

Eines per la representació de
dades en Python.

Eines per la representació de dades en Python.

- 1.- Tipus de dades.
- 2.- La llibreria *matplotlib*.
- 3.- Dades estàtiques. Exemples de representació.
- 4.- Representació en 3D.
- 5.- Dades dinàmiques. Animació.
- 6.- Dades dinàmiques. Connexió amb una font externa.

1.- Tipus de dades.

Int:

`x=12, y=6, z = x+y, print z`

Float:

`x=3.0, y=6.0, z= x/y, print z`

Complex:

`x=2+1j, y=3-1j, z=x+y, print z (z.real,z.imag,...)`

String:

`s1='hola', s2='adeu', print s1+s2`

Set:

`x={1,2,3} , y={2,3} , print x and y, print x or y, print x-y`

List:

`x=[0,1,2] , y=[0,2,4] , print x+y , x.append(1), print x`

1.- Tipus de dades.

Llistes

Creació de llistes:

```
x=range(20)
y=range(20)
print x
for i in range(20):
    y[i]=x[i]*x[i]
print y
```

però...

```
y.append('patata')
print y
```

Tipus heterogeni !!!!

2.- La llibreria *matplotlib*.

Permet la representació gràfica de dades (2d,3d,animació)

Exemple:

```
import matplotlib.pyplot as plt
```

```
x=[0,1,2,3]
```

```
y=[1,2,4,8]
```

```
plt.plot(x,y)
```

```
plt.show()
```

2.- La llibreria *matplotlib*.

Permet la representació gràfica de dades (2d,3d,animació)

Exemple:

```
import matplotlib.pyplot as plt
```



Importem la llibreria i li
diem **plt**

```
x=[0,1,2,3]  
y=[1,2,4,8]
```



Creem dues llistes d'elements

```
plt.plot(x,y)
```



Pintem les dues llistes

```
plt.show()
```



Mostrem el gràfic

2.- La llibreria *matplotlib*.

Variacions en la representació:

`plt.plot(x,y,'r') ... plt.plot(x,y,'ro') ... plt.plot(x,y,'--go')`

```
import matplotlib.pyplot as plt
```

```
x=[0,1,2,3]
```

```
y=[1,2,4,8]
```

```
plt.plot(x,y,'r')
```

```
plt.show()
```

2.- La llibreria *matplotlib*.

Dominem el gràfic!

```
import matplotlib.pyplot as plt
```

```
x=[0,1,2,3]
```

```
y=[1,2,4,8]
```

```
plt.axis([0,5,0,10])
```

```
plt.plot(x,y,'--go')
```

```
plt.show()
```


2.- La llibreria *matplotlib*.

Més!

```
import matplotlib.pyplot as plt
```

```
x=[0,1,2,3]
```

```
y=[1,2,4,8]
```

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

```
plt.axis([0,5,0,10])
```

```
plt.grid(True)
```

```
plt.xlabel('Dies')
```

```
plt.ylabel('Diners')
```

```
plt.plot(x,y,'--go')
```

```
plt.plot(x,z,'--bo')
```

```
plt.show()
```

