

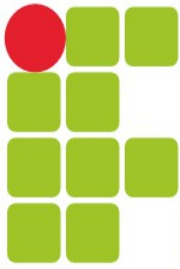
INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE  
Campus Parnamirim

# Introdução a Programação

Prof. André Gustavo Duarte de Almeida  
[andre.almeida@ifrn.edu.br](mailto:andre.almeida@ifrn.edu.br)  
[docente.ifrn.edu.br/andrealmeida](http://docente.ifrn.edu.br/andrealmeida)

Aula 02 – Primeiro Programa





# Roteiro

- Primeiros Passos
- Variáveis
- Expressões
- Comandos
- Estrutura de controle (Decisão)

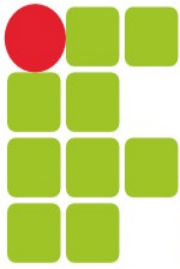




# Primeiros Passos

## •Python

- [www.python.org](http://www.python.org)
- Linguagem Interpretada
- Utilizada em diversas instituições como primeira linguagem de Programação
- Que usa Python: Google, Nasa, Apple ....



# Primeiros Passos

## •Python

- Para testar nossos programas, vamos utilizar a versão online do interpretador do Python
  - Acessar: <http://repl.it/languages/Python>
  - Ou Abrir o programa Python IDLE
  - Primeiro comando
  - No editor, digite:
  - `print("Meu Primeiro Programa")`



# Primeiros Passos

- Usando como Calculadora
- Em Python, podemos escrever expressões numérica e verificar o resultado
- Para computadores alguns símbolos da matemática são diferentes:
  - +(Adição)  $\rightarrow$  +
  - x(Multiplicação)  $\rightarrow$  \*
  - -(Subtração)  $\rightarrow$  -
  - \_\_ (Divisão)  $\rightarrow$  /
  - Potência  $\rightarrow$  \*\*



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE  
Campus Parnamirim

# Primeiros Passos

- Vejam algumas expressões:

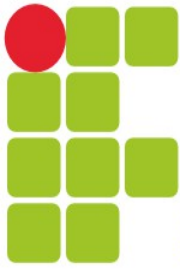




# Variáveis

## • Conceito

- O que significa dentro da matemática?
- Em programação, significa que temos um espaço na memória reservado para armazenar um valor qualquer, que pode ser modificado/lido a qualquer momento durante nosso programa
- Para criar um variável em python, basta definir o nome da mesma e atribuir algum valor inicial

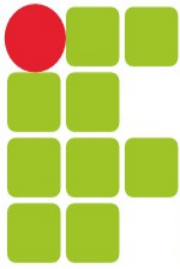


# Variáveis

## • Utilizando

- Considere que tenhamos a seguinte expressão aritmética:  $2-8/4$ . Para executar basta digitar a expressão junto com o comando print
- `print(2-8/4)`
- Se quisermos trocar o 8 por 12, por exemplo, precisamos redigitar a expressão
- Para evitar tal situação, podemos trocar 8 por um variável por exemplo com o nome x e atribuir o valor inicial como 12 e modificar sempre que necessário, sem mexer na expressão





# Variáveis

- Utilizando

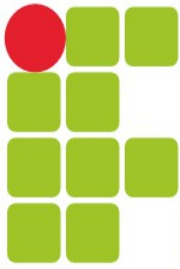
  - $x=12$

  - `print (2-x/4)`

- Mas se quisermos, por exemplo, usar o resultado da expressão para realizar outro cálculo?!

- Exemplo

```
x=12
y=2-x/4
z=(y**2)*x
print (z)
```



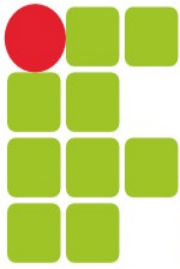
## • Nomenclatura

- Para definir o nome das variáveis, temos algumas regras e boas práticas
- Nome de variáveis não podem conter: espaços em branco e caracteres como: { } ( ) + - / \* , Espaço em branco dentre outros
- Não deve começar com números
- Evite o uso de acentos e ç
- Em Python, letras maiúsculas e minúsculas são diferentes
- nomecompleto é diferente de nomeCompleto



# Variáveis

- Utilizando – Exercício
- Escreva um programa que calcula a área de um retângulo.  $\text{Área} = \text{Base} * \text{Altura}$
- Use variáveis e defina os valores da base e altura. Imprima na tela o valor da área



