



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



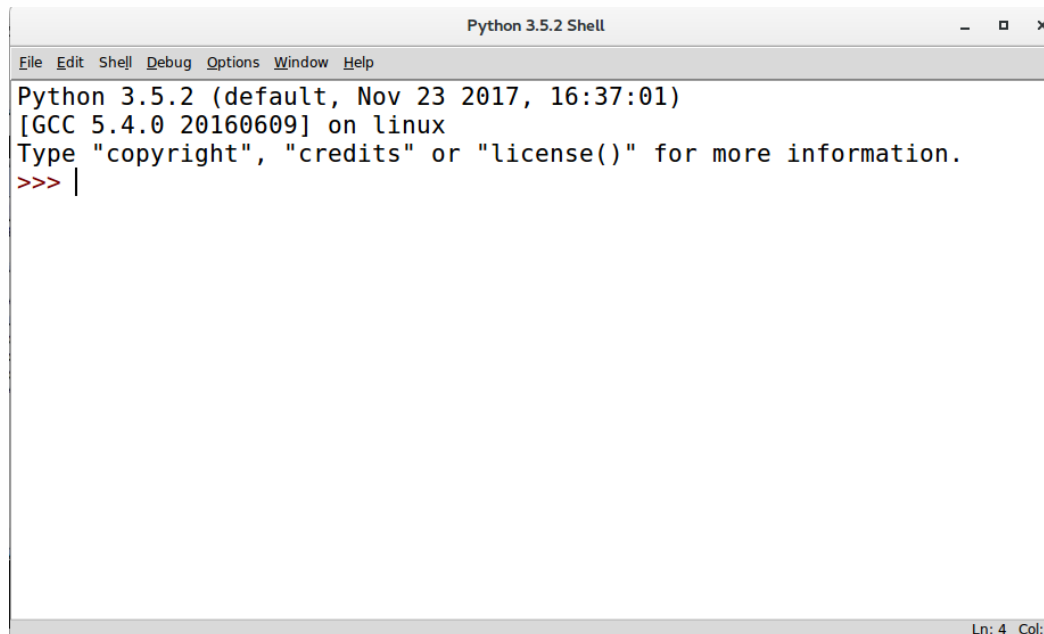
Fundamentos de Lógica e Algoritmos

Aula 2.7 – IDLE, Variáveis e Tipos

Prof. Dr. Bruno Moreno
bruno.moreno@ifrn.edu.br

IDLE

- O IDLE é um ambiente de desenvolvimento integrado para Python;
- O IDLE permite:
 - Escrever código, através do editor de código IDLE;
 - Executar o código, pelo interpretador **Python Shell**;



