

Introdução à Programação com Python

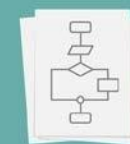
ICC - Introdução à Ciência da Computação

UFCG/UASC

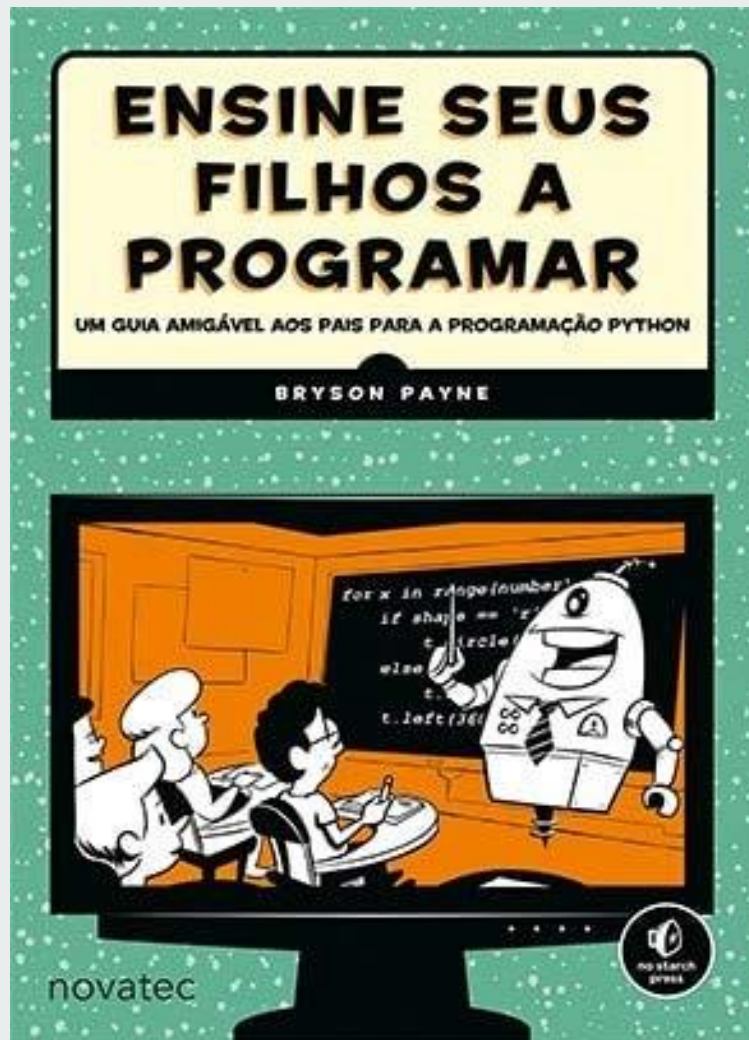
Prof. Roberto M. de Faria

Por que aprender programação?

- ajuda a **entender como os softwares funcionam**, e nos tornam, potencialmente, **melhores usuários**;
- **capacita** o engenheiro a **desenvolver pequenos sistemas** para utilidade no **dia a dia**;
- **bom para qualquer carreira, como um segundo idioma**;
- **prepara** melhor os jovens para **atuarem na vida profissional** no mundo moderno;
- **não é tão difícil de aprender.**



Por que aprender programação?



As crianças de hoje já estão conhecendo algoritmos na escola primária.

Por que aprender programação?

Ainda não está convencido? Toma aqui então:

<http://engenharialivre.com/programacao-uma-habilidade-essencial-para-o-engenheiro/>

<http://engenharialivre.com/por-que-engenheiros-devem-aprender-programar/>

<https://www.ctrlplay.com.br/entenda-a-importancia-da-programacao-para-as-profissoes-do-futuro/>

O que vamos aprender?

Você não vai aprender Python, necessariamente!



Você vai aprender a desenvolver **ALGORITMOS**.



O que é um programa?

