

PROGRAMAÇÃO DE COMPUTADORES

PRIMEIRO PROGRAMA

COPYRIGHT © 2024 DIATINF/CNAT/IFRN
JORGIANO VIDAL



- Sequenciamento de instruções
 - Resolução de problemas
 - Modelagem
- Mostrando valores
- Criação de um programa
- Execução
- Lendo valores
 - a função `input()`
 - Conversão para inteiro: função `int()`
 - Conversão para real: função `float()`
- Mostrando valores
 - a função `print()`
 - Parâmetros `sep` e `end`
 - Lendo mais de um valor por linha





- Um computador é capaz de entender/calcular uma instrução/expressão
- O resultado pode ser associado a uma **variável**
- As expressões/instruções podem manipular **valores**
 - Valores possuem um **tipo**
 - Vimos inteiros, reais e textos (*strings*)
- Expressões complexas podem ser divididas e **sequencializadas** para facilitar o entendimento



RESOLVER PROBLEMAS

- Equação do segundo grau
 - Qual(is) o(s) valor(es) de x_1 e x_2 ?
 - Precisamos dos valores de a , b e c
- Passos
 1. Saber os valores de a , b e c
 2. Calcular delta ($b^2 - 4*a*c$)
 3. Calcular raiz de delta (r_delta)
 4. Calcular $x_1 = (-b + r_delta) / (2*a)$
 5. Calcular $x_2 = (-b - r_delta) / (2*a)$

```
a=2
b=5
c=2
delta = 9
r_delta = 3
x1=(-5+3)/(2*2)
x2=(-5-3)/(2*2)
```



- Definir elementos do problema
 - Valores de **entrada**
 - **Operações** a serem realizadas
 - Valores de **saída**
- *Exemplo*: Tempo de deslocamento t
 - **Entrada**: Espaço e e velocidade v
 - **Operações**: fórmula $t = \frac{e}{v}$
 - **Saída**: Tempo de deslocamento



- Várias operações
- **Mesma** sequencia de operações, **SEMPRE!**
- Cálculo de média
 1. Definir nota 01. Armazena o valor na variável *nota01*
 2. Definir nota 02. Armazena o valor na variável *nota02*
 3. Multiplicar *nota01* por 2 e *nota02* por 3 e atribuir a variável *pontos*
 4. Dividir *pontos* por 5 e atribuir a variável *media*
 5. Mostrar o valor armazenado na variável *media*



PROGRAMA DE COMPUTADOR

- Sequencia de instruções
 - Arquivo texto com instruções a serem executadas
 - Em python, usamos a extensão **.py**

media.py

```
nota1=8.3  
nota2=9.2  
media=(nota1*2+nota2*3)/5  
print(media)
```

