

Curso básico de Python para estudantes de Física

Germán A. Racca

Universidade do Estado do Rio Grande do Norte
Faculdade de Ciências Exatas e Naturais
Departamento de Física
Mossoró - RN

25 de Maio de 2016

- 1 Introdução
- 2 Instalação de pacotes básicos
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos
- 5 Recipientes
 - Listas
 - Strings
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso

- 1 **Introdução**
- 2 Instalação de pacotes básicos
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos
- 5 Recipientes
 - Listas
 - Strings
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso



- Linguagem de programação criada por Guido van Rossum em 1991.
- Modelo de desenvolvimento comunitário gerenciado pela organização sem fins lucrativos *Python Software Foundation*.
- Possui licença *Python Software Foundation License*, compatível com GPL mas não é *copyleft*.
- Linguagem de alto nível, interpretada, interativa, orientada a objetos e de tipagem dinâmica.
- Multiplataforma: Linux 😊, FreeBSD, Solaris, MacOS X, Windows, etc.

- 1 Introdução
- 2 **Instalação de pacotes básicos**
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos
- 5 Recipientes
 - Listas
 - Strings
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso

Instalação de pacotes básicos

Fedora 😊

```
$ sudo dnf install python-ipython numpy scipy python2-matplotlib \  
python-astropy
```

Ubuntu

```
$ sudo apt-get install ipython python-numpy python-scipy \  
python-matplotlib python-astropy
```

- IPython → console interativo
- NumPy → operações com vetores e matrizes
- SciPy → computação científica (interpolação, integração, otimização, estatística, etc)
- Matplotlib → gráficos 2D com qualidade de publicação
- Astropy → pacotes para Astronomia (coordenadas, arquivos FITS, fotometria, espectroscopia, cosmologia, etc)

- 1 Introdução
- 2 Instalação de pacotes básicos
- 3 Python interativo**
- 4 Tipos básicos e Operadores básicos
- 5 Recipientes
 - Listas
 - Strings
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso

Python interativo

- Console interativo do Python

```
german@skytux:~  
german@skytux ~ $ python  
Python 2.7.11 (default, Mar 31 2016, 20:46:51)  
[GCC 5.3.1 20151207 (Red Hat 5.3.1-2)] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> █
```

- Console interativo do IPython

```
IPython: home/german  
german@skytux ~ $ ipython  
Python 2.7.11 (default, Mar 31 2016, 20:46:51)  
Type "copyright", "credits" or "license" for more information.  
  
IPython 3.2.1 -- An enhanced Interactive Python.  
?          -> Introduction and overview of IPython's features.  
%quickref  -> Quick reference.  
help       -> Python's own help system.  
object?    -> Details about 'object', use 'object??' for extra details.  
  
In [1]: █
```


- 1 Introdução
- 2 Instalação de pacotes básicos
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos**
- 5 Recipientes
 - Listas
 - Strings
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso

- Inteiros

```
>>> 1 + 2
3
>>> a = 4
>>> type(a)
<type 'int'>
```

- Floats

```
>>> c = 2.1
>>> type(c)
<type 'float'>
```

- Conversão de tipos

```
>>> float(1)
1.0
```

- Complexos

```
>>> a = 1.5 + 0.5j
>>> a.real
1.5
>>> a.imag
0.5
>>> type(a)
<type 'complex'>
```

- Booleanos

```
>>> 3 > 4
False
>>> test = (3 > 4)
>>> test
False
>>> type(test)
<type 'bool'>
```

```
>>> int(4.3)
4
```

```
>>> str(2)
'2'
```

- Soma e subtração

```
>>> 2.5 + 4
6.5
>>> -2 + 4
2
>>> 3 + -7
-4
>>> 4 - -4
8
```

- Multiplicação

```
>>> 2 * 1.5
3.0
>>> 2 * -1
-2
>>> -4 * 2
-8
```

- Divisão

```
>>> 5 / 2
2
>>> 5 / 2.0 # ou 5.0/2, ou 5.0/2.0
2.5
>>> -5 / 2
-3
>>> -5.0 / 2 # ou -5/2.0, ou -5.0/2.0
-2.5
```

- Potenciação

```
>>> 2**3
8
>>> (-2)**4 # -2**4 = -16
16
>>> pow(2, 3)
8
>>> pow(-2, 4)
16
```

- 1 Introdução
- 2 Instalação de pacotes básicos
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos
- 5 Recipientes**
 - Listas
 - Strings
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso

- 1 Introdução
- 2 Instalação de pacotes básicos
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos
- 5 Recipientes**
 - **Listas**
 - Strings
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso

Listas

- Coleção ordenada e mutável de objetos de diversos tipos

```
>>> numeros = [1, 2, 3, 4]
```

```
>>> print numeros
```

```
[1, 2, 3, 4]
```

```
>>> opcoes = ['sim', 'nao', 'talvez']
```

```
>>> print opcoes
```

```
['sim', 'nao', 'talvez']
```

```
>>> l = [numeros, opcoes]
```

```
>>> print l
```

```
[[1, 2, 3, 4], ['sim', 'nao', 'talvez']]
```

```
>>> L = ['red', 'blue', 1, 5.7, 3+2.1j]
```

```
>>> print L
```

```
['red', 'blue', 1, 5.7, (3+2.1j)]
```

```
>>> type(L)
```

```
<type 'list'>
```

Slicing

Mecanismo para criar 'fatias', ou *slices*, de uma lista. Um *slice* é uma lista gerada a partir de um fragmento de outra lista.

```
>>> linux = ['fedora', 'debian', 'opensuse', 'ubuntu', 'arch']
>>> print linux[0]
fedora
```

```
>>> print linux[3]
ubuntu
```

```
>>> print linux[5]
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

IndexError: list index out of range