# Variáveis

- Até agora vimos variáveis numéricas
- Porém podemos armazenar diversos tipos de dados
- Mas, o que é tipo de dados???
- Tipo de Dados
  - Determina a natureza e a forma como uma informação é armazenada
  - Numérico, Texto, Lógico, são alguns tipos de dados reconhecidos por um computador



# Variáveis

## • Tipos Lógicos

- **True** ou **False**, equivalem a verdadeiro e falso, ou também a 0 e 1
- Unidade básica, equivalente ao bit
- Exemplo
  - achou=True
  - perdeu=False



# Variáveis

## • Tipos de Texto

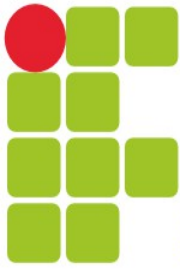
- Cadeia de caracteres, que podem incluir letras e números
- Normalmente delimitados por “ ” aspas duplas
- Exemplos
  - nome=”André Gustavo Duarte”
  - movimento=”papel”



# Expressões

## • Conceito

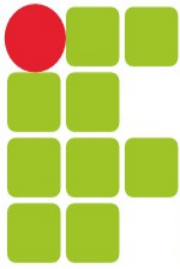
- Utilizamos mesmo de maneira inconscientes expressões aritméticas
- Expressões aritméticas, combinam variáveis, valores e operadores numéricos
- Usamos parêntesis para determinar a precedência de operadores
- Lembram da precedência de operadores?!



# Expressões

- Avaliando Expressões
- Testem as seguintes expressões
- $12-6/3$ ,  $(22-12)/5$
- Na primeira expressão, a divisão é avaliada antes da subtração
- O padrão de avaliação é da esquerda para direita.
- Na dúvida, use os parêntesis!!!





# Expressões

- Expressões Lógicas ou Booleanas
  - Expressões que produzem valores falsos(0) ou verdadeiros(1)
  - Utilizam operadores relacionais junto com operadores booleanos

Operador	Significado
==	Igualdade
>	Maior
<	Menor
>=	Maior ou igual
<=	Menor ou Igual

```
x=2  
y=3  
print x>2  
print ((x-y)==x)  
print((x**2)<5)
```



## • Operadores Lógicos

- Implementam a lógica booleana
- Utilizam o conceito de tabela verdade para avaliar uma expressão
- Cada operador possui sua tabela verdade

Operador	Significado
not	Negação
and	E Lógico
or	Ou Lógico



# Expressões

- Operador not

- Realiza operação de negação, invertendo o valor da expressão booleana

Valor V1	Not V1
True	False
False	True



# Expressões

## • Operador and

- Lembra uma Multiplicação
- Será True apenas se as duas expressões booleanas forem verdadeiras

V1	V2	V1 and V2
True	False	False
False	True	False
False	False	False
True	True	True



# Expressões

## • Operador or

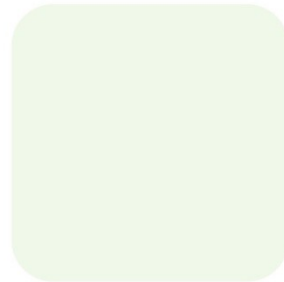
- Lembra uma Adição
- Será False apenas se as duas expressões booleanas forem Falsas

V1	V2	V1 and V2
True	False	True
False	True	True
False	False	False
True	True	True



# Expressões

- Ordem de avaliação
- Os operadores lógicos são avaliados na seguinte ordem:
- not  $\rightarrow$  and  $\rightarrow$  or

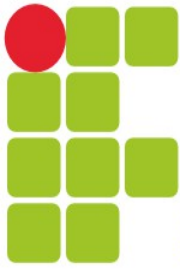




# Expressões

- Considere a seguinte expressão booleana
- $(A > B)$  and C or D

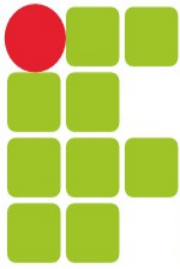
A	B	C	D	Resultado
1	2	True	False	??
10	3	False	False	??
5	1	True	True	??