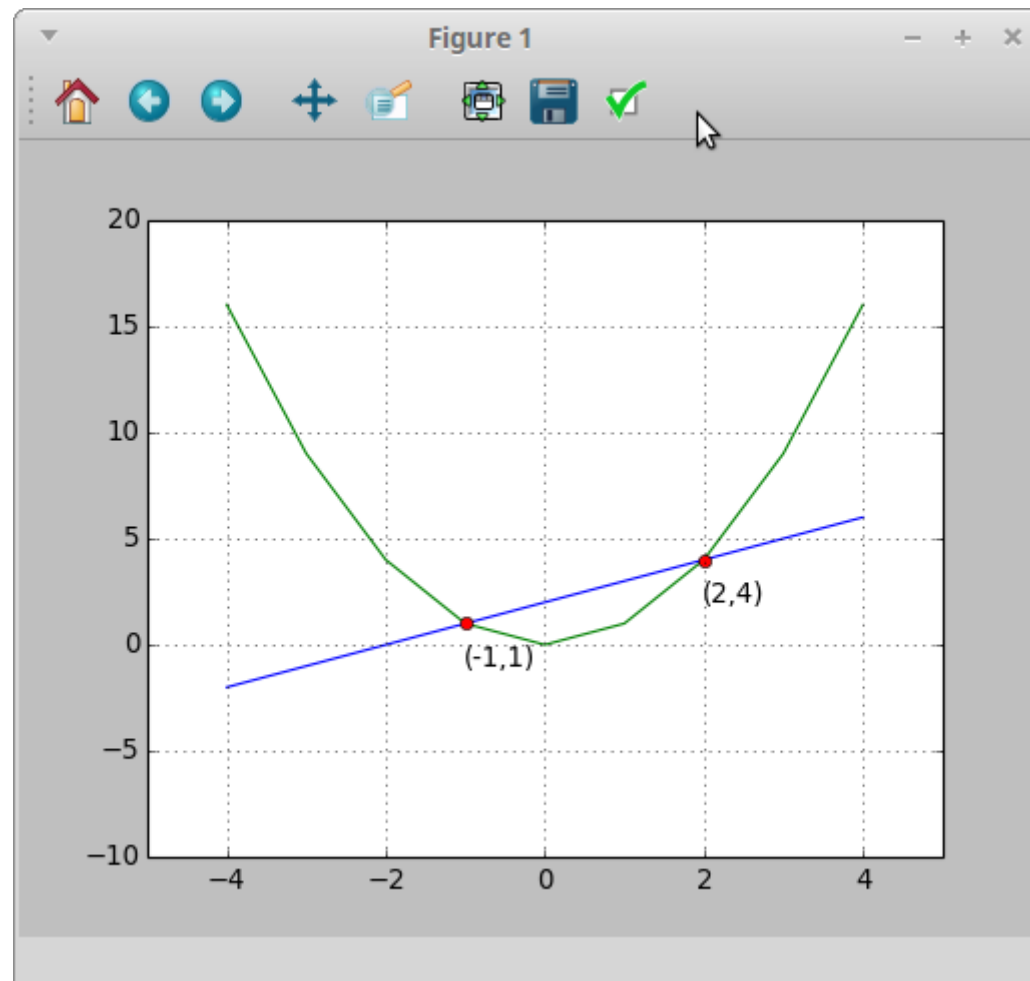
2.- La llibreria *matplotlib*.

Exercici:

Pintar les funcions $y = x^2$, $z = 2+x$
i mostrar els seus punts de tall...

2.- La llibreria *matplotlib*.

Exemple:



2.- La llibreria *matplotlib*.

Exemple:

```
import matplotlib.pyplot as plt
```

```
x=[-4,-3,-2,-1,0,1,2,3,4]
```

```
y=[16,9,4,1,0,1,4,9,16]
```

```
z=[-2,-1,0,1,2,3,4,5,6]
```

```
tx=[-1,2]
```

```
ty=[1,4]
```

```
plt.axis([-5,5,-10,20])
```

```
plt.grid(True)
```

```
plt.plot(x,y,'g')
```

```
plt.plot(x,z,'b')
```

```
plt.plot(tx,ty,'ro')
```

```
plt.text(2,2,'(2,4)')
```

```
plt.text(-1,-1,'(-1,1)')
```

```
plt.show()
```

3. Dades estàtiques. Exemples de representació.

Les llistes ofereixen problemes de tipus i d'eficiència.
S'aconsella treballar amb 'arrays' i concretament de la
llibreria numpy (numerical python)

```
import numpy as np
```

```
PI=np.pi
```

```
t1 = np.arange(0.0, PI, 0.01)
```

```
print t1
```

3. Dades estàtiques. Exemples de representació.

```
import numpy as np
import matplotlib.pyplot as plt
PI=np.pi
t1 = np.arange(0.0, 4*PI, 0.05)
```

```
plt.figure(1)
plt.plot(t1, np.sin(t1),'g')
```

