

Introdução a Lógica de programação - Python

Filipe Herculano

Sênior Blockchain e Backend Software Engineer na VHS
Bacharel em Ciência da Computação



Convento de Programadores

1. Setup
2. Console e Sintaxe Básica
3. Operações Básicas
4. Controle de Fluxo
5. Recebendo dados
6. Desafio - Calculadora

Aula 1

Setup

- Git - clonando um repositório
- Python
- Terminal



Aula 2

O console e Sintaxe Básica

- O que é o console (ou terminal)?
 - Modos de uso e "Hello World!"
 - Sintaxe básica
 - Identificadores
 - Palavras Reservadas
 - Indentação
 - Comentários
-

O que é o console (ou terminal)?

1. Uma interface gráfica
2. Interage com componentes físicos da sua máquina
3. Utilizado como dispositivo de Input e Output
4. Útil para programas simples.
Conseguimos rodar programas diretamente nele, sem muito setup



Modos de uso

1. Modo Interativo - REPL
2. Modo Script
3. Help



The screenshot shows the Warp terminal window on a Mac. The title bar indicates the user is at the home directory of a MacBook Pro. The terminal content is as follows:

```
~ (2m 23.77s)
python3
Python 3.10.0 (default, Aug 31 2023, 16:34:21) [Clang 13.1.6 (clang-1316.0.21.2.5)] on dar
win
Type "help", "copyright", "credits" or "license" for more information.
>>>
KeyboardInterrupt
>>>
KeyboardInterrupt
>>>

~ (0.148s)
python3 dev/convento/hello_world.py
Hello World!

~
Type '#' for AI command suggestions
```

Sintaxe Básica - Identificadores

1. Exemplo da vida real: Um nome próprio, ou endereço de uma casa
2. Usado para identificar valores, funções, classes, módulos entre outros.
3. Possuem algumas limitações
4. Alguns caracteres não são permitidos
5. Tipagem dinâmica!

Sintaxe Básica - Palavras Reservadas

1. São nada mais que identificadores por si só
2. São identificadores que o interpretador da linguagem usa
3. "invalid syntax" será o erro recebido caso um destes seja utilizado

and	exec	not
as	finally	or
assert	for	pass
break	from	print
class	global	raise
continue	if	return
def	import	try
del	in	while
elif	is	with
else	lambda	yield
except		

Sintaxe Básica - Indentação

1. Contexto de código
2. Python não utiliza {}
3. Indentação é a maneira de identificar contexto em Python

Sintaxe Básica - Comentários

1. Informações adicionais
2. Single Line
3. Multi Line

Desafio 1 - O indeciso

1. Crie um arquivo com extensão .py
2. Escreva seu nome em uma variável
3. Escreva sua idade
4. Escreva 2 atividades que goste de fazer
5. Faça seu código escolher por você

DICA1: Use if-else

DICA2:

```
from random import randrange
randrange(2)
```

Aula 3

Operações Básicas

- Atribuição
- Aritmética
- Lógico
- Relacional



Operações Aritméticas

1. Adição $+$
2. Subtração $-$
3. Multiplicação $*$
4. Divisão $/$
5. Módulo $\%$
6. Exponente $**$
7. Floor Division $//$

Operações de Atribuição

1. $=$

2. $+=$

3. $-=$

4. $*=$

5. $/=$

6. $\% =$

7. $**=$

8. $//=$

Operações Lógicas

1. AND
2. OR
3. NOT

AND Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Operações Relacionais

1. Igualdade ($==$)
2. Desigualdade (\neq)
3. Maior e menor ($>$ e $<$)
4. Maior OU igual (\geq)
5. Menor OU igual (\leq)

Desafio 2 - Lucky Game

1. Inserir o nome dos dois jogadores
2. Cada jogador "randomizará" um valor de 0 a 9
3. Se a soma dos dois valores for divisível por 2, os dois ganharão
4. Se os dois valores forem divisíveis por 3, haverá um empate
5. Se somente um jogador tiver um valor divisível por 3, ele será o ganhador
6. Senão, os dois perdem

