



86ª EDIÇÃO

**SEQ
UFRJ**

26 a 30 de agosto



Introdução à programação para ciência e engenharia em *Python*

Iuri Segtovich

Parte 2: Lógica e Sintaxe

Escrita e leitura de arquivo (`open/close`, `write`, `readline`)



spyder

 3.0.2

Scientific PYthon Development
EnviRonment. Powerful Python IDE with
advanced editing, interactive testing,
debugging and introspection features

Launch

Working directory (%pwd)

/home/segtoivichisv/Desktop/curso-python-chemeng/exemplos/spyder_projeto_1

IPython console



Console 1/A ✕

Type "copyright", "credits" or "license" for more information.

IPython 6.4.0 -- An enhanced Interactive Python.

In [1]:

In [1]: %pwd

Out[1]: '/home/segtoivichisv'

In [2]: %pwd

Out[2]: '/home/segtoivichisv/Desktop/curso-python-chemeng/tests'

Abrir arquivo (open, write, close)

```
9 f = open('arquivo.txt', 'w')
10 f.write('FULANO esteve aqui' + '\n')
11 f.write('segunda linha')
12 f.close()
```

Formatação (format)

```
FMTsci = '{:.2e}\n' #científica
FMTdec = '{:.2f}\n' #decimal
print(FMTsci.format(6.02e23))
print(FMTdec.format(3.1415))
```

- Definir um string de formatação: Servirá de base para impressão.
- As chaves determinam onde os valores formatados entrarão
- Antes dos dois pontos pode vir um número indicando qual de vários valores entra na lacuna em análise.
- Entre os dois pontos e o ponto pode vir um tamanho mínimo da lacuna, a ser preenchido por espaços em branco quando o valor formatado não for ocupar tudo.
- Depois do ponto vem a precisão para formatar o seu float.
- Fechamos o string de formatação com \n se quisermos que haja quebra de linha após imprimir esse string com os valores formatados.

Alguns exemplos

```
f.write('{:s}\n'.format("português brasil cachaça açúcar!")) #imprimindo strings
f.write('{:d}\n'.format(1)) #imprimindo inteiros
f.write('{:.2e}\n'.format(1.963865200092e-3)) #imprimindo floats
f.write('{:.2f}\n'.format(1.963865200092e-3))
f.write('{:.2e}\n'.format(1.963865200092e4))
f.write('{:.2f}\n'.format(1.963865200092e4))
#imprimindo dois items simultaneamente
f.write(" primeiro {:s} depois {:s}\n".format("os primeiros","os últimos"))
f.write(" primeiro {1:s} depois {0:s}\n".format("os primeiros","os últimos"))
```

Formatações especiais e tabela de formatação completa

```
'{:Y-%m-%d %H:%M}'.format(datetime(2001, 2, 3, 4, 5))
```

```
2 0 0 1 - 0 2 - 0 3 0 4 : 0 5
```

<https://pyformat.info/>

```
replacement_field ::= "{" [field_name] ["!" conversion] [":" format_spec] "}"
field_name         ::= arg_name ( "." attribute_name | "[" element_index "]" ) *
arg_name           ::= [identifier | digit+]
attribute_name     ::= identifier
element_index      ::= digit+ | index_string
index_string       ::= <any source character except "]"> +
conversion         ::= "r" | "s" | "a"
format_spec        ::= <described in the next section>
```

```
format_spec ::= [[fill]align][sign][#][0][width][grouping_option][.precision][type]
fill        ::= <any character>
align       ::= "<" | ">" | "=" | "^"
sign        ::= "+" | "-" | " "
width       ::= digit+
grouping_option ::= "_" | ","
precision   ::= digit+
type        ::= "b" | "c" | "d" | "e" | "E" | "f" | "F" | "g" | "G" | "n" | "o" | "s" | "x" | "X" | "%"
```

<https://docs.python.org/3/library/string.html#string-formatting>

Leitura (readline)

```
f = open("input.txt", 'r')
varname=f.readline()
print("alimentando", varname, "em x")
x=[]
linha=f.readline()
lista=linha.split(",")
for xi in lista:
    print("convertendo", xi)
    x.append(float(xi))

print("x: ",x)
n=len(x)
print("n: ",n)
```


module

- "Leitura1.py"
- "Programa1.py"

```
1 import Leitura1
```

Referências principais

[https://www.tutorialspoint.com/
python3/
python_basic_syntax.htm](https://www.tutorialspoint.com/python3/python_basic_syntax.htm)

[https://stackoverflow.com/
search](https://stackoverflow.com/search)

perguntas