Passem dos arrays de numpy.
El sinus s'aplica a tot l'array!

```
plt.show()
```

3. Dades estàtiques. Exemples de representació.

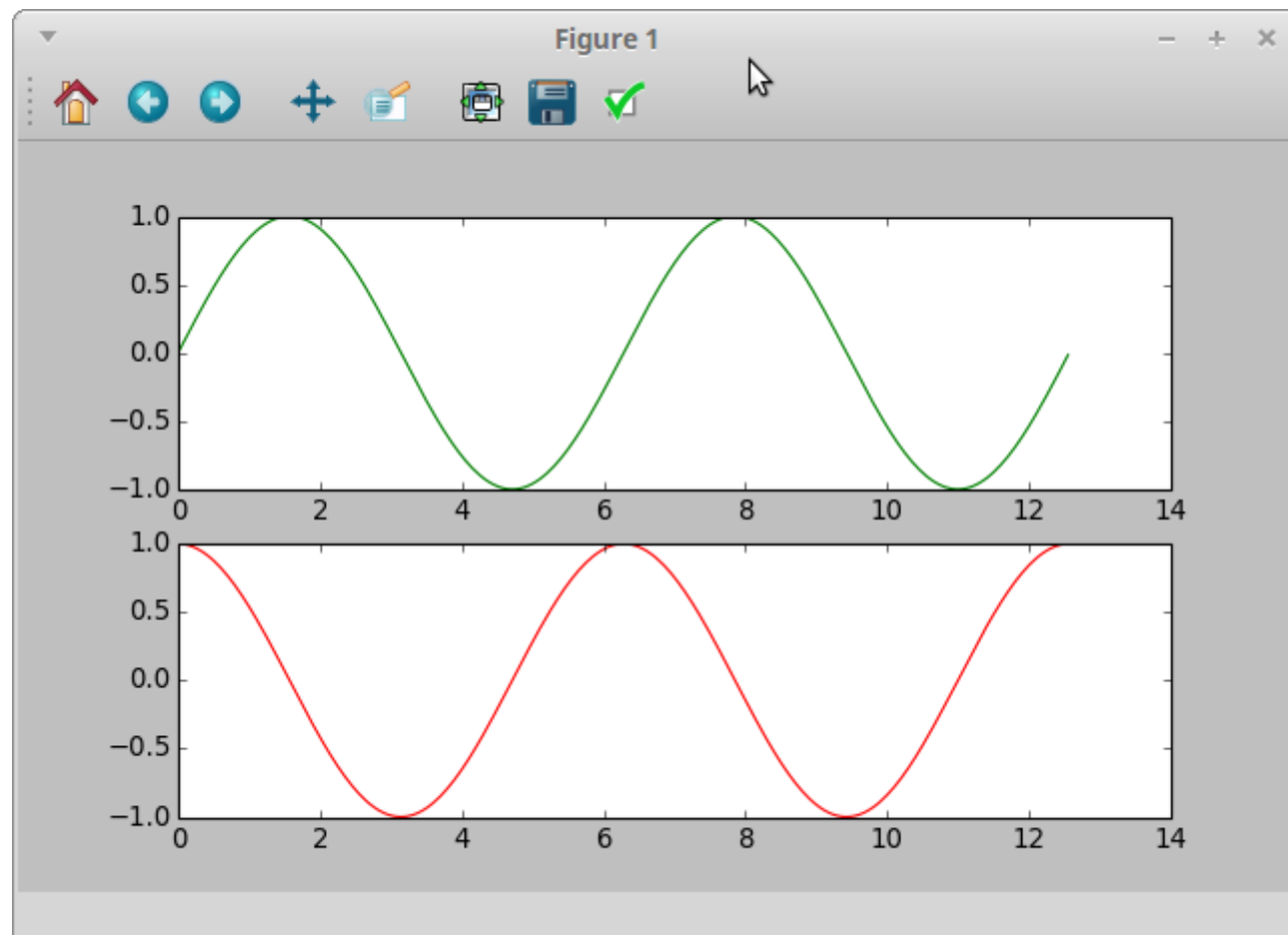
Múltiples plots!

```
import numpy as np
import matplotlib.pyplot as plt
PI=np.pi
t1 = np.arange(0.0, 4*PI, 0.05)

plt.figure(1)
plt.subplot(211)
plt.plot(t1, np.sin(t1),'g')
plt.subplot(212)
plt.plot(t1, np.cos(t1),'r')

plt.show()
```

3. Dades estàtiques. Exemples de representació.



3. Dades estàtiques. Exemples de representació.

Múltiples plots!

```
import numpy as np
import matplotlib.pyplot as plt
PI=np.pi
t1 = np.arange(0.0, 4*PI, 0.05)
```

```
plt.figure(1)
plt.subplot(121)
plt.plot(t1, np.sin(t1),'g')
plt.subplot(122)
plt.plot(t1, np.cos(t1),'r')
```

```
plt.show()
```