Um **programa de computador** é um conjunto de instruções que descrevem as operações a serem realizadas por um computador para realizar uma tarefa

```
31: }
32:     else // numero < 0
33:     {
34:         // numero < 0
35:         printf("O numero %d 'e menor qu", numero);
36:         negativos = negativos + 1;
37:     }
38:     else // numero = 0
39:     {
40:         printf("O numero %d 'e igual a", numero);
41:         nulos = nulos + 1;
42:     }
43: }
44:
45: /* Apresentacao dos resultados. */
46: printf("\n\n");
47: printf("Na sequencia de %d numeros temos:\n", N);
48: printf(" - Numeros positivos: %d\n", positivos);
49: printf(" - Numeros negativos: %d\n", negativos);
50: printf(" - Numeros nulos: %d\n", nulos);
51:
52: /* Pausa para apreciacao dos resultados. Pressionar ENTER
53:    ou outra tecla qualquer para finalizar o programa */
54: system("pause");
55: return 0;
56: }
```



Conceitos

**Como os programas de computador são
feitos?**



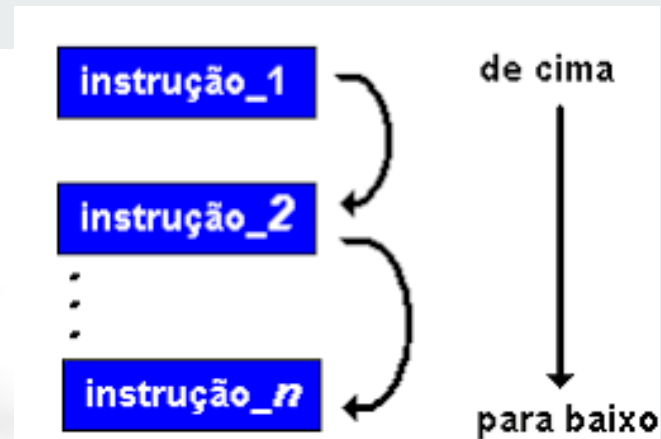
Conceitos

Como os programas de computador são feitos?



Conceitos

Como os programas de computador são
feitos?

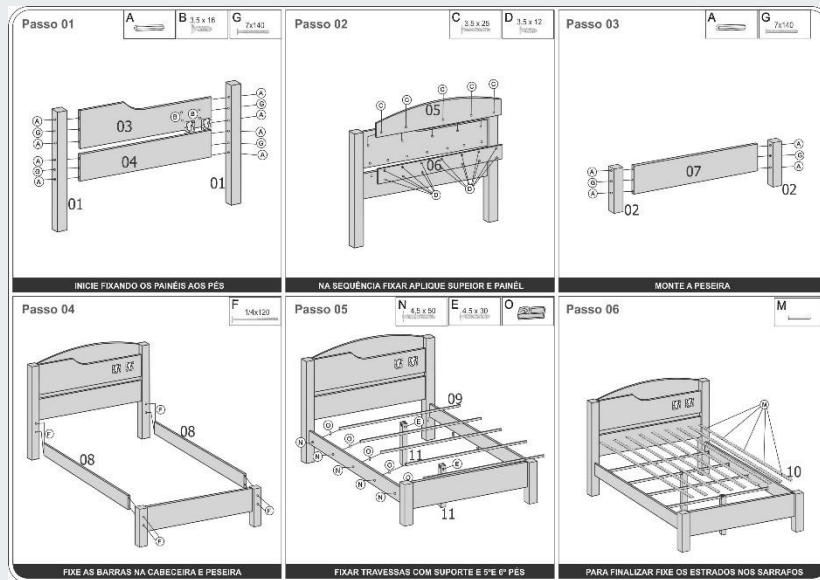


+



Conceitos

No mundo real:



+



Conceitos

Como os programas de computador são feitos?



Problema

- + Projeto (Objetivo/Meta)
- + Organização lógica das operações
- + Instruções
- + Linguagem entendida pela máquina

Conceitos

- **Lógica de programação:** técnica de encadear (ligar) pensamentos (ou instruções) para atingir determinado objetivo



Conceitos

- **Sequência Lógica:** passos executados até atingir um objetivo ou solução de um problema



Chamamos o conjunto de instruções obedecendo uma
sequencia lógica de **ALGORITMO**

Algoritmo

- Sequência lógica de instruções para realização de uma determinada tarefa
- **Instruções**: conjunto de regras ou normas definidas para a realização ou emprego de algo.
- Na computação, algoritmo é o que **orienta um computador** com uma ação ou várias ações a serem executadas.



