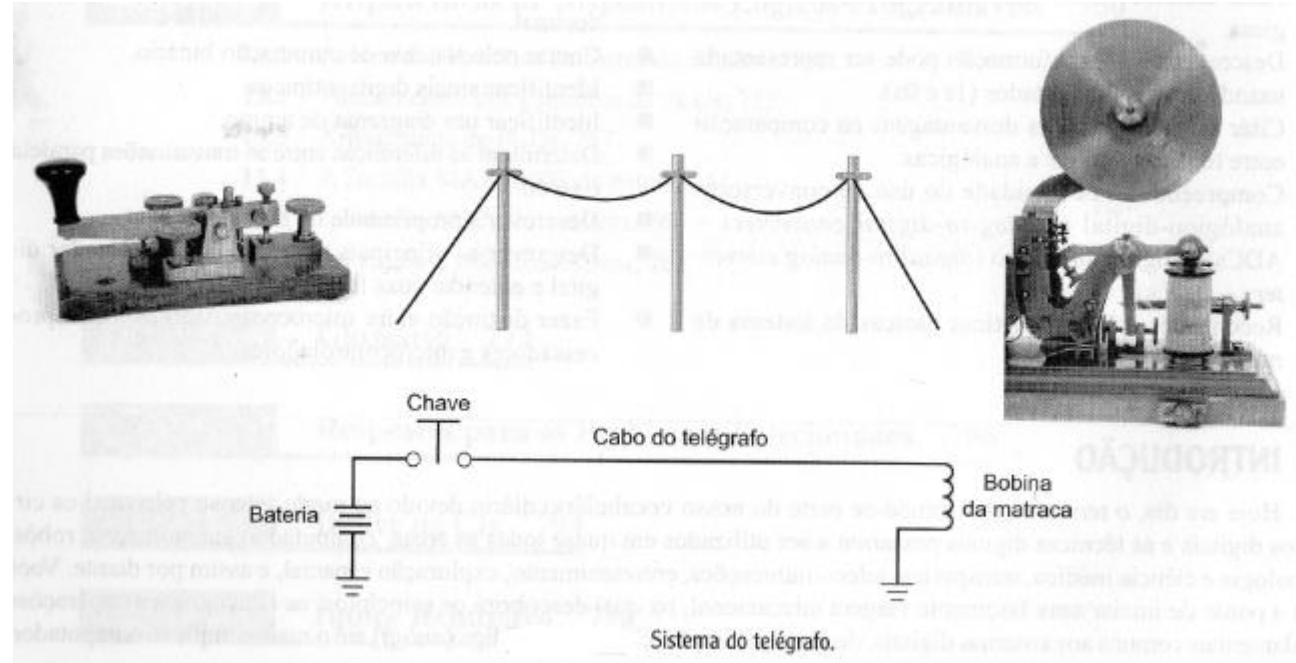


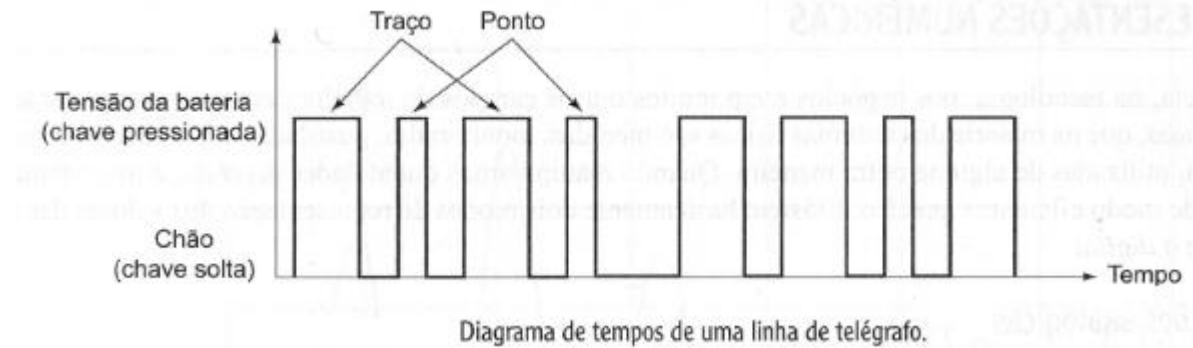


*Algoritmos e Introdução à Computação*  
→ *Ambiente de programação com Python*  
→ *Tipos e declaração de variáveis*