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (default, Nov 23 2017, 16:37:01)
[GCC 5.4.0 20160609] on linux
Type "copyright", "credits" or "license()" for more information.
>>> |
```

Ln: 4 Col: 4

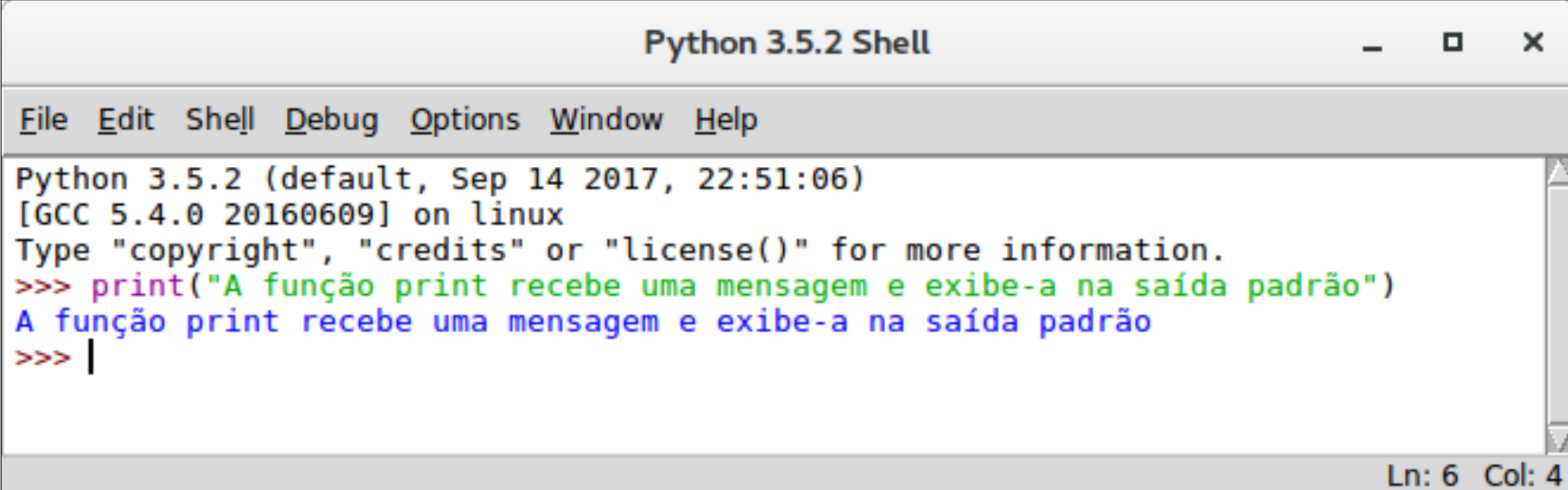
Uso do IDLE

- Ajuda em questões relacionadas à sintaxe:
 - Oferece **dicas de término** que aparecem quando se usa uma **função embutida**, como **print ()**;
 - As funções embutidas são chamadas de BIFs:
 - *Built-in Functions*;
 - O BIF **print ()** exibe mensagens na saída padrão;



Uso do IDLE

- O IDLE destaca os comandos do código:



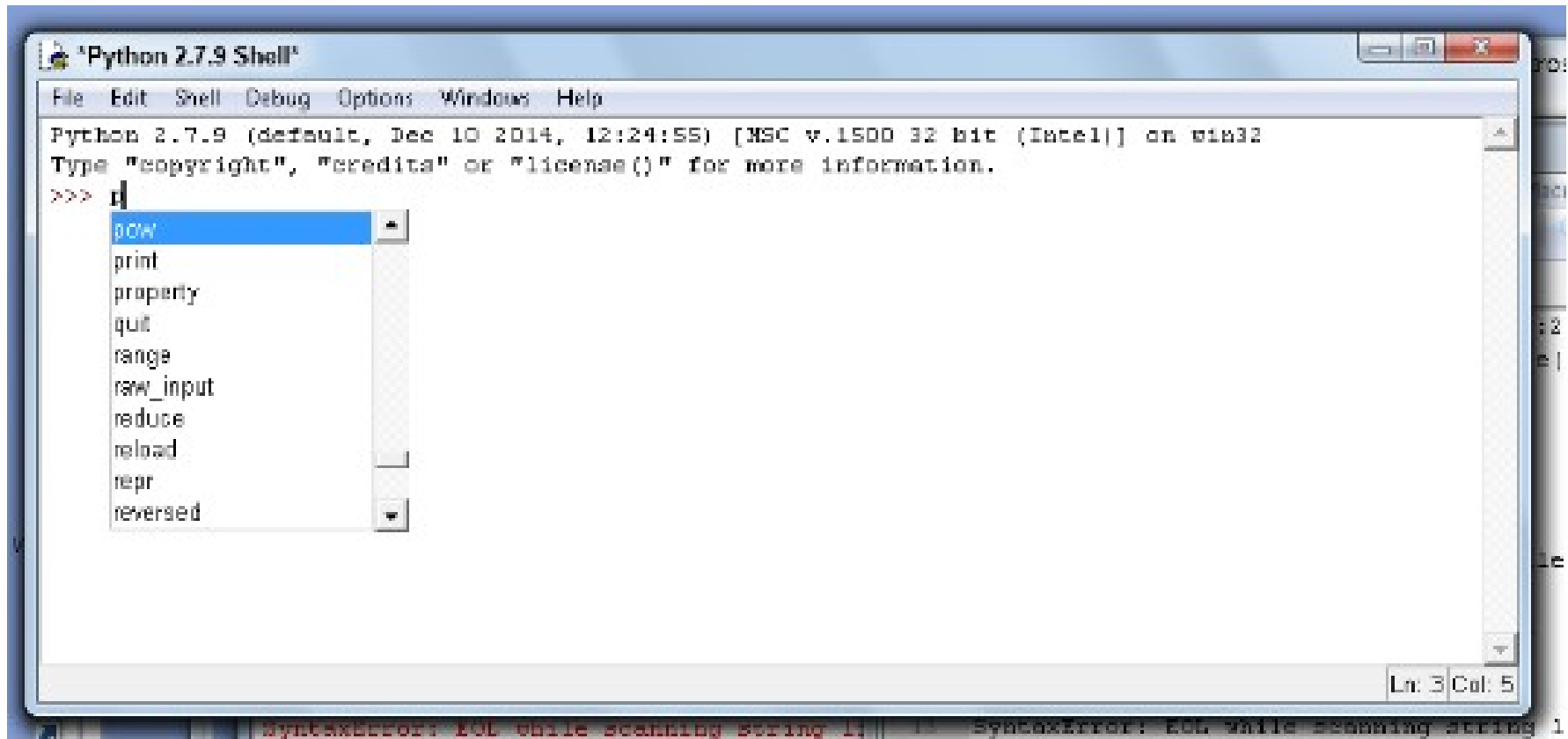
```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (default, Sep 14 2017, 22:51:06)
[GCC 5.4.0 20160609] on linux
Type "copyright", "credits" or "license()" for more information.
>>> print("A função print recebe uma mensagem e exibe-a na saída padrão")
A função print recebe uma mensagem e exibe-a na saída padrão
>>> |
```

Ln: 6 Col: 4

- As BIFs são roxas;
