

**Prof. Dr. Valdir Gil Pillat**

# Definição

- ➊ Python é uma linguagem de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991.
- ➋ Está disponível para diversos sistemas operacionais no site: <https://www.python.org/downloads/>
- ➌ Existe duas versões Python 2 e Python 3

# Tipos de Dados

- **String** - Na programação String representa um conjunto de caracteres disposto numa determinada ordem. A partir de agora, todas as vezes em que falarmos o termo String, estaremos nos referindo a um conjunto de caracteres.
- **Numérico** - Um segundo tipo de informação são os dados compostos por caracteres numéricos. Os números são divididos em 2 partes:
  - **inteiros** - chamados de integer ou int
  - **ponto flutuante** - chamado de float

# Operadores

## Aritméticos

Operação	Símbolo
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Exponenciação	**
Módulo	%

## Relacionais

Operação	Símbolo
Igual a	==
Diferente de	!=
Maior que	>
Menor que	<
Maior ou igual a	>=
Menor ou igual a	<=

## Lógicos

Operação	Símbolo
E	AND
OU	OR
NÃO	NOT

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 26 2016, 10:47:25)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.
```

É possível executar o código direto por aqui, mas neste curso vamos criar um programa para cada programa.

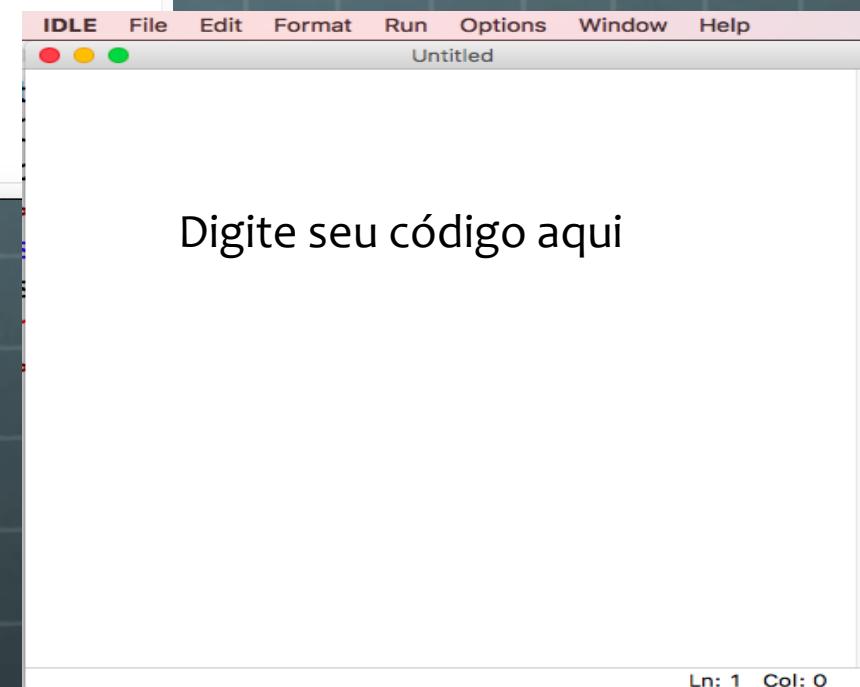
Portanto:

**Selezione: File -> New File**

Aparecerá a tela ao lado

IDE

**Para desenvolver um programa em Python podemos utilizar diversas IDE, durante este curso é utilizado o IDLE (nativo do Python)**



Digite seu código aqui

# Alô Mundo

- ➊  `print ('Alô Mundo!!!!')`
- ➋  Digite esta linha na janela que foi aberta.
- ➌  Depois selecione no menu: **Run -> Run Module**
- ➍  Ou pressione a tecla: **F5**
- ➎  Será solicitado para salvar o arquivo (`ex1.py`). Salve na pasta **Oficina** disponível no seu desktop (**Área de trabalho**)

# Entrada de Dados

- A entrada de dados utilizando o teclado na linguagem Python é realizada pela função **input**. Esta função lê toda informação digitada pelo usuário como texto (string).
- Exemplo (ex2.py):