# Sinais digitais e analógicos



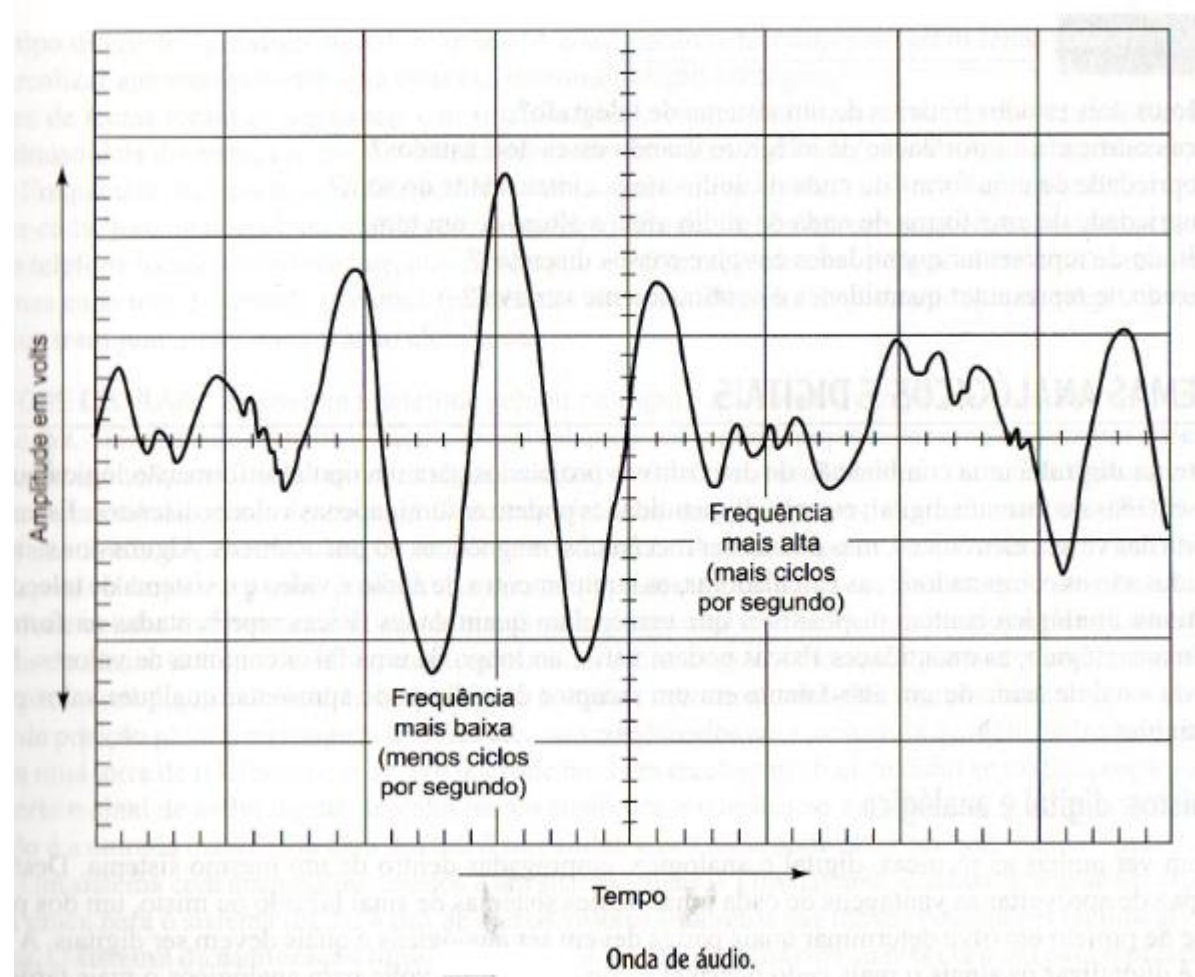
sinal digital





# Sinais digitais e analógicos

sinal analógico: onda de áudio



# Sistemas de numeração

Sistema decimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Sistema binário: 0, 1.