- As strings são verdes;
- As palavras-chave são laranjas;
- Quaisquer resultados são em azul;

Uso do IDLE

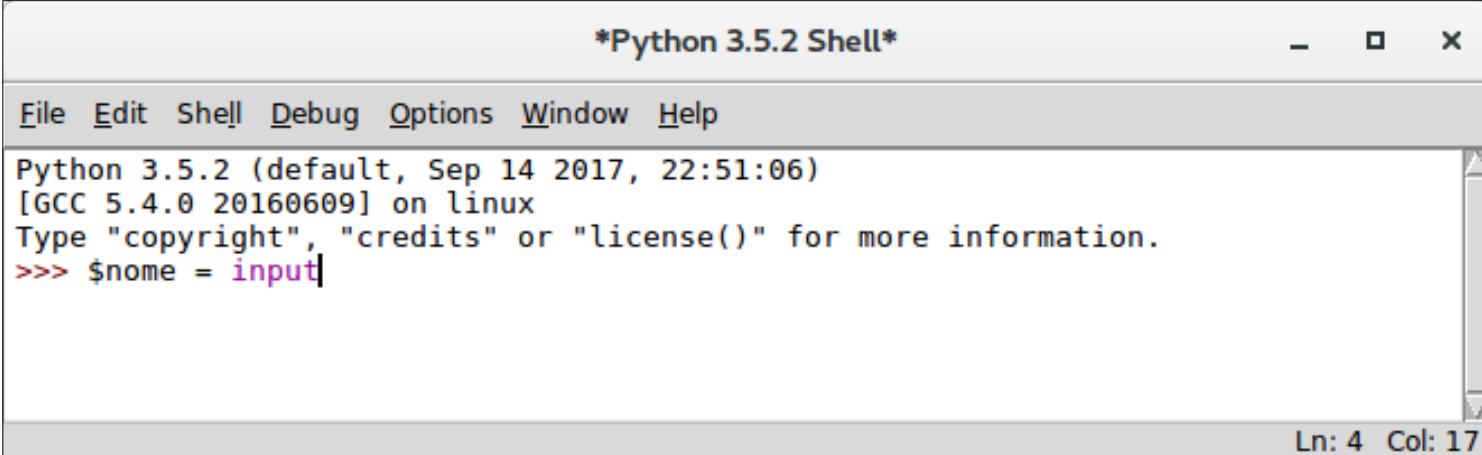
- O IDLE auxilia na completar seu código:
 - Digite um código qualquer e pressione as teclas CTRL+ESPAÇO:



The screenshot shows a window titled "Python 2.7.9 Shell" with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help). The main text area displays the Python version and system information: "Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win32". Below this, it says "Type 'copyright', 'credits' or 'license()' for more information." The prompt is ">>> d", and a dropdown menu is open, listing the following options: "pow", "print", "property", "quit", "range", "raw_input", "reduce", "reload", "repr", and "reversed". The "pow" option is currently selected. At the bottom right of the window, the status bar shows "Ln: 3 Col: 5".

Uso do IDLE

- O IDLE auxilia na completar seu código:
 - Digite um código qualquer e pressione a tecla TAB:




```
*Python 3.5.2 Shell*
```

File Edit Shell Debug Options Window Help

```
Python 3.5.2 (default, Sep 14 2017, 22:51:06)
[GCC 5.4.0 20160609] on linux
Type "copyright", "credits" or "license()" for more information.
>>> $nome = input|
```

Ln: 4 Col: 17



```
Python 3.5.2 Shell
```

File Edit Shell Debug Options Window Help

```
Python 3.5.2 (default, Sep 14 2017, 22:51:06)
[GCC 5.4.0 20160609] on linux
Type "copyright", "credits" or "license()" for more information.
>>> $nome = input("Digite o seu nome: ")
SyntaxError: invalid syntax
>>>
```

Ln: 6 Col: 4

Veremos logo adiante a razão desse erro!

Uso do IDLE

- O auxílio do IDLE em completar seu código é um recurso conhecido pelos programadores.
- Este recurso é comumente chamado de “*code completion*”.
- O code completion está presente na grande maioria dos IDEs das várias linguagens de programação
 - IDE: Integrated Development Environment.

Relembrando: Variáveis

Variáveis

- São formas de se armazenar dados para uso posterior dentro do código;
- Toda variável pode **receber** valores:

```
nome = "Bruno"  
idade = 26  
peso = 60.3
```

nome recebe "Bruno"
idade recebe 26
peso recebe 60.3

- Em Python, variáveis podem ser classificadas em três **tipos básicos**
 - Variável inteira: valores inteiros;
 - Variável de ponto flutuante: valores decimais;
 - Variável do tipo **string**: mensagens;

Valores possuem tipos

Variáveis manipulam valores

Variáveis possuem tipos

A função `type()`

- Digite no IDLE:

```
>>> type("Alô, Mundo!")  
<class 'str'> ← É um tipo de  
>>> type(17)  
<class 'int'> ← É um tipo de  
>>> type(3.2)  
<class 'float'> ← É um tipo de  
>>> type(True)  
<class 'bool'> ← É um tipo de
```

- A função **`type(argumento)`** retorna o **tipo** de um valor passado por parâmetro.

A função **type()**

- Digite no IDLE:

```
>>> type("Alô, Mundo!")  
<class 'str'>  
>>> mensagem = "Alô, Mundo!"  
>>> type(mensagem)  
<class 'str'>
```

- A função **type(argumento)** também pode ser utilizada para retornar o tipo de variáveis
 - Afinal, variáveis representam valores

O tipo string

- A mensagem “Alô, Mundo!” é uma string

```
>>> mensagem = "Alô, Mundo!"  
>>> type(mensagem)  
<class 'str'>
```

- Uma string é um conjunto de caracteres
 - Cada caractere possui uma posição dentro da string
 - A quantidade de caracteres representa o tamanho da string
- Por exemplo: palavra “Bruno”
 - Qual a posição do caractere “n”?
 - Qual o tamanho da string “Bruno”?

O tipo string

- Por exemplo: palavra “Bruno”

0	1	2	3	4	← Posições
B	r	u	n	o	← Conteúdo

- Qual a posição do caractere “n”?
- Qual o tamanho da string “Bruno”?
- Exemplo: “Hoje é um belo dia!”

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
H	o	j	e		é		u	m		b	e	l	o		d	i	a	!

- Qual a posição do primeiro espaço da string?
- Qual a posição do último “i” da string?
- Qual o tamanho da string?

Tipos Numéricos

- Tipo **int** e **float**:

```
>>> type(17)
<class 'int'>
>>> type(3.2)
<class 'float'>
```

- **inteiros** pertencem a um tipo chamado **int**;
- Números decimais pertencem a um tipo chamado **float**.
- Pode-se utilizar operadores aritméticos com os tipos numéricos:
 - + - * /

O tipo **float**

- Tipo **float** é o tipo decimal de Python