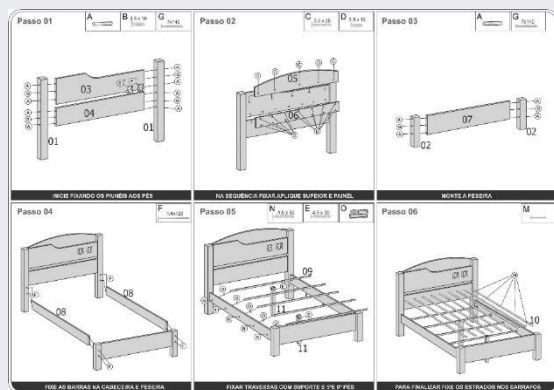
Algoritmo
Computacional

Algoritmo

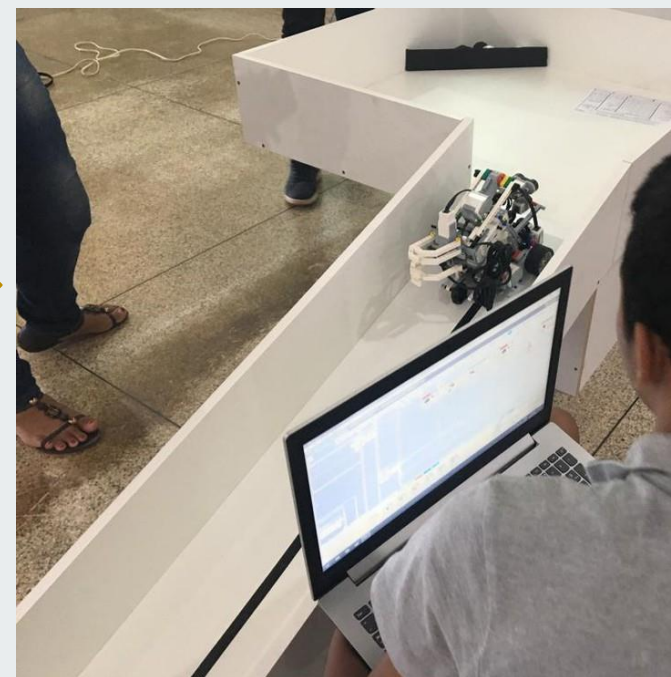
- Um conjunto de instruções ordenadas e finitas que quando executadas irão alcançar a **solução de um problema** (exemplos: receita de comida, manuais de eletro-eletrônicos, uso do Waze/Maps, etc.);
- Existem também os **algoritmos computacionais**, que são aqueles cujas instruções podem ser executadas por um computador.

Algoritmo

No mundo real:



```
{  
  divs();  
  var atpos=inputs[i].indexOf("#");  
  var dotpos=inputs[i].lastIndexOf(".");  
  if (atpos<1 || dotpos<atpos+2 || dotpos>inputs[i].length-1)  
    document.getElementById("errEmail").innerHTML += "  
  else  
    document.getElementById(div).innerHTML += "  
  atpos=inputs[i].lastIndexOf("#");  
  dotpos=inputs[i].lastIndexOf(".");  
  if (atpos<1 || dotpos<atpos+2 || dotpos>inputs[i].length-1)  
    document.getElementById("errEmail").innerHTML += "  
}
```



Exemplo de **algoritmo computacional**

Resolução de problemas computáveis

- Para resolver problemas computáveis é necessário que alguns passos sejam realizados:
 1. Entender o **problema**
 2. Conhecer os dados do problema
 3. Desenvolver um método de obtenção da solução - **algoritmo**
 4. Otimizar o método de obtenção da solução
 5. Descrever o algoritmo com uma **linguagem de programação**
 6. Compilar ou **interpretar** o programa
 7. Testar o programa

- Como os programas de computador são feitos?
 - Usando **linguagens de programação** para descrever os algoritmos computacionais
 - Existem inúmeras linguagens para os mais variados propósitos



Linguagem de programação

- Cada linguagem define uma sequência de símbolos que manipulam a memória, realizam operações e trabalham com a entrada e saída da máquina
- Existem diferentes linguagens de programação, cada uma com seus símbolos, palavras e regras
- **Python** é uma das linguagens mais utilizadas, simples e poderosa
- Utilizadas em diferentes instituições
- Bastante próxima a MatLab



Programa



- Em **Python**, um programa é uma sequência de instruções (comandos), com sintaxe própria, que o computador deve executar para resolver um problema computável

Comando de programa

Uma ordem dada ao computador.