```
>>> linux[-1]
'arch'
>>> linux[-3]
'opensuse'
```

```
>>> linux[1] = 'mint'
>>> print linux
['fedora', 'mint', 'opensuse', 'ubuntu', 'arch']
```

Listas

```
>>> linux = ['fedora', 'debian', 'opensuse', 'ubuntu', 'arch']
>>> linux[0:2]
['fedora', 'debian']
```

```
+-----+-----+-----+-----+-----+
| fedora | debian | opensuse | ubuntu | arch |
+-----+-----+-----+-----+-----+
0         1         2         3         4         5
-5        -4        -3        -2        -1
```

```
>>> linux[:2]
['fedora', 'debian']
>>> linux[2:4]
['opensuse', 'ubuntu']
>>> linux[2:]
['opensuse', 'ubuntu', 'arch']
>>> linux[1:-1]
['debian', 'opensuse', 'ubuntu']
```


- 1 Introdução
- 2 Instalação de pacotes básicos
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos
- 5 **Recipientes**
 - Listas
 - **Strings**
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso

Strings

- Seqüência imutável que armazena cadeias de caracteres

```
>>> print 'Corinthians, Palmeiras ou Sao Paulo...'
Corinthians, Palmeiras ou Sao Paulo...
```

```
>>> print 'vai 'curintia''
File "<stdin>", line 1
    print 'vai 'curintia''
           ^
```

SyntaxError: invalid syntax

```
>>> print 'vai \'curintia\''
vai 'curintia'
```

```
>>> print 'times:\n\tCorinthians\n\tPalmeiras\n\tSao Paulo\n\n...vai \'curintia\''
```

```
times:
    Corinthians
    Palmeiras
    Sao Paulo
```

```
...vai 'curintia'
```

Strings

```
>>> print "vai 'curintia'"  
vai 'curintia'
```

```
>>> print "vai "curintia"  
File "<stdin>", line 1  
    print "vai "curintia"  
            ^
```

SyntaxError: invalid syntax

```
>>> print "vai \"curintia\""  
vai "curintia"
```

```
>>> print """times:  
...     Corinthians  
...     Palmeiras  
...     Sao Paulo  
...  
... ..vai "curintia"  
... """
```

```
times:  
    Corinthians  
    Palmeiras  
    Sao Paulo  
  
...vai "curintia"
```

- 1 Introdução
- 2 Instalação de pacotes básicos
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos
- 5 Recipientes**
 - Listas
 - Strings
 - Dicionários**
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso

- Coleção de elementos identificados por uma chave e um valor (*key/value*)
 - Chave (ou índice) → qualquer tipo imutável
 - Valor → qualquer coisa

```
batman = {'nome': 'Bruce Wayne', 'filmes': 9, 'amigos': ['Alfred', 'Robin']}
```

```
>>> print batman['nome']
```

```
Bruce Wayne
```

```
>>> print batman['filmes']
```

```
9
```

```
>>> print batman['amigos']
```

```
['Alfred', 'Robin']
```

```
>>> batman['inicio'] = 1939
```

```
>>> print batman
```

```
{'amigos': ['Alfred', 'Robin'], 'inicio': 1939, 'filmes': 9, 'nome': 'Bruce Wayne'}
```

- 1 Introdução
- 2 Instalação de pacotes básicos
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos
- 5 Recipientes
 - Listas
 - Strings
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso

- 1 Introdução
- 2 Instalação de pacotes básicos
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos
- 5 Recipientes
 - Listas
 - Strings
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso

if/elif/else

- Estruturas que controlam a ordem na qual o código é executado
- Condições → análises que retornam valores booleanos: Verdadeiro (*True*) ou Falso (*False*)
- Operadores: == (igual), > (maior), < (menor), >= (maior ou igual), <= (menor ou igual), != (diferente)

```
numero = int(raw_input("Por favor, insira um numero:"))
if numero > 0:
    print "Voce digitou um numero positivo."
elif numero < 0:
    print "Voce digitou um numero negativo."
else:
    print "Voce digitou zero."
```

```
>>> x = 1
>>> if x > 0:
...     print 'positivo'
...
positivo
```

```
>>> if x > 0:
...     print 'positivo'
... else:
...     print 'negativo'
...
positivo
```


- 1 Introdução
- 2 Instalação de pacotes básicos
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos
- 5 Recipientes
 - Listas
 - Strings
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 Material do curso

for/range

for...in

```
for VARIAVEL in ITERAVEL:  
    # BLOCO DE CODIGO
```

```
>>> meses = ['janeiro', 'fevereiro', 'marco']  
>>> for mes in meses:  
...     print mes  
...  
janeiro  
fevereiro  
marco
```

range

```
range(inicio, fim, passo)  
range(i, j) retorna [i, i+1, i+2, ..., j-1]
```

```
>>> for i in range(4):  
...     print i  
...  
0  
1  
2  
3
```

- 1 Introdução
- 2 Instalação de pacotes básicos
- 3 Python interativo
- 4 Tipos básicos e Operadores básicos
- 5 Recipientes
 - Listas
 - Strings
 - Dicionários
- 6 Controle de fluxo
 - if/elif/else
 - for/range
- 7 **Material do curso**

Aulas do Lázaro Camargo (INPE):

<https://scbi9y.s.cld.pt>

material-python-00.tar.bz2

```
python_basico.pdf
python_aula_01_introducao.pdf
python_aula_02_tipos.pdf
python_aula_03_listas.pdf
python_aula_04_estruturas_controle.pdf
```