Proveu 12 en comptes de 21

3. Dades estàtiques. Exemples de representació.

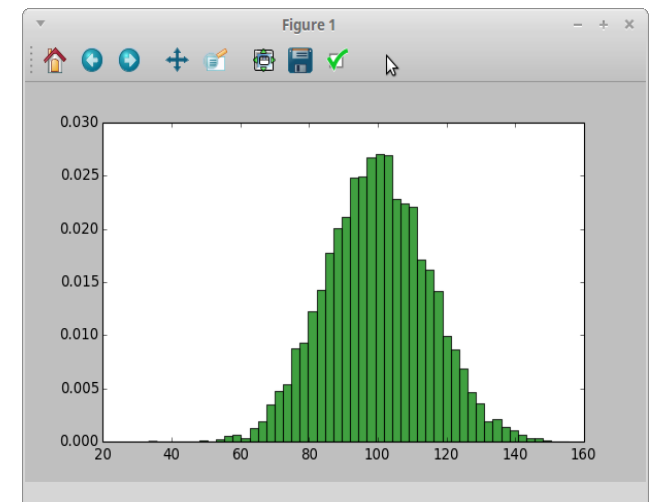
```
import numpy as np  
import matplotlib.pyplot as plt
```

```
mig, desv = 100, 20  
x = mig + desv * np.random.randn(10000)
```

```
n, bins, patches = plt.hist(x, 50, normed=1, facecolor='g', alpha=0.75)
```

```
plt.show()
```

Histograma:



4.- Representació en 3D.

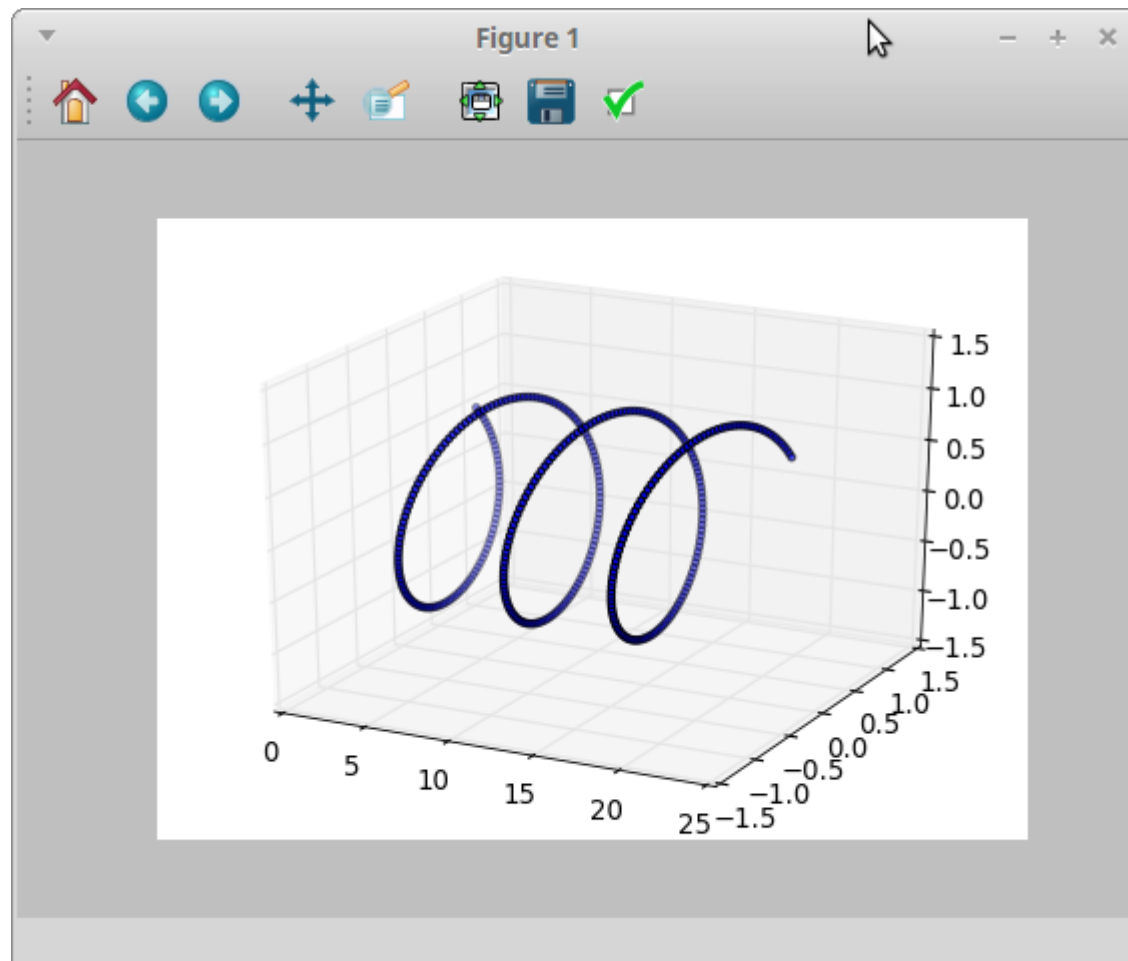
```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

xs=np.arange(1,20,0.05)
ys=np.sin(xs)
zs=np.cos(xs)

Axes3D.scatter(ax,xs,ys,zs,zdir='z',s=10,c='b')
plt.show()
```

4.- Representació en 3D.