Exemplo de comando simples:

```
meu_numero = 10
```

Tradução:

"O pedaço de memória de nome meu numero,
a partir de agora, armazena o valor 10".

Variável

Entidade que armazena valores e representa estes valores armazenados

```
dinheiro = 500  
dinheiro = dinheiro - 100  
print(dinheiro)
```

Tradução:

"dinheiro armazena o valor 500"

"dinheiro armazena o valor que estava armazenado em dinheiro, menos 100"

"imprima o valor armazenado em dinheiro"

Valores, variáveis e expressões

Valores:

12

-5

16.97

Expressões: sequências de valores, variáveis e símbolos que, ao serem interpretados, retornam um valor:

$2 + 2 * (\text{peso} - 5)$

$2 - 2 * 3 - 1$

`base ** 4`

Precedência das operações

Assim como na matemática, () vem antes de **, que vem antes de * e /, que vem antes de + e -, ou seja:

$$2 + 5 * (3 + 1) ** 2$$

$$2 + 5 * 4 ** 2$$

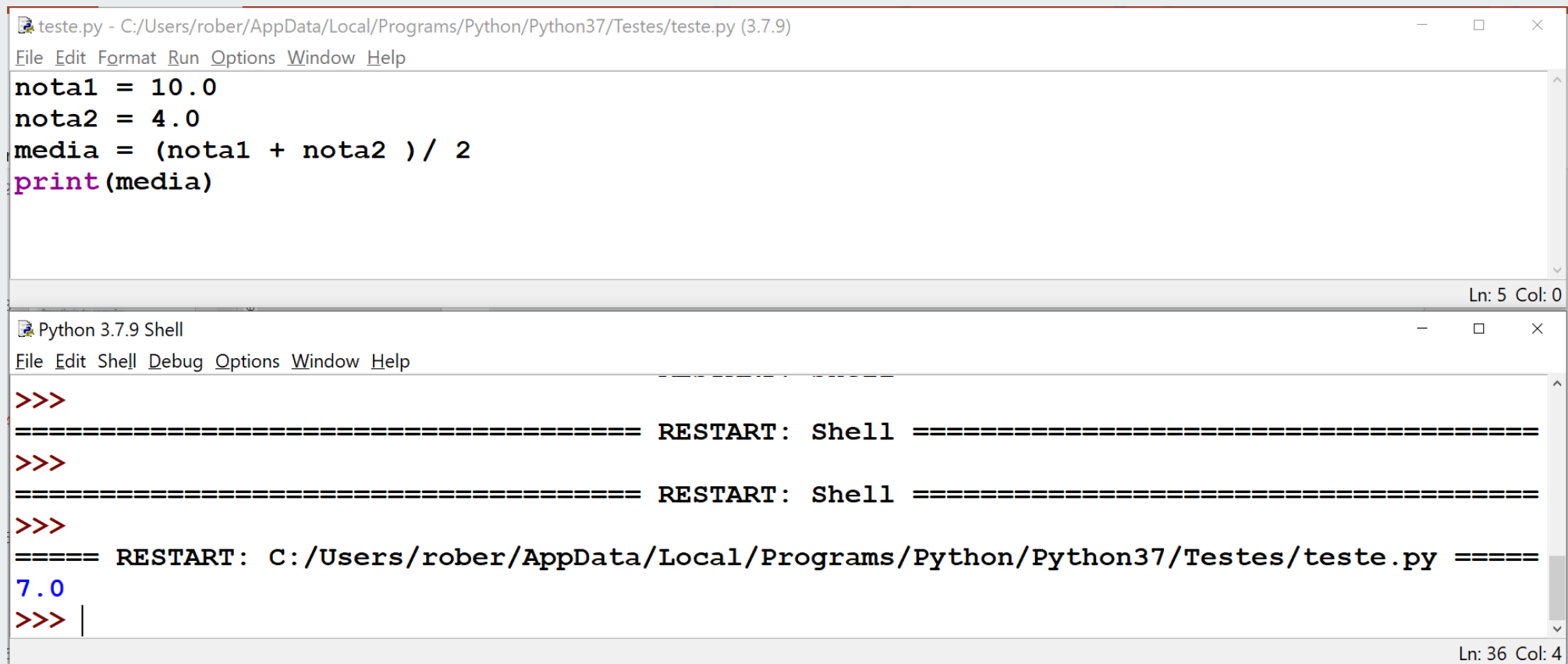
$$2 + 5 * 16$$

$$2 + 80$$

$$82$$

Saída de dados

O `print()` mostra valores de dados na janela de saída do Python.