```
nome=input('Digite seu nome:')
```

```
print('Bem vindo', nome)
```

# Entrada de Dados

- Não tem declaração de variáveis. O tipo da variável é definido dinamicamente, conforme sua primeira atribuição.
- Se for necessário converter a entrada em um inteiro por exemplo, deve ser utilizada uma operação *cast*. A conversão pode ser realizada para qualquer variável: int, float, str ...
- Exemplo (ex3.py)

```
num=int(input('Digite um número:'))
```

```
print('O número digitado foi:', num)
```

# Saída de dados

- Para exibir as informações do programa na tela é utilizada a função print.
- Esta função pode ser utilizada diretamente sem formatação do texto, apenas concatenando as informações.

```
print('Exemplo de print')
```
- ```
A=10
```
- ```
print('Exemplo com variável',A,'e concatenando outro texto',A,'só separar por vírgula')
```

# format

- O **print** e o **input** podem ser formatados utilizando a função **format**.
- Exemplo: (ex4.py)  
`for x in range(1, 11):  
 print('{0:2d} {1:3d} {2:4d}'.format(x, x*x, x*x*x))`

```
1   1     1
2   4     8
3   9     27
4  16    64
5  25   125
6  36   216
7  49   343
8  64   512
9  81   729
10 100 1000
```

<https://docs.python.org/3/tutorial/inputoutput.html>

# Outros exemplos

- `print('{0} and {1}'.format('spam', 'eggs'))`
  - spam and eggs
- `print('{1} and {0}'.format('spam', 'eggs'))`
  - eggs and spam
- `print('The value of PI is approximately {0:.3f}'.format(math.pi))`
  - The value of PI is approximately 3.142.
- `table = {'Sjoerd': 4127, 'Jack': 4098, 'Dcab': 7678}`
  - `for name, phone in table.items():`
  - `print('{0:10} ==> {1:10d}'.format(name, phone))`
    - Jack ==> 4098
    - Dcab ==> 7678
    - Sjoerd ==> 4127

# Exercícios 1

1. Efetuar a soma dos números 5 e 10 e imprimir o resultado
2. Efetuar a soma de três números digitados pelo usuário e imprimir o resultado
3. Crie um programa para ler e escrever a primeira letra de seu nome
4. Calcular o salário líquido do funcionário sabendo que este é constituído pelo salário bruto mais o valor das horas extras subtraindo 8% de INSS do total. Serão lidos o salário bruto, o valor das horas extras e o número de horas extras e o número de horas extras. Apresentar ao final o salário líquido, no seguinte formato R\$ xxxx,xx.

# Estruturas de programação

- ➊ **Decisão ou Desvio:** Decide qual será a próxima linha a ser executada.
  - ➌ if
- ➋ **Repetição:** Define até quando um determinado trecho de código será executado.
  - ➌ for
  - ➌ while
- ➌ Em Python, os **limites das estruturas de programação** (Decisão, repetição ou funções) são definidos pela **identação** (tabulação).

# **if ... elif ... else**

#Exemplo: (ex5.py)

```
n1=float(input('Digite um número:'))
```

```
n2=float(input('Digite um número:'))
```

```
media=(n1+n2)/2
```

```
if media >=5:
```

```
    print ('Aluno aprovado', media)
```

# **if ... elif ... else**

#Exemplo: (ex5.py)