Sistema hexadecimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,  
A, B, C, D, E, F.

# Sistemas de numeração

Exemplo: converter o número  $1011_{(b2)}$  para decimal.

MSD = *most significant digit*

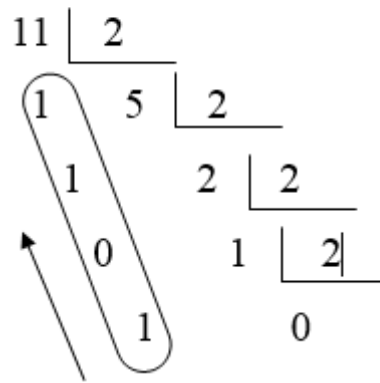
LSD = *least significant digit*

$$\begin{aligned} 1011_2 &= (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\ &= 8 + 0 + 2 + 1 \\ &= 11_{10} \end{aligned}$$

# Sistemas de numeração

Exemplo: converter o número  $11_{(b10)}$  para binário.

método das divisões sucessivas por 2

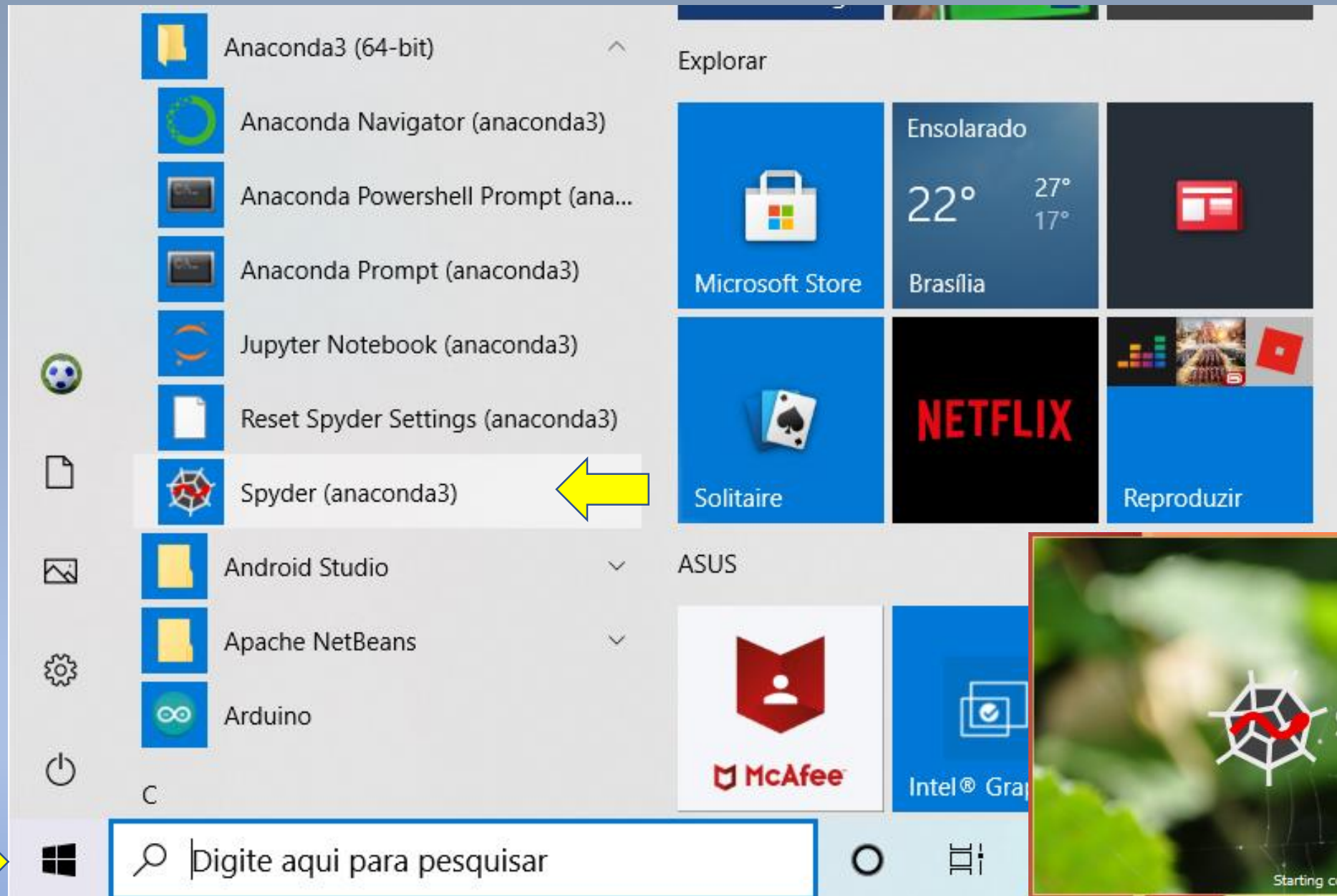


$$11_{(b10)} = 1011_{(b2)}$$

# Sistemas de numeração

decimal	binário	hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

# Acesso ao ambiente de programação *Spyder*





```

1  #-*- coding: utf-8 -*-
2  """
3  Created on Wed May 26 11:13:52 2021
4
5  @author: cmari
6  """
7
8

```

# Ambiente de programação *Spyder*

View Help

Panels

- ☐ Lock panes and toolbars Ctrl+Shift+F5