The screenshot displays two windows from a Python IDE. The top window, titled 'teste.py - C:/Users/rober/AppData/Local/Programs/Python/Python37/Testes/teste.py (3.7.9)', contains the following Python code:

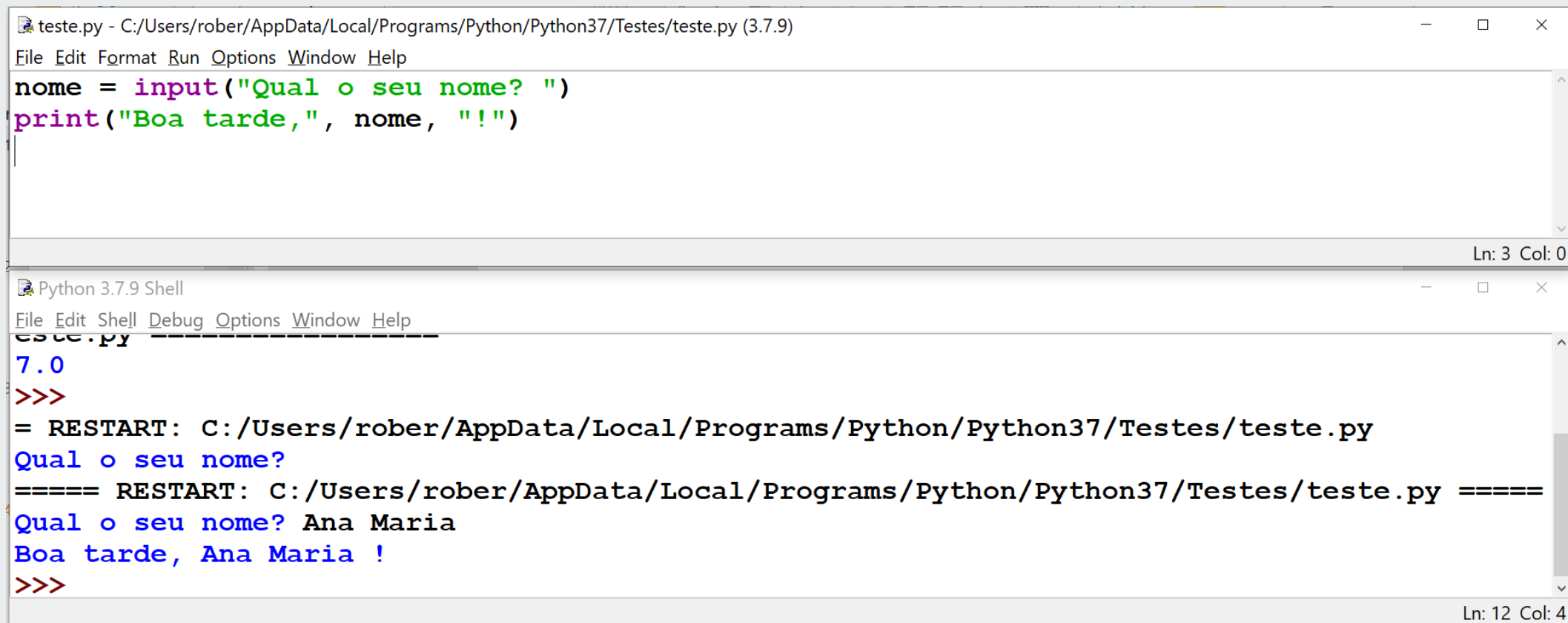
```
nota1 = 10.0
nota2 = 4.0
media = (nota1 + nota2) / 2
print(media)
```

The bottom window, titled 'Python 3.7.9 Shell', shows the execution output. It includes three 'RESTART: Shell' messages, followed by the calculated value '7.0' in blue text. The shell prompt '>>>' is visible at the end of the line.

```
>>>
===== RESTART: Shell =====
>>>
===== RESTART: Shell =====
>>>
===== RESTART: C:/Users/rober/AppData/Local/Programs/Python/Python37/Testes/teste.py =====
7.0
>>> |
```

