

85ª EDIÇÃO

**SEQ UFRJ**

20 a 24 de agosto



# Introdução à programação para ciência e engenharia em *Python*

*Iuri Soter Viana Segtovich*

Parte 2: Lógica e Sintaxe

Escrita e leitura de arquivos (with open, 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/segtoovichisv/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/segtoovichisv'

In [2]: %pwd

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

# Abrir arquivo (with open, write)

```
Escrita.py ✕ arquivo.txt ✕  
1 #português brasil cachaça açúcar  
2 with open("arquivo.txt", 'w') as f:  
3     print("abrindo...")  
4     f.write("uma frase qualquer na primeira linha\n")  
5     f.write("o \n com uma barra quebra linha\n")  
6     f.write("com duas a gente imprime a própria barra\n")  
7  
8     print(f.closed)  
9 print(f.closed)
```

```
In [8]: runfile('/home/segrovichisv/  
spyder_projeto_1/Escrita.py', wdir='  
chemeng/exemplos/spyder_projeto_1')  
abrindo...  
False  
True
```

```
Escrita.py ✕ arquivo.txt ✕  
1 uma frase qualquer na primeira linha  
2 o \n com uma barra quebra linha  
3 com duas a gente imprime a própria barra  
4
```

# 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)

```
with open("input.txt", 'r') as f:
    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)
```

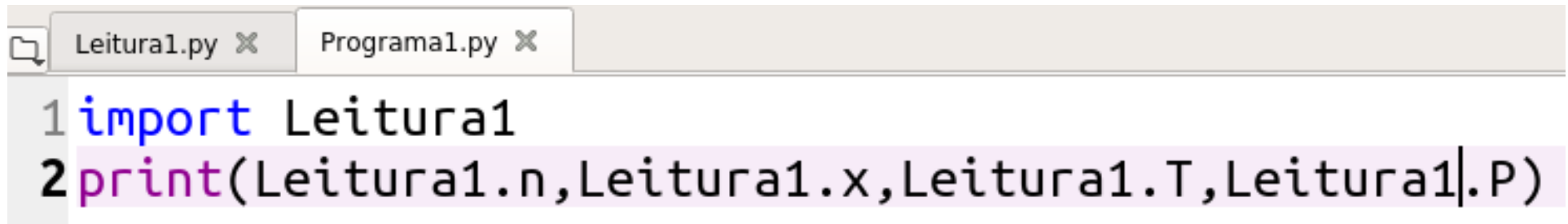


# Leitura (csv.reader)

```
with open("data.txt", 'r') as f:
    import csv
    data = csv.reader(f)
    next(data) # skip header
    T=[]
    P=[]
    for row in data:
        T.append(float(row[0]))
        P.append(float(row[1]))
print("T: ", T)
print("P: ", P)
```

# module

- "Leitura1.py"
- new console / restart kernel / %reset -f
- "Programa1.py"



```
1 import Leitura1
2 print(Leitura1.n, Leitura1.x, Leitura1.T, Leitura1.P)
```

# Leitura.fazer\_leitura()

```
Leitura.py ✕
1 #português brasil cachaça açúcar
2 '''
3 esse módulo faz a leitura'''
4
5 def fazer_leitura():
6     with open("input.txt", 'r') as f:|
7         varname=f.readline()
8     #     print("alimentando", varname,
9         x=[]
10        linha=f.readline()
11        lista=linha.split(",")
12        for xi in lista:
13            #         print("convertendo", xi)
14            x.append(float(xi))
15    #     print("x: ",x)
```

```
1 import Leitura
2 print(dir(Leitura))
3 print(help(Leitura))
4 n,x,T,P = Leitura.fazer_leitura()
5 print(n,x,T,P)
```

In [22]: Leitura?

Type: module  
String form: <module 'Leitura' from '/home/segtoovichisv/Desktop/curso-python-chemeng/exemplos/spyder\_projeto\_1/Leitura.py'>  
File: ~/Desktop/curso-python-chemeng/exemplos/spyder\_projeto\_1/Leitura.py  
Docstring: esse módulo faz a leitura

In [23]: Leitura??

# 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