```
n1=float(input('Digite um número:'))
```

```
n2=float(input('Digite um número:'))
```

```
media=(n1+n2)/2
```

```
if media >= 5:
```

```
    print ('ALUNO APROVADO')
```

```
else:
```

```
    print ('ALUNO REPROVADO')
```

# **if ... elif ... else**

```
num = int(input('Digite um numero menor que 5'))
```

```
if num > 5:
```

```
    print ('Oops!')
```

```
    print ('O numero digitado e maior que 5')
```

```
elif num < 0:
```

```
    print ('Oops!')
```

```
    print ('O numero digitado e negativo')
```

```
else:
```

```
    print ('Numero digitado: ',num)
```

# Exercícios 2

- 1. Construa um programa que leia dois valores numéricos inteiros e efetue a adição; caso o resultado seja maior que 10, apresenta-lo.**
  
- 2. Escreva um programa que receba um número e imprima uma das mensagens: “É múltiplo de 3” ou “Não é múltiplo de 3”.**
  
- 3. Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um programa que calcule seu peso ideal, utilizando as seguintes fórmulas:**
  - 1. Para homens:  $(72.7 * h) - 58$**
  - 2. Para mulheres:  $(62.1 * h) - 44.7$**

**(h=altura)**

# while

#Exemplo: (ex6.py)

Nr = 0

while Nr <=100:

Nr = Nr + 1

print (Nr)

# for / range

```
for Nr in range(100):  
    print (Nr)
```

**range:** é uma função em Python que cria uma lista de números  
Exemplos:

```
x=range(100)  
print(x) #[0,1,2,3,...,98,99]  
y=range(0,10,2)  
print(y) #[0,2,4,6,8]
```

# Outros exemplos

```
for i in ['a', 'b', 'c', 'd', 'e', 'f', 'g']:
```

```
    if i >= 'd':
```

```
        print i
```

```
    else:
```

```
        print 'd'
```

```
d = ['arroz', 'feijao', 'carne', 'batata']
```

```
for i in d:
```

```
    print 'Estou cozinhando ', i
```

# Exercícios 3

1. Faça um programa que imprima todos os números inteiros de 0 a 50.
2. Imprima os múltiplos de 3 entre o intervalo 0 e 50.
3. A conversão de graus Fahrenheit para Celsius é obtida pela fórmula  $[C=5*(TF-32)/9]$ . Construa um programa que calcule e apresente TODAS as temperaturas (em Celsius) correspondentes aquelas em Fahrenheit de 1 até 50.

# String e listas

## Quebrando um texto

- poesia = "O orvalho no carvalho..."
- x=poesia.split() #['O', 'orvalho', 'no', 'carvalho...']
- y=poesia.split('va') #['O or', 'lho no car', 'lho...']

## Juntando um texto

- lista = ['O', 'orvalho', 'no', 'carvalho...']
- y=' '.join(lista) #'O orvalho no carvalho...'
- z='\_'.join(lista)
- print(y,z) #'O\_orvalho\_no\_carvalho...'

# Funções

- A criação de funções em Python é utilizada através do comando **def**. Veja o exemplo abaixo

```
def func(a,b): #definição da função
```

```
c=a+b
```

```
return c #retorno da função [opcional]
```

```
x = int(input("Entre com um numero"))
```

```
y = int(input("Entre com outro numero"))
```

```
z = func(1,2)
```

```
print (z)
```

# Funções (Exemplo 2)