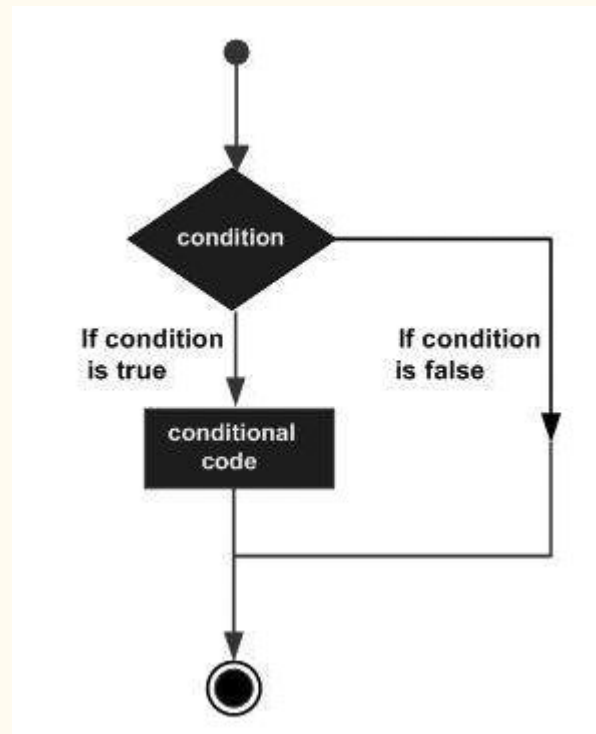
Aula 4

Controle de Fluxo

- O que é fluxo? E controle de fluxo?
- Estruturas de controle de fluxo
 - Controle de Decisão
 - Repetição
 - Sequencial

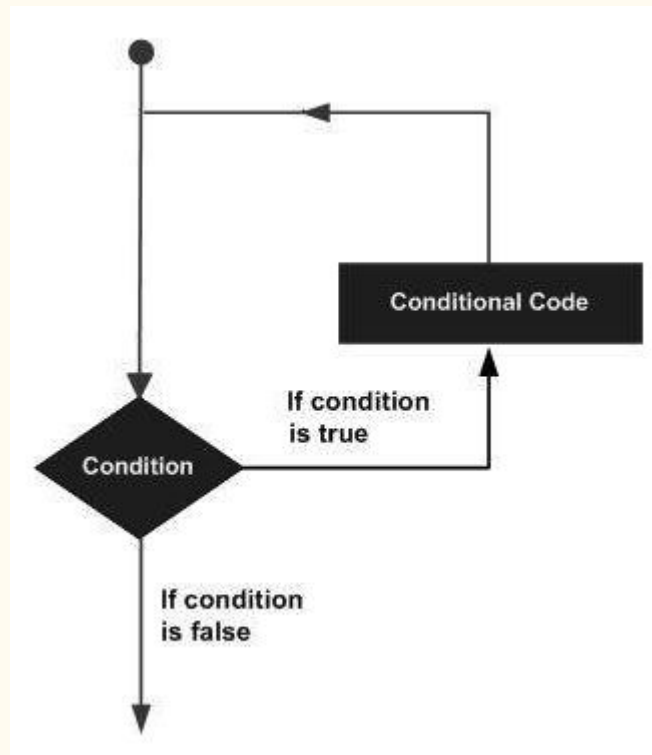
Estruturas de Controle de Fluxo - Controle de Decisão

1. O tipo **Boolean**
2. Tomada de decisão baseada em uma variável condicional
3. Bloco **indentado** será executado
4. **if-else**