```
>>> type(3.2)  
<class 'float'>
```

- Números decimais são do tipo **float**
 - Esses números são chamados de **ponto flutuante**;
- Utiliza-se **ponto no lugar da vírgula** para separar a parte inteira da decimal;

Tipo Booleano

- O tipo `bool` é o tipo lógico de Python

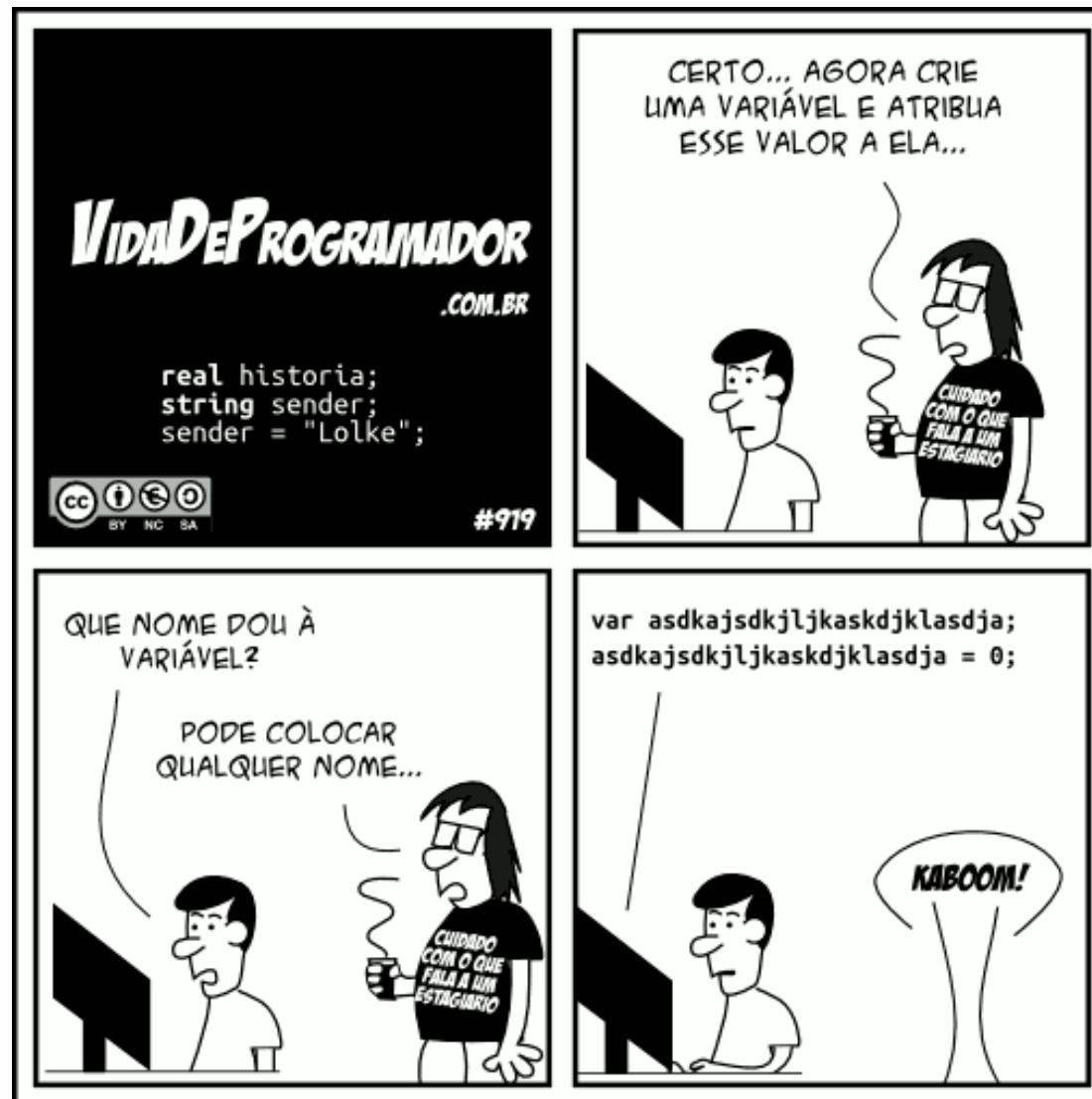
```
➤➤➤ type(True)  
<class 'bool'>
```

- Valores lógicos têm um tipo específico: **bool**;
 - Tipo booleano: Lógica de Boole;
- Possíveis valores em Python: **True** e **False**;

Nomes de Variáveis: regras gerais

Nomes de Variáveis

- Escolha um nome intuitivo!



Nomes de Variáveis

- Nomes de variáveis podem ter letras e números
 - Contanto que o nome comece com letras minúsculas → boa prática de programação;
- Não podem ser utilizados nomes de palavras reservadas de Python:

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	break
except	in	raise		

Nomes de Variáveis

- Lembre-se: Python é case sensitive!
- O caractere sublinhado `_` pode aparecer em um nome de variável
 - É muito utilizado quando o nome da variável contém múltiplas palavras, e.g.:
 - `nome_do_usuario`
 - `nome_do_arquivo`
 - `venda_de_produto`

Prática 1

- Por que as variáveis a seguir não podem ter os nomes que receberam?