# Entrada de Dados

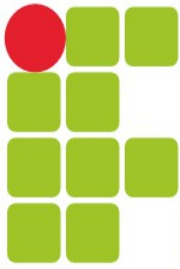
- A grande maioria dos programas necessita de dados externos para funcionar e para garantir que o mesmo funciona de acordo com os valores determinados pelo usuário
- Nesse momento a entrada será pelo teclado utilizando a tela em modo texto
- Para realizar entrada de dados em Python, usamos o comando:
- `input()`. Entre os parêntesis podemos colocar alguma informação para o usuário





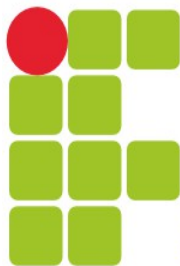
# Entrada de Dados

- Exemplo
- `nome=input("Digite seu Nome")`
- `print "Seu Nome é "+nome`
- Dependendo da versão do Python, talvez seja necessário usar `raw_input`, no lugar do `input`
- Na versão 3.0 devemos colocar os elementos do comando `print` entre parêntesis
- Na versão 3 precisamos converter a entrada de texto para número (inteiro e float)



# Entrada de Dados

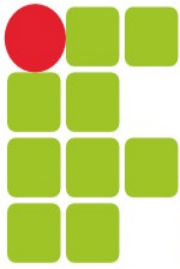
- Convertendo
- Convertendo para float (Número Real)
- `base=float(input("Digite a base"))`
- Convertendo para inteiro (int)
- `idade=int(input("Digite a idade"))`



# Exercício

- Refaça o exercício do retângulo, solicitando agora a entrada de dados do usuário
- Escreva um programa que pergunte a quantidade de KM percorridos por um carro alugado pelo usuário, assim como a quantidade de dias pelos quais o carro foi alugado. Calcule o preço a pagar, sabendo que o carro custa 60,00 por dia e 0,15 KM por KM rodado





# Estruturas de Controles

## • Comando if

- Determina se um trecho de código será executado sem uma condição for verdadeira
- A condição deve ser obrigatoriamente uma expressão booleana
- Ex: Dados duas variáveis a e b numéricas, qual tem o maior valor

```
if (a>=b):  
    print(a)
```

```
if (b>a):  
    print(b)
```



# Estruturas de Controle

## •Else

- No problema anterior, se a primeira avaliação ( $a \geq b$ ) for falsa, automaticamente sabemos que o valor de  $b$  é maior
- Não seria necessário testar de novo
- Para situações em que todas as possibilidades de uma expressão/problemas foram testadas, podemos substituir o último if por um else

- Exemplo Anterior    `if (a >= b):`  
                          `print(a)`  
                          `else:`  
                          `print(b)`



# Estruturas de Controle

## • IF Aninhando

- Dentro de um IF podemos adicionar qualquer tipo de comando, inclusive outro if
- Exemplo: Dado 3 Números determinar o maior deles

```
if (a>=b):  
    if(a>=c):  
        print(a)  
if (b>a):  
    if(b>c):  
        print(b)  
if (c>a):  
    if(c>b):  
        print (c)
```



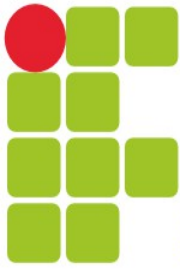
# Estruturas de Controle

## • IF Aninhado - ELIF

```
if (a>=b):  
    if(a>=c):  
        print (a)  
if (b>a):  
    if(b>c):  
        print(b)  
if (c>a):  
    if(c>b):  
        print(c)
```



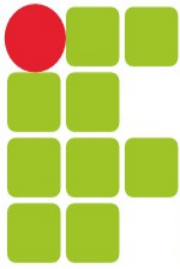
```
if (a>=b) and (a>=c):  
    print(a)  
elif (b>=a) and (b>c):  
    print(b)  
else:  
    print(c)
```



## •Atividade

- Imagine que queremos implementar o jogo de Pedra, Papel e Tesoura
- Lembra como funciona?
  - Pedra ganha da tesoura
  - Tesoura ganha do Papel
  - Papel ganha da Pedra
- Considerando que seu programa vai receber o movimento dos dois jogadores, como podemos representar a solução:
- Linguagem Natural





# Exercício

## • Pedra, Papel e Tesoura

- Tentando em linguagem Natural
- Receber os movimentos dos jogadores
- Avaliar os movimentos:
  - Se o Jogador01 for Papel ele ganha de Pedra e empata se for Papel
  - Se o Jogador01 for Tesoura ele ganha de Papel e empata se for Tesoura
  - Se o Jogador01 for Pedra ele ganha de Tesoura e empata se for Pedra
- Caso Contrário o Jogador02 é o vencedor



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE  
Campus Parnamirim

# Exercício

- Implementando Pedra, Papel e Tesoura