```
def func(a,b): #definição da função
```

```
c=a+b
```

```
return c #retorno da função [opcional]
```

```
def principal():
```

```
x = int(input("Entre com um numero"))
```

```
y = int(input("Entre com outro numero"))
```

```
z = func(1,2)
```

```
print (z)
```

```
principal() #função que é chamada na execução do programa
```

# numpy

**Lista, Vetor, Array**

# numpy

- ➊ NumPy é um módulo do Python que suporta operações com vetores e matrizes.
- ➋ Instalação do numpy
  - ➌ pip install numpy
  - ➌ python -m pip install numpy (windows)
- ➌ Importar o módulo
  - ➌ import numpy as np

# Listas em Python

- Em Python uma variável do tipo array, vetor ou matriz é definida como lista.
- Uma lista em Python é criada da seguinte forma:
- Lista vazia
  - NomeLista = []
- Lista com argumentos
  - NomeLista = [0,1,2,3,45,67]
  - NomeLista2 = ['a','b','c']
  - Lista1 = range (1,5) #[1,2,3,4]
  - Lista2 = range(0,5) #[0,1,2,3,4]
  - Lista3 = range(1,10,2) #[1,3,5,7,9]
- Adicionando elemento na lista
  - NomeLista.append(30)

# Exercícios 4

1. Vamos alterar o exercício 3.3 para armazenar os valores em Fahrenheit em uma lista, depois mostrar esta lista.

# numpy

- A função **array** transforma uma lista do Python para um array em NumPy.
- **Lista vazia**
  - `a=np.array([])`
- **Lista com argumentos**
  - `NomeLista= np.array([0,1,2,3,45,67])`
  - `NomeLista2 = np.array(['a','b','c'])`
  - `Lista1 = np.arange (1,5) #[1,2,3,4]`
  - `Lista2 = np.arange(5) #[0,1,2,3,4]`
  - `Lista3 = np.arange(1,10,2) #[1,3,5,7,9]`
  - `Lista4 = np.zeros(3) #é possível adicionar o tipo , dtype=np.int16`
- **Adicionando elemento na lista**
  - `Lista3=np.append(Lista3,50)`
  - `Lista3=np.insert(Lista3,0,30) #Lista, posição, valor`

# Exercícios 5

1. Vamos alterar o exercício 4.1 para armazenar os valores em Fahrenheit em uma array numpy, depois mostrar esta lista.

# Numpy (array)

- Acessar elementos
  - `print (Lista3[0]) #30`
  - `print (Lista3[-1]) #50`
  - Índices negativos nas listas a contagem começa do final
- Membro de uma lista
  - `print (1 in Lista3, 6 in Lista3)`
  - `for i in Lista3:`
  - `print (i)`
- Comprimento da lista
  - `len (Lista3) #7`
  - `Lista3.shape #(7,)`
- Tipo do array
  - `Lista3.dtype #int32`

# Numpy Operações com arrays

## Concatenar listas

- a = [1, 2, 3]
- b = [4, 5, 6]
- c=np.concatenate((a,b), axis=0)
- print (c) #[1, 2, 3, 4, 5, 6]

# Operações com listas (ex7.py)

## Fatiamento de listas

-  `lista = np.array(['a', 'b', 'c', 'd', 'e', 'f'])`
-  `lista[1:3] #['b', 'c']`
-  `lista[:4] #['a', 'b', 'c', 'd']`
-  `lista[3:] #['d', 'e', 'f']`
-  `lista[:] #['a', 'b', 'c', 'd', 'e', 'f']`

## Remoção de lista

-  `lista = ['a', 'b', 'c', 'd', 'e', 'f']`
-  `lista1=np.delete(lista,range(1,5))`
-  `print (lista1) # ['a', 'f']`

## Clonando Lista

-  `a = np.array([1, 2, 3])`
-  `b = np.copy(a); print (b) #[1, 2, 3]`

# Matrizes

n\_linhas=3

n\_colunas=3