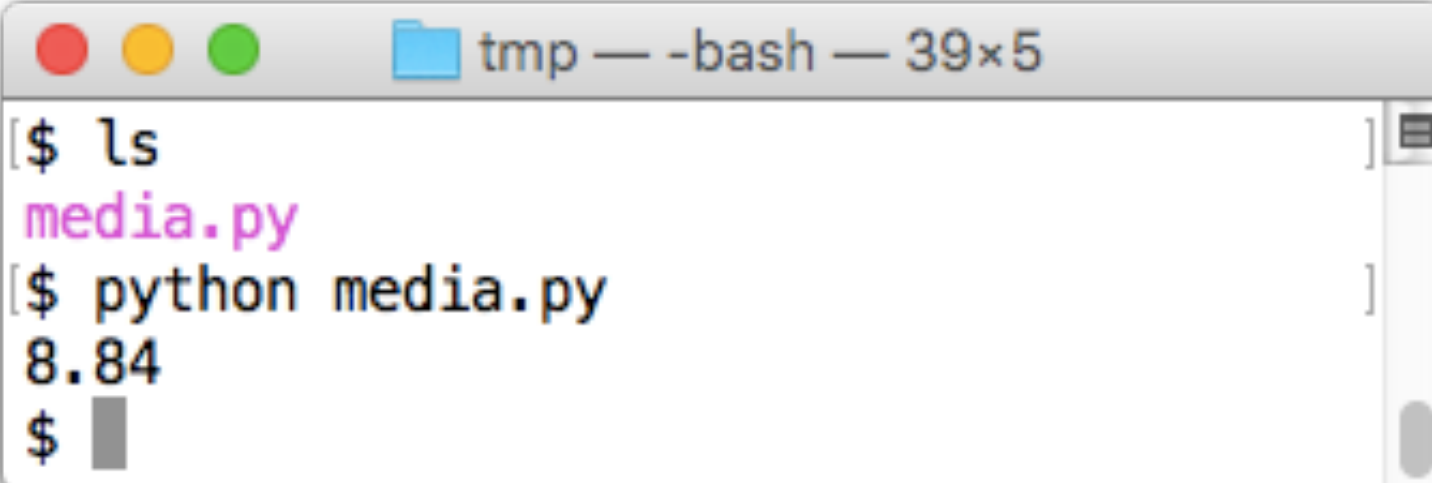


- Programa para editar arquivos de texto
 - Não contém formatação (rtf, doc, docs, odf, etc)
- O que usar
 - Bloco de notas (Win)
 - gedit (Linux)
 - nano (Linux)
 - **VSCode** (win, lin, etc)
 - Pico (Linux)
 - Vi, vim (Unix, linux)
 - notepad++ (windows)
 - SciTE (windows)
 - emacs (win, linux, etc)
 - jEdit (win, linux, etc)
 - Kate
 - SublimeText
 - etc...





- Na linha de comando:
`python NOME_ARQUIVO`
 - onde `NOME_ARQUIVO` é o arquivo salvo.



```
tmp — -bash — 39x5
[$ ls
media.py
[$ python media.py
8.84
$
```



MOSTRANDO VALORES

- Função `print()`
 - Imprime na console um texto
 - Sintaxe:
 - `print(CONTEÚDO)`

- Exemplos:

```
print("Olá mundo")
x=10
print(x)
y=x*2
print(y)
z=x/3.0
print(z)
print(x+y)
print(x*z+y*2-(x+y))
```

```
tmp — -bash — 40x8
Olá mundo
10
20
3.3333333333333335
30
43.333333333333334
[$
$
```



FUNÇÃO PRINT()



- Vírgula entre parâmetros

olaNome.py

```
nome="Jorgiano"  
print("Olá", nome, ", bom dia!")
```

- Espaço é adicionado

Olá Jorgiano , bom dia!



- Vírgula entre parâmetros

olaNome.py

```
nome="Jorgiano"  
print("Olá", nome, ", bom dia!")
```

- Espaço é adicionado

Olá Jorgiano , bom dia!



- Vírgula entre parâmetros

olaNome.py

```
nome="Jorgiano"  
print("Olá", nome, ", bom dia!")
```

- Espaço é adicionado

Olá Jorgiano , bom dia!



- Define separador



- Define separador

olaNome.py

```
nome="Jorgiano"  
print("Olá", nome, ", bom dia!", sep="X")
```




- Define separador

olaNome.py

```
nome="Jorgiano"  
print("Olá", nome, ", bom dia!", sep="X")
```

OláXJorgianoX, bom dia!



- Define separador

olaNome.py

```
nome="Jorgiano"  
print("Olá", nome, ", bom dia!", sep="X")
```

OláXJorgianoX, bom dia!



- Define separador

olaNome.py

```
nome="Jorgiano"  
print("Olá", nome, ", bom dia!", sep="X")
```

OláXJorgianoX, bom dia!

olaNome.py

```
nome="Jorgiano"  
print("Olá ", nome, ", bom dia!", sep="")
```



- Define separador

olaNome.py

```
nome="Jorgiano"  
print("Olá", nome, ", bom dia!", sep="X")
```

OláXJorgianoX, bom dia!

olaNome.py

```
nome="Jorgiano"  
print("Olá", nome, ", bom dia!", sep="")
```



- Define separador

olaNome.py

```
nome="Jorgiano"  
print("Olá", nome, ", bom dia!", sep="X")
```

OláXJorgianoX, bom dia!

olaNome.py

```
nome="Jorgiano"  
print("Olá", nome, ", bom dia!", sep="")
```

Olá Jorgiano, bom dia!