Entrada de dados

Para receber dados da linha de comando em Python, faça:



The image shows a screenshot of a Python IDE with two windows. The top window, titled 'teste.py - C:/Users/rober/AppData/Local/Programs/Python/Python37/Testes/teste.py (3.7.9)', contains the following Python code:

```
nome = input("Qual o seu nome? ")
print("Boa tarde,", nome, "!")
```

The bottom window, titled 'Python 3.7.9 Shell', shows the execution of the script. It displays the Python version '7.0', a restart message, the prompt 'Qual o seu nome?', the user input 'Ana Maria', and the output 'Boa tarde, Ana Maria !'.

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
teste.py -----
7.0
>>>
= RESTART: C:/Users/rober/AppData/Local/Programs/Python/Python37/Testes/teste.py
Qual o seu nome?
===== RESTART: C:/Users/rober/AppData/Local/Programs/Python/Python37/Testes/teste.py =====
Qual o seu nome? Ana Maria
Boa tarde, Ana Maria !
>>>
```


Entrada de dados



Outro exemplo:

```
teste.py - C:/Users/rober/AppData/Local/Programs/Python/Python37/Testes/teste.py (3.7.9)
File Edit Format Run Options Window Help
nota1 = input("Informe a nota 1: ")
nota2 = input("Informe a nota 2: ")
media = (nota1 + nota2 )/ 2
print(media)
Ln: 5 Col: 0

Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/Users/rober/AppData/Local/Programs/Python/Python37/Testes/teste.py =====
Informe a nota 1: 10.0
Informe a nota 2: 4.0
Traceback (most recent call last):
  File "C:/Users/rober/AppData/Local/Programs/Python/Python37/Testes/teste.py", line 3, in
<module>
    media = (nota1 + nota2 )/ 2
TypeError: unsupported operand type(s) for /: 'str' and 'int'
>>>
Ln: 54 Col: 4
```

Tipos

Cada valor tem um conjunto de operações que podem ser feitas sobre ele. O que define quais operações podem ser feitas é o TIPO do valor. Um valor do TIPO inteiro (int) pode ser somado, subtraído, multiplicado, etc...

```
valor_inteiro = 2
outro_valor_inteiro = valor_inteiro * 2
print(outro_valor_inteiro)
```

Diferentes tipos

O tipo inteiro (int) representa os valores dos números inteiros. O tipo string (str) representa cadeias de caracteres. Os reais de ponto flutuante (float) são números com pontos decimais e frações. Veja exemplos:

```
valor_inteiro = 10  
valor_flutuante = 0.5  
valor_string = "Oi vc!"
```

Em Python, as aspas servem pra dizer que o que tiver dentro delas, será uma string (um texto).

Funções de conversão

Funções de conversão operam sobre valores e retornam um novo valor.

Exemplo, a função recebe um valor e converte esse valor para outro tipo:

```
valor_inteiro_truncado = int(10.5)
```

```
valor_inteiro = int("1")
```

```
valor_real = float("3.14159")
```

```
texto = str(123)
```

Entrada de dados



Outro exemplo:

```
teste.py - C:/Users/rober/AppData/Local/Programs/Python/Python37/Testes/teste.py (3.7.9)
File Edit Format Run Options Window Help
nota1 = float(input("Informe a nota 1: "))
nota2 = float(input("Informe a nota 2: "))
media = (nota1 + nota2 )/ 2
print(media)
Ln: 5 Col: 0

Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
<module>
    media = (nota1 + nota2 )/ 2
TypeError: unsupported operand type(s) for /: 'str' and 'int'
>>>
===== RESTART: C:/Users/rober/AppData/Local/Programs/Python/Python37/Testes/teste.py =====
Informe a nota 1: 10
Informe a nota 2: 4
7.0
>>> |
Ln: 59 Col: 4
```

Referências e créditos

- Estes slides foram compilados a partir dos trabalhos:
 - da Profa. Eliane Araújo na disciplina Introdução à Ciência da Computação/UASC/UFCG
 - do Prof. Maxwell Guimarães na disciplina Introdução à Programação/UASC/UFCG
 - do Prof. Roberto Faria nas disciplinas Introdução à Ciência da Computação e Introdução à Programação/UASC/UFCG