```
matriz = np.zeros((n_linhas,n_colunas)) # lista c/ zeros
```

```
for i in range(n_linhas):
```

```
    for j in range(n_colunas):
```

```
        matriz[i,j]=i+j
```

```
print (matriz)
```

# Operações com Arrays (numpy)

## Elemento a elemento ou por escalar

- Adição/subtração (ex8.py)

- `a = np.array([20,30,40,50])`
  - `b = np.arange(4) #([0, 1, 2, 3])`
  - `c = a-b #([20, 29, 38, 47])`

- Multiplicação/Divisão

- `d=a*b`

- Outras operações

- `e=5*np.sin(a)`
  - `f=a<30 #retorna verdadeiro ou falso para cada elemento`

# Operações com Arrays (numpy)

## Como matriz

- Operações aritméticas entre matrizes precisa utilizar a função dot
- Multiplicação (ex9.py)
  - a.dot(b) #ou np.dot(a,b)
    - #note que b deve ter o mesmo número de linhas que a possui de colunas
  - b.transpose() #gera a transposta de uma matriz

# Operações com Arrays (numpy)

## Outras operações

- matriz.max() #(Maior valor do array)
- matriz.min() #(Menor valor do array)
- matriz.sum() #Soma todos os elementos do array
- Utilizando o parâmetro axis. Obs.: Pode ser utilizada em outras funções aqui é mostrada na função sum
  - matriz.sum(axis=0) #soma das linhas
  - matriz.sum(axis=1) #soma das colunas
- matriz.argmax() #(posição do maior valor do array)
- matriz.sort() #ordena o array
- Outras funções no site: <http://www.numpy.org/>
- Inclusive a correlação de algumas funções com o Matlab

# Exercícios

1. Gerar um vetor com os números inteiros de zero a cinco
2. Gerar um vetor com os números reais de zero a cinco com o passo 0,1
3. Mostrar o segundo elemento do vetor
4. Gerar um vetor constituído de 10 zeros
5. Agora com 100 zeros
6. Construa um vetor constituído pelos números pares de 0 a 10 seguidos pelos números ímpares de 0 a 10.
7. Sendo  $x=[2\ 3\ 7]$  e  $y= [0\ -1\ 3]$  escreva a resposta de cada um desses comandos executados no Python.
  - a)  $x + y$
  - b)  $x - y$
  - c)  $x * y$
8. Como gerar a partir do vetor  $x=np.arange(0,1,0.001)$  um vetor com números de 1 a 11.
9. Sendo  $x=[2.1\ -2\ 3]$  e  $y= [0\ -1\ 3]$  escreva a resposta de cada um desses comandos executados no Python.
  - a)  $x+y$
  - b)  $3*x$
  - c)  $x*y$
  - d)  $x/y$
  - e)  $y**2$
  - f)  $x**y$

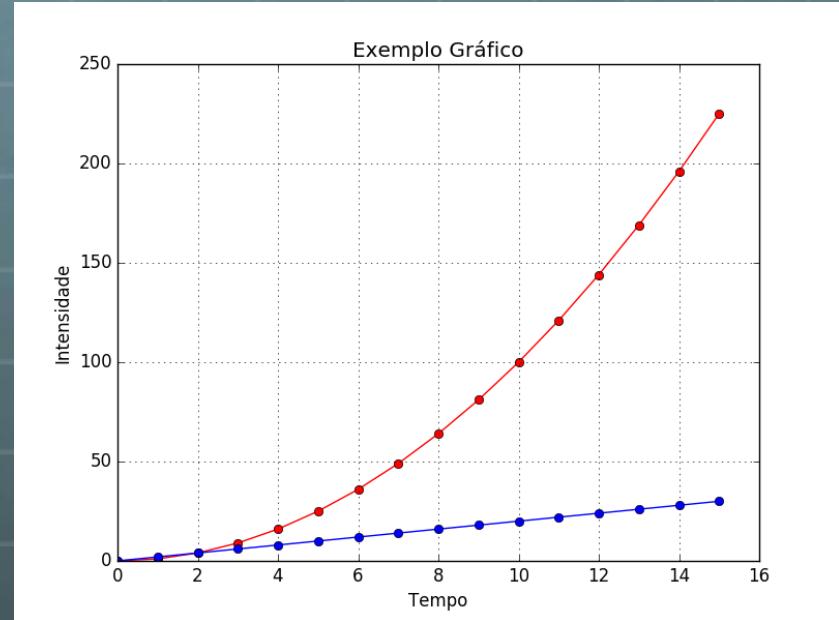
# Gráficos

matplotlib (2D)

# Instalação

- **Matplotlib:** Módulo utilizado para plotar gráficos. Existem vários recursos neste módulo: plot, plot3D, animações
- **Instalação**
  - `python -m pip install matplotlib` (windows)
  - `pip install matplotlib` (outros)
- <http://matplotlib.org/>
- <http://www.labri.fr/perso/nrougier/teaching/matplotlib/>