- Finalizador de linha



- Finalizador de linha

olaNome.py

```
nome="Jorgiano"  
print("Olá")  
print(nome)  
print(", bom dia!")
```



- Finalizador de linha

olaNome.py

```
nome="Jorgiano"  
print("Olá")  
print(nome)  
print(", bom dia!")
```



```
Olá↵  
Jorgiano↵  
, bom dia!↵
```




- Finalizador de linha

olaNome.py

```
nome="Jorgiano"  
print("Olá")  
print(nome)  
print(", bom dia!")
```



```
Olá↵  
Jorgiano↵  
, bom dia!↵
```

olaNome.py

```
nome="Jorgiano"  
print("Olá",end="")  
print(nome,end="")  
print(", bom dia!")
```



- Finalizador de linha

olaNome.py

```
nome="Jorgiano"  
print("Olá")  
print(nome)  
print(", bom dia!")
```



```
Olá↵  
Jorgiano↵  
, bom dia!↵
```

olaNome.py

```
nome="Jorgiano"  
print("Olá",end="")  
print(nome,end="")  
print(", bom dia!")
```



```
OláJorgiano, bom dia!↵
```



- Programa espera do usuário dados
 - Cálculo da média deve ser feito para quaisquer notas
- Instrução para ler uma linha
 - `input()`
 - Lê linha como *string*
 - Programa para esperando usuário digitar algo
 - Leitura é feita ao pressionar a tecla **[ENTER]**



ola_mundo.py

```
nome=input()  
print("Olá", nome)
```

```
tmp — -bash — 40x6  
[$ ls  
ola_mundo.py  
[$ python ola_mundo.py  
Jorgiano  
Olá Jorgiano  
$
```



- Função `input()` lê *string*
 - necessário **converter** para inteiro ou real
- Para inteiro
 - função `int()`
- Para real
 - função `float()`



- Função `input()` lê *string*
 - necessário **converter** para inteiro ou real
- Para inteiro
 - função `int()`
- Para real
 - função `float()`

`media.py`

```
nota1_txt=input()  
nota1=float(nota1_txt)  
nota2_txt=input()  
nota2=float(nota2_txt)  
media=(nota1*2+nota2*3)/5.0  
print(media)
```



- Função `input()` lê *string*
 - necessário **converter** para inteiro ou real
- Para inteiro
 - função `int()`
- Para real
 - função `float()`

media.py

```
nota1_txt=input()  
nota1=float(nota1_txt)  
nota2_txt=input()  
nota2=float(nota2_txt)  
media=(nota1*2+nota2*3)/5.0  
print(media)
```

```
tmp - -bash - 40x6  
[$ python media.py  
8  
9.1  
8.66  
$ ]
```



- Função `input()` lê *string*
 - necessário **converter** para inteiro ou real
- Para inteiro
 - função `int()`
- Para real
 - função `float()`

media.py

```
nota1_txt=input()  
nota1=float(nota1_txt)  
nota2_txt=input()  
nota2=float(nota2_txt)  
media=(nota1*2+nota2*3)/5.0  
print(media)
```



```
nota1=float(input())  
nota2=float(input())  
media=(nota1*2+nota2*3)/5.0  
print(media)
```

```
tmp — -bash — 40x6  
[$ python media.py  
8  
9.1  
8.66  
$
```




MAIS DE UM VALOR POR LINHA

- Divide a linha
 - `split()`
- Mapeia cada valor
 - `map()`



MAIS DE UM VALOR POR LINHA

- Divide a linha
 - `split()`
- Mapeia cada valor
 - `map()`

```
n1, n2 = input().split()  
nota1 = float(n1)  
nota2 = float(n1)  
media = (nota1*2+nota2*3)/5.0  
print(media)
```



MAIS DE UM VALOR POR LINHA

- Divide a linha
 - `split()`
- Mapeia cada valor
 - `map()`



MAIS DE UM VALOR POR LINHA

- Divide a linha
 - `split()`
- Mapeia cada valor
 - `map()`

```
notas = input().split()
nota1, nota2 = map(float, notas)
media = (nota1*2+nota2*3)/5.0
print(media)
```



MAIS DE UM VALOR POR LINHA

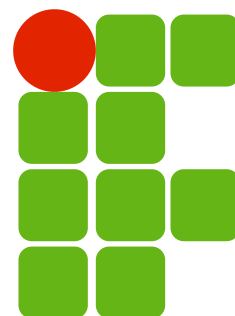
- Divide a linha
 - `split()`
- Mapeia cada valor
 - `map()`



MAIS DE UM VALOR POR LINHA

- Divide a linha
 - `split()`
- Mapeia cada valor
 - `map()`

```
nota1, nota2 = map(float, input().split())  
media = (nota1*2+nota2*3)/5.0  
print(media)
```



PROGRAMAÇÃO DE COMPUTADORES

PRIMEIRO PROGRAMA

COPYRIGHT © 2024 DIATINF/CNAT/IFRN
JORGIANO VIDAL