- ☒ Close current pane Ctrl+Shift+F4
- ☒ Maximize current pane Ctrl+Alt+Shift+M

Toolbars

- ☐ Hide toolbars Alt+Shift+T
- ☐ Window layouts
- ☐ Use previous layout Alt+Shift+PgUp
- ☐ Use next layout Alt+Shift+PgDown

- ☒ Fullscreen mode F11
- ☐ Attached console window (debugging)

- ☒ Editor
- ☒ IPython console Ctrl+Shift+I
- ☒ Variable explorer Ctrl+Shift+V
- ☒ Help Ctrl+Shift+H
- ☐ Plots Ctrl+Shift+G
- ☐ Files Ctrl+Shift+X
- ☐ Outline Ctrl+Shift+O
- ☐ Project Ctrl+Shift+P
- ☐ Find Ctrl+Shift+F
- ☐ History Ctrl+Shift+L
- ☐ Profiler
- ☐ Breakpoints Ctrl+Shift+B
- ☐ Code Analysis
- ☐ Online help Ctrl+Shift+D
- ☐ Internal console

O código *Python* pode ser escrito por editores de texto que usem codificações de caracteres ASCII ou o unicode UTF-8, um subconjunto do ASCII que pode representar um caracter em qualquer linguagem. Os arquivos *Python* possuem extensão “.py”.

o tipo inteiro - int

Variable explorer Help

LSP Python: ready

Kite: ready (no index)

conda: base (Python 3.8.3)