```
>>> 7nome = "Bruno"  
SyntaxError: invalid syntax  
>>> valore$ = 1000000  
SyntaxError: invalid syntax  
>>> class = "Fundamentos de Lógica e Algoritmos"  
SyntaxError: invalid syntax
```


Prática 2

- Qual a razão do erro apresentado no slide 6?

```
*Python 3.5.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.5.2 (default, Sep 14 2017, 22:51:06)
[GCC 5.4.0 20160609] on linux
Type "copyright", "credits" or "license()" for more information.
>>> $nome = input|
```

Ln: 4 Col: 17

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (default, Sep 14 2017, 22:51:06)
[GCC 5.4.0 20160609] on linux
Type "copyright", "credits" or "license()" for more information.
>>> $nome = input("Digite o seu nome: ")
SyntaxError: invalid syntax
>>>
```

Ln: 6 Col: 4

Prática 3

- Quando se deseja imprimir mais do que um valor de variável, pode-se utilizar, na função print, vírgulas separando essas variáveis:

```
>>> estado = input("Qual é o seu estado? ")
Qual é o seu estado? Paraíba
>>> cidade = input("Qual é a sua cidade? ")
Qual é a sua cidade? João Pessoa
>>> bairro = input("Qual é o seu bairro? ")
Qual é o seu bairro? Jardim Oceania
>>> print(estado, cidade, bairro)
Paraíba João Pessoa Jardim Oceania
```

Prática 4

- Considerando a Prática 2 e o resultado da impressão da mesma:

```
>>> print(estado, cidade, bairro)  
Bruno João Pessoa Jardim Oceania
```

- Faça com que a impressão das três informações seja separada por um hífen, conforme o exemplo a seguir:

```
>>> SEU COMANDO AQUI  
Paraíba - João Pessoa - Jardim Oceania
```

- Dica: pode-se separar variáveis e valores por vírgulas na função print

Prática 5

- Reimplemente um sisteminha que seja capaz de imprimir o estado, cidade o bairro do usuário
 - O usuário deve informar esses dados.

```
>>> estado = input("Qual é o seu estado? ")
Qual é o seu estado? Paraíba
>>> cidade = input("Qual é a sua cidade? ")
Qual é a sua cidade? João Pessoa
>>> bairro = input("Qual é o seu bairro? ")
Qual é o seu bairro? Jardim Oceania
>>> print(estado, cidade, bairro)
Paraíba João Pessoa Jardim Oceania
```

Prática 6

- Implemente um sisteminha que seja capaz de imprimir o estado, cidade o bairro de um parente próximo do usuário:
 - O usuário deve informar esses dados.

```
>>> estado = input("Qual é o seu estado? ")
Qual é o seu estado? Paraíba
>>> cidade = input("Qual é a sua cidade? ")
Qual é a sua cidade? João Pessoa
>>> bairro = input("Qual é o seu bairro? ")
Qual é o seu bairro? Jardim Oceania
>>> print(estado, cidade, bairro)
Paraíba João Pessoa Jardim Oceania
```

Prática 7

- Implemente um sisteminha que seja capaz de imprimir o estado, cidade o bairro de um amigo próximo do usuário:
 - O usuário deve informar esses dados.

```
>>> estado = input("Qual é o seu estado? ")
Qual é o seu estado? Paraíba
>>> cidade = input("Qual é a sua cidade? ")
Qual é a sua cidade? João Pessoa
>>> bairro = input("Qual é o seu bairro? ")
Qual é o seu bairro? Jardim Oceania
>>> print(estado, cidade, bairro)
Paraíba João Pessoa Jardim Oceania
```

Pareceu ser cansativo reescrever os comandos?



Sim! Para situações como essas é que precisamos criar um arquivo py!



Um arquivo py nada mais é do que
um **script** de comandos Python

Ou seja: um programa em Python!



Utilize o interpretador apenas para testar comandos



Para programar: crie arquivos py!!!

