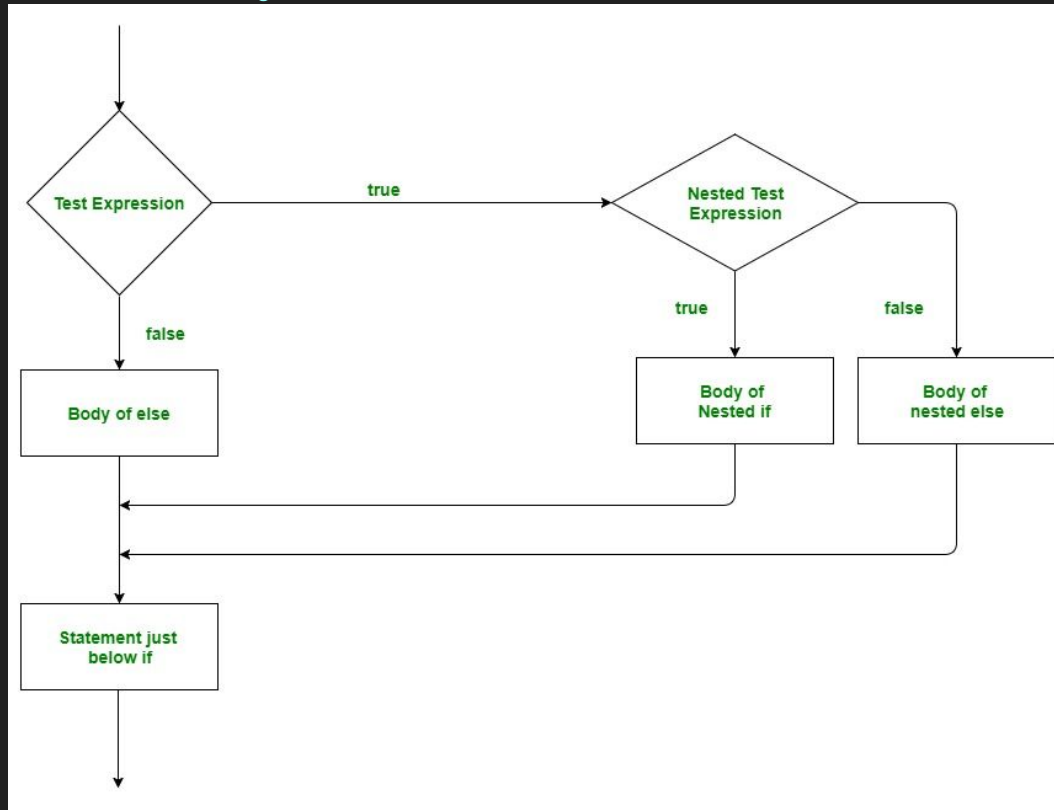
Fluxograma do If-Elif-Else. Fonte [5]

Exemplos

Aninhamento de If-Elif-Else

Aninhamento de If-Elif-Else

- É possível aninhar declarações condicionais caso uma sequência de condições precise ser satisfeita
- É uma alternativa à conectar expressões com operadores lógicos



Fluxograma de aninhamentos. Fonte [5]

Exemplos

```

function register()
{
    if (empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] === $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}

```



Evite o if Hadouken!

Evite o Hadouken!

- Normalmente, se você está com muitos *ifs* aninhados, provavelmente vai ser muito mais fácil inverter a condição e tratar os casos negativos (*elses*)
- Outras vezes funções podem te ajudar. Estudaremos sobre elas mais adiante no curso :)

If-Else "resumido"

```
i = 10
```

```
print(True) if i < 15 else print(False)
```

If-Else "resumido" [5]

```
def digitSum(n):  
    dsum = 0  
    for ele in str(n):  
        dsum += int(ele)  
    return dsum  
List = [367, 111, 562, 945, 6726, 873]  
newList = [digitSum(i) for i in List if i & 1]  
print(newList)
```



Comparação de Ponto Flutuante

Comparação de ponto flutuante

- Devido aos erros de arredondamento de ponto flutuante, não é recomendado usar uma comparação de "==" para pontos flutuantes.
- Uma solução é verificar se a diferença entre eles é menor do que um valor arbitrariamente pequeno.

Exemplos

Referências

Referências

1. <https://www.learnpython.org/>
2. <https://www.w3schools.com/python/>
3. <https://panda.ime.usp.br/cc110/static/cc110/index.html>
4. https://www.youtube.com/playlist?list=PLcoJJSvnDgcKpOi_UeneTNTIV0igRQwcn
5. <https://www.geeksforgeeks.org/python-if-else/>