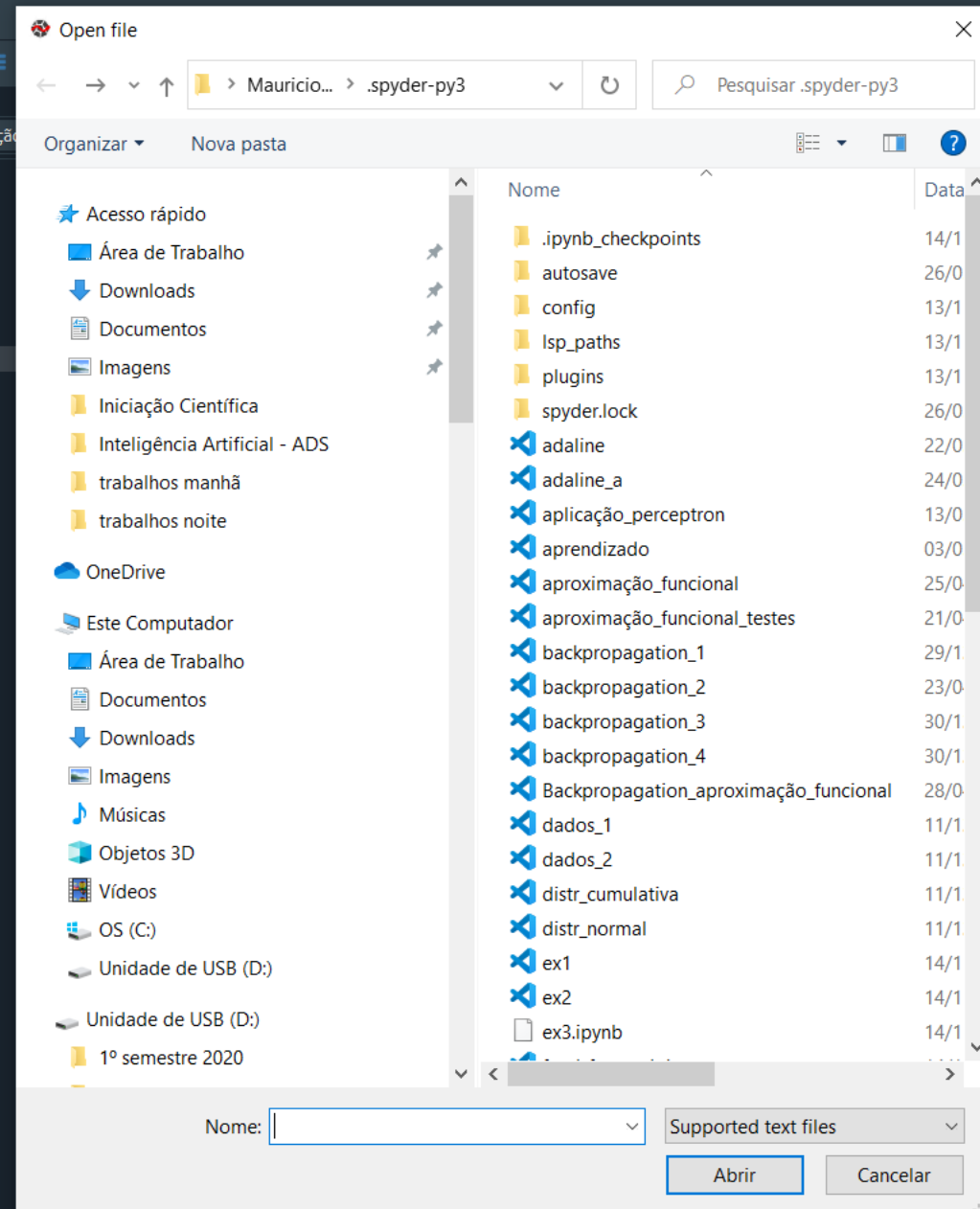
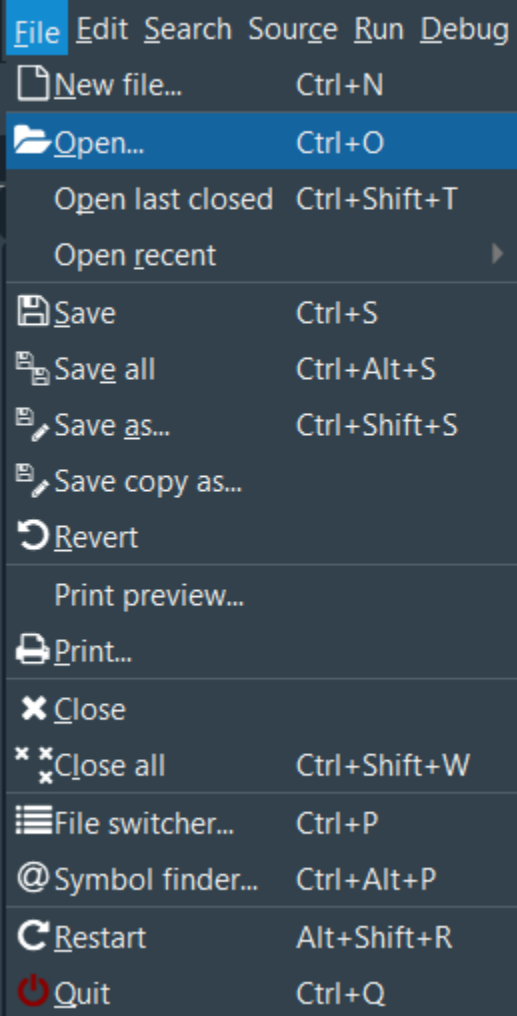
Line 8, Col 1

UTF-8

CRLF

RW

Mem 77%



### Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

[New to Spyder? Read our tutorial](#)

diretório onde são  
gravados os arquivos  
*Python* pelo *Spyder*:  
“spyder-py3”

# Declaração de variável:

→ Quando se declara uma variável, equivale a reservar um espaço de memória (com uma identificação  $\equiv$  endereço) no computador.