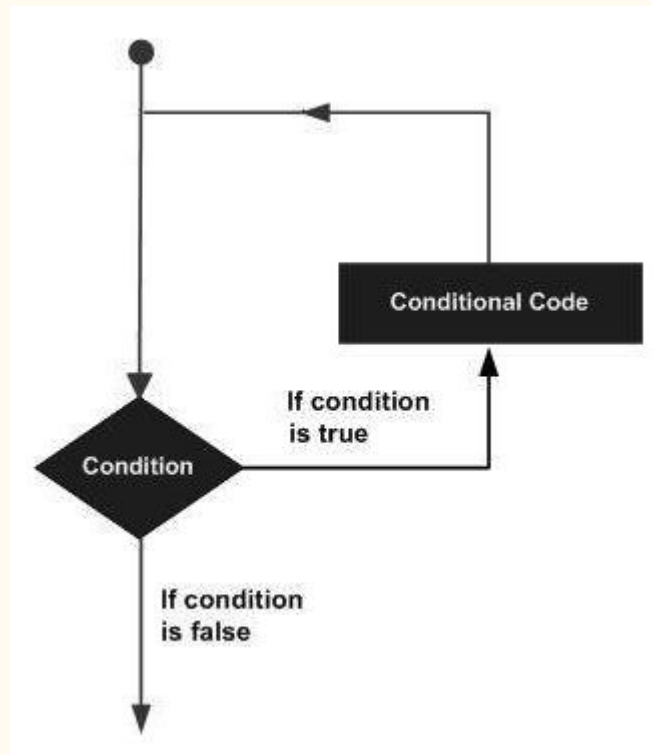
Estruturas de Controle de Fluxo - Loops

1. Também avalia uma variável booleana
2. Volta para o começo do bloco enquanto a condição é verdadeira
3. Sai do bloco assim que se tornar falsa
4. **while**, **for** e loops aninhados
5. Controle de loop - **break**, **continue** e **pass**



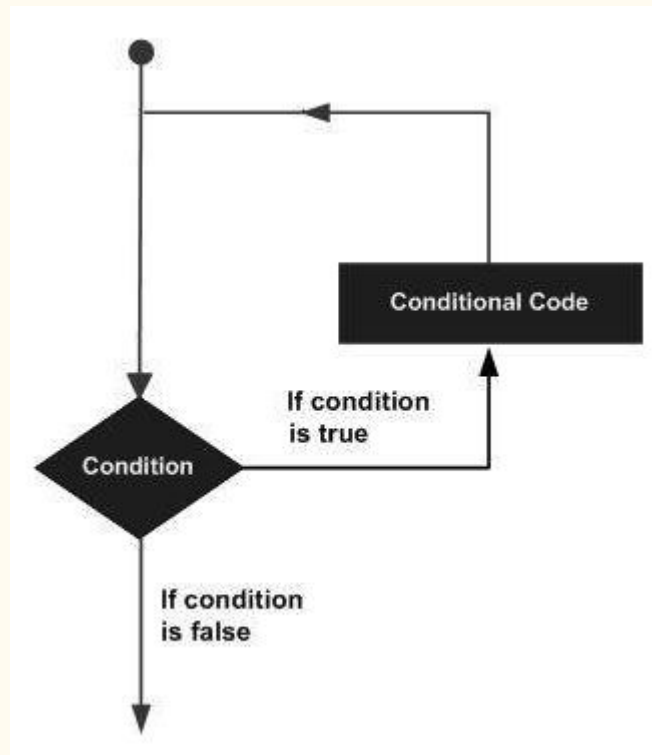
Desafio 3 - Números divisíveis por 7

1. Print todos os números divisíveis por 7
2. Valores deverão ser entre [0, 100]
3. Dica: `for x in range(6):`



Desafio 4 - Números divisíveis por 10, random

1. Crie um while-loop infinito
2. Cada loop vai randomizar um valor inteiro de [0,100]
3. Se o valor não for divisível por 10 print, caso contrário
 - a. "X nao eh divisivel por 10"
 - b. "Y eh divisivel por 10"
4. Dica: `while BOOL_CONDITION:`



Estruturas de Controle de Fluxo - Sequencial

1. Modo padrão de interpretação
2. Linha X será executada um passo antes da linha $X+1$
3. Linhas em branco serão ignoradas
4. Linhas com comentários serão ignoradas

Aula 5

Recebendo dados - Rápida
introdução

- Inteiros
- String



Recebendo dados

1. `valor_str = input(...)`
2. `valor_int = int(input(...))`

Aula 6

Desafio - Calculadora

- Deverá ter uma mensagem inicial de boas vindas
 - Cada input deverá ser escrito um por um. Isso significa que operandos e operadores serão separados
 - O programa deverá emitir uma mensagem de erro caso o padrão "NUMERO OPERACAO NUMERO" não seja respeitada.
 - Erros com operações desconhecidas ou valores impróprios devem ser alertadas
 - Programa deve rodar até que seja solicitado uma parada com a palavra "EXIT"
 - Caso contrário, a calculadora deverá solicitar os valores novamente
 - O resultado deverá ser sempre printado
 - Deverá ter uma mensagem de despedida
-