```
import numpy as np
import matplotlib.pyplot as plt
Dados=np.zeros((16,2))
for x in range(0,16):
    Dados[x,0]=x
    Dados[x,1]=x**2
Dados2=np.zeros((16,2))
for x in range(0,16):
    Dados2[x,0]=x
    Dados2[x,1]=x*2
plt.plot(Dados[:,0],Dados[:,1],'-ro')
plt.plot(Dados2[:,0],Dados2[:,1],'-bo')
plt.title('Exemplo Gráfico')
plt.ylabel('Intensidade')
plt.xlabel('Tempo')
plt.grid(True)
plt.savefig('Grafico1.png')
plt.show()
```



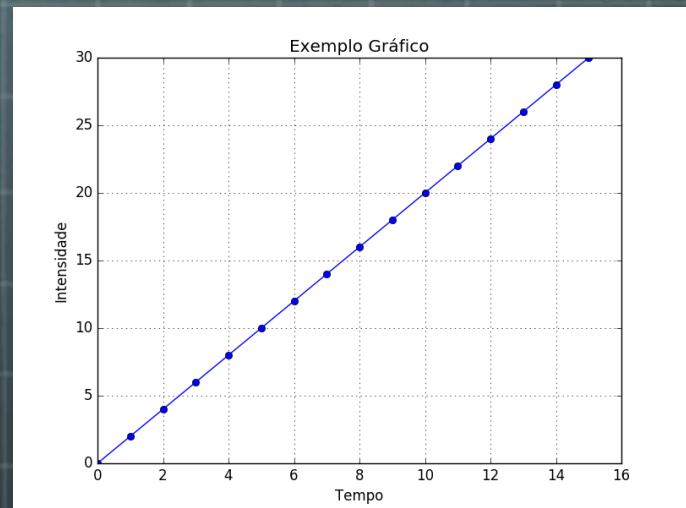
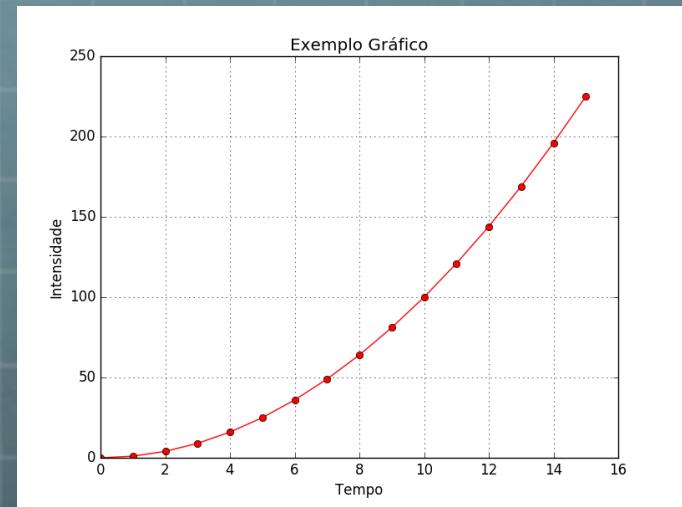
# Exercícios 6

1. Vamos plotar um gráfico com o array criado no exercício 4.1