## Exemplo:

Declaração de uma variável numérica do tipo inteiro **x**, e atribuição de valor **-10**:

**inteiro** **x** = **-10**

Haverá um espaço de memória para guardar esta variável.

- endereço de memória



# Tipos de dados int e string

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu May 27 08:57:57 2021
4
5  @author: cmari
6  """
7
8
9  # Comentários em uma linha podem ser precedidos por "jogo da velha #"
10 """
11 Comentários em mais de uma linha devem
12 estar entre aspas triplas
13 """
14
15 #atribuições de valores a variáveis do tipo inteiro - int
16 a = 12
17 b = -8
18 #atribuições de valores a variáveis do tipo string - str
19 c = "Ciência de Dados"
20 d = '@*$? '
21 e = ""
22 f = "88"
23
24 #comando de impressão no console: print
25 print ("a = ",a)
26 print ("b = ",b)
27 print ("c = ",c)
28 print ("d = ",d)
29 print ("e = ",e)
30 print ("f = ",f)
31
```

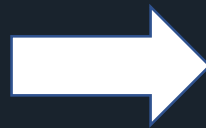


```
a = 12
b = -8
c = Ciência de Dados
d = @*$?
e = 
f = 88
```



# Conversão entre tipos de dados

```
31
32 #conversão de string para inteiro
33 h = int(f)
34 g = b + h
35 print ("g = (-8) + 88 = ", g)
36
37 #conversão de inteiro para string
38 str(a)
39 a = c
40 print ("a: string = ", a)
41
```



```
g = (-8) + 88 = 80
a: string = Ciência de Dados
```

```
15 #atribuições de valores a variáveis do tipo inteiro - int
16 a = 12
17 b = -8
18 #atribuições de valores a variáveis do tipo string - str
19 c = "Ciência de Dados"
20 d = '@*$? '
21 e = ""
22 f = "88"
```

Python 3.8

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\cmari\.spyder-py3

pagação\_aproximação\_funcional.py × aproximação\_funcional.py × padrão\_caracteres\_numéricos.py × padrão\_caracteres.py × ex1.py × ex2.py × dados\_1.py × dados\_2.py × uso\_variáveis.py ×

```
1  -*- coding: utf-8 -*-
2  """
3  Created on Thu May 27 08:57:57 2021
4
5  @author: cmari
6  """
7
8
9  # Comentários em uma linha podem ser precedidos por "jogo da velha #"
10 """
11 Comentários em mais de uma linha devem
12 estar entre aspas triplas
13 """
14
15 #atribuições de valores a variáveis do tipo inteiro - int
16 a = 12
17 b = -8
18 #atribuições de valores a variáveis do tipo string - str
19 c = "Ciência de Dados"
20 d = '@*${?}'
21 e = ""
22 f = "88"
23
24 #comando de impressão no console: print
25 print ("a = ",a)
26 print ("b = ",b)
27 print ("c = ",c)
28 print ("d = ",d)
29 print ("e = ",e)
30 print ("f = ",f)
31
32 #conversão de string para inteiro
33 h = int(f)
34 g = b + h
35 print ("g = (-8) + 88 = ", g)
36
37 #conversão de inteiro para string
38 str(a)
39 a = c
40 print ("a: string = ", a)
41
```

Run Debug Consoles Projects Tools View Help

Run F5

Run cell Ctrl+Return

Run cell and advance Shift+Return

Re-run last cell Alt+Return

Run selection or current line F9

Re-run last script F6

Configuration per file... Ctrl+F6

Profile F10

Nam	Type	Size	Value
a	str	1	Ciência de Dados
b	int	1	-8
c	str	1	Ciência de Dados
d	str	1	@*\${?}
e	str	1	
f	str	1	88
g	int	1	80
h	int	1	88

Variable explorer Help

Console 1/A

In [14]: runfile('C:/Users/cmari/.spyder-py3/uso\_variáveis.py', wdir='C:/Users/cmari/.spyder-py3')

a = 12

b = -8

c = Ciência de Dados

d = @\*\${?}

e =

f = 88

g = (-8) + 88 = 80

a: string = Ciência de Dados

In [15]:

(Summerfield, 2013)

LSP Python: ready Kite: ready conda: base (Python 3.8.3) Line 41, Col 1 UTF-8 CRLF RW Mem 66%

09:39 27/05/2021

Tipos de dados int e string: Código completo

***Exercício 1:*** Escrever código *Python* para atribuir a uma variável “*funcionário*” o nome do funcionário de uma empresa; à variável “*idade*”, sua idade; à variável “*endereço*”, o endereço do funcionário; à variável “*função*”, o cargo do funcionário e à variável “*salário*”, o valor do seu salário. Utilizar o comando *print* para mostrar os resultados no console.

# Referência de objeto: tipagem dinâmica

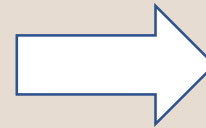
```
# Uma referência de objeto (=variável) assume o valor de um tipo de dado através do operador " = "  
cor = "azul" #cor <- string azul  
numero = 6 #numero <- número 6  
print("a cor é: ", cor)  
print("o número é: ", numero)
```

```
# o valor do objeto (=variável) pode ser mutável, independente do tipo  
# Python utiliza tipagem dinâmica
```

```
cor = numero  
print("o número é: ", cor)
```

```
#redefinição dos valores  
cor = "laranja"  
numero = 19
```

```
#imprimindo os tipos de dados das variáveis - "\n" -> muda de linha  
print("a cor é: ", cor, "\ntipo de dado: ", type(cor))  
print("o número é: ", numero, "\ntipo de dado: ", type(numero))
```



```
a cor é: azul  
o número é: 6  
o número é: 6  
a cor é: laranja  
tipo de dado: <class 'str'>  
o número é: 19  
tipo de dado: <class 'int'>
```

(Summerfield, 2013)



# Referência de objeto: tipagem dinâmica

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Fri May 28 15:51:11 2021
4
5  @author: cmari
6  """
7
8  # Uma referência de objeto (=variável) assume o valor de um tipo de dado através do operador " = "
9  cor = "azul" #cor <- string azul
10 numero = 6 #numero <- número 6
11 print("a cor é: ", cor)
12 print("o número é: ", numero)
13
14 # o valor do objeto (=variável) pode ser mutável, independente do tipo
15 # Python utiliza tipagem dinâmica
16
17 cor = numero
18 print("o número é: ", cor)
19
20 #redefinição dos valores
21 cor = "laranja"
22 numero = 19
23
24 #imprimindo os tipos de dados das variáveis - "\n" -> muda de linha
25 print("a cor é: ", cor, "\ntipo de dado: ", type(cor))
26 print("o número é: ", numero, "\ntipo de dado: ", type(numero))
```

(Summerfield, 2013)

```
Console 1/A X
In [2]: runfile('C:/Users/cmari/.spyder-py3/
referência_objeto.py', wdir='C:/Users/cmari/.spyder-
py3')
a cor é: azul
o número é: 6
o número é: 6
a cor é: laranja
tipo de dado: <class 'str'>
o número é: 19
tipo de dado: <class 'int'>
```

*Exercício 2:* Utilizar o comando *print* para mostrar no console os *tipos* de dados das variáveis utilizadas no exercício 1.

# Referências Bibliográficas

- *Programação em Python 3*

Uma Introdução Completa à Linguagem *Python*

Mark Summerfield

editora **Alta Books** – Rio de Janeiro – 2013

- Introdução à Programação com Python

Algoritmos e Lógica de Programação para iniciantes

Nilo Ney Coutinho Menezes – editora Novatec – 3ª edição – 2019

- *Data Science do Zero*

Primeiras Regras com o *Python*

Joel Grus - O'REILLY

editora **Alta Books** – Rio de Janeiro – 2016

- *Sistemas Digitais – princípios e aplicações - 11a. Edição*

Ronald J. Tocci / Neal S. Wildmer /  
Gregory L. Moss

Pearson / Prentice Hall - 2011