```

import numpy as np
import matplotlib.pyplot as plt
Dados=np.zeros((16,2))
for x in range(0,16):
    Dados[x,0]=x
    Dados[x,1]=x**2
Dados2=np.zeros((16,2))
for x in range(0,16):
    Dados2[x,0]=x
    Dados2[x,1]=x**2
plt.figure(1)
plt.plot(Dados[:,0],Dados[:,1],'-ro')
plt.title('Exemplo Gráfico')
plt.ylabel('Intensidade'); plt.xlabel('Tempo')
plt.grid(True); plt.savefig('Grafico2.png')
plt.figure(2)
plt.plot(Dados2[:,0],Dados2[:,1],'-bo')
plt.title('Exemplo Gráfico')
plt.ylabel('Intensidade'); plt.xlabel('Tempo')
plt.grid(True);plt.savefig('Grafico3.png'); plt.show()

```



# Subplot

subplot(2,1,1)

subplot(2,1,2)

subplot(1,2,1)

subplot(1,2,2)

subplot(2,2,1)

subplot(2,2,2)

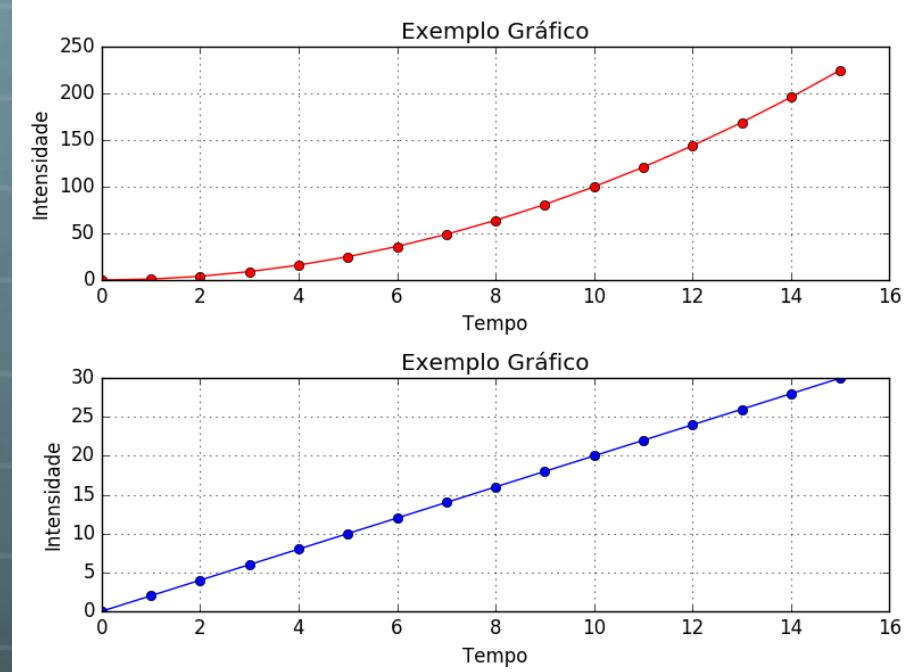
subplot(2,2,3)

subplot(2,2,4)

```

import numpy as np
import matplotlib.pyplot as plt
Dados=np.zeros((16,2))
for x in range(0,16):
    Dados[x,0]=x
    Dados[x,1]=x**2
Dados2=np.zeros((16,2))
for x in range(0,16):
    Dados2[x,0]=x
    Dados2[x,1]=x**2
plt.subplot(211)
plt.plot(Dados[:,0],Dados[:,1],'-o',color=(1.0, 0.0,0.0))
plt.title('Exemplo Gráfico')
plt.ylabel('Intensidade'); plt.xlabel('Tempo');
plt.grid(True)
plt.subplot(212)
plt.plot(Dados2[:,0],Dados2[:,1],'-bo')
plt.title('Exemplo Gráfico')
plt.ylabel('Intensidade'); plt.xlabel('Tempo');
plt.grid(True)
plt.tight_layout(); plt.savefig('Grafico4.png');plt.show()

```



# Exercício 7

1. Modifique o exemplo anterior para a apresentar os gráficos em duas colunas e uma linha. (ex10.py)

# stem

```
x=np.arange(-5,5)
y=x**2
plt.subplot(211)
plt.stem(x,y)
plt.subplot(212)
plt.stem(x,y,'-k',markerfmt='ko',label='test', basefmt='k-')
plt.legend()
plt.show()
```