4.- Representació en 3D.

```
import matplotlib.pyplot as plt  
from mpl_toolkits.mplot3d import Axes3D  
import numpy as np
```

Llibreria mplot3d!

```
fig = plt.figure()  
ax = fig.add_subplot(111, projection='3d')
```

Creació d'un objecte 'axes'

```
xs=np.arange(1,20,0.05)  
ys=np.sin(xs)  
zs=np.cos(xs)
```

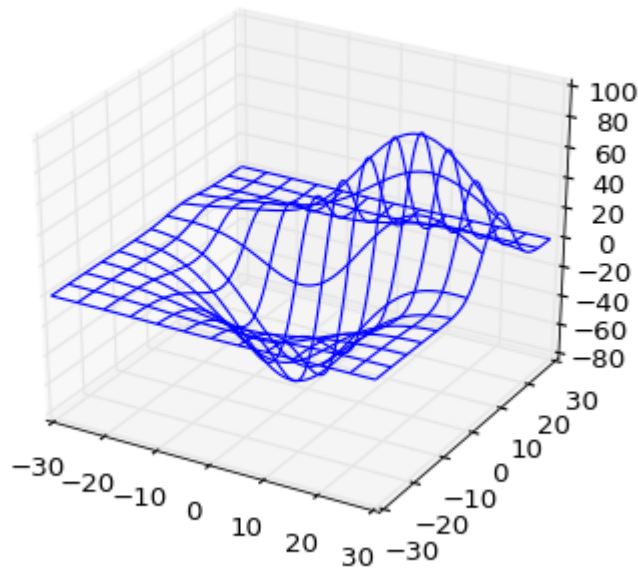
Omplenat amb uns punts x,y,z
Proveu:

```
Axes3D.scatter(ax,xs,ys,zs,zdir='z',s=10,c='b')  
plt.show()
```

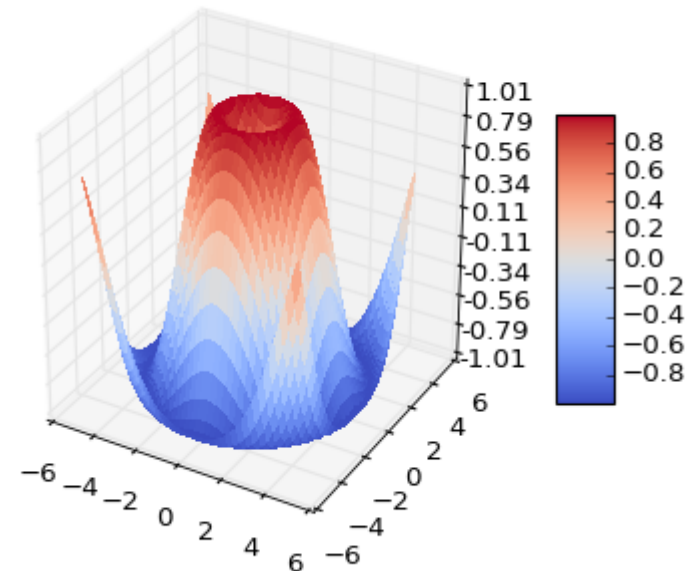
- canviar el rang per posar menys punts: 0.05 → 0.5
- canviar scatter per plot!

(Recordeu: Scatter és discret!)

4.- Representació en 3D.



`plot_wireframe(...)`



`plot_surface(...)`

(+info: http://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html)

5.- Dades dinàmiques. Animació.

Nova estructura!

```
import matplotlib.pyplot as plt
```

```
import matplotlib.animation as animation
```

Llibreria animation

```
def animate(i):
```

Funció que es crida periòdicament

```
...
```

```
def init():
```

Funció per inicialitzar la figura (estat inicial)

```
...
```

```
fig = plt.figure()
```

```
ax = fig.add_subplot(111)
```

```
...
```

Crida que engega el procés. Interval (ms). Blit (aprofitar)

```
ani = animation.FuncAnimation(fig, animate, 100, interval=50, blit=False)
```

```
plt.show()
```

Example, punt:

```
import matplotlib.pyplot as plt
import matplotlib.animation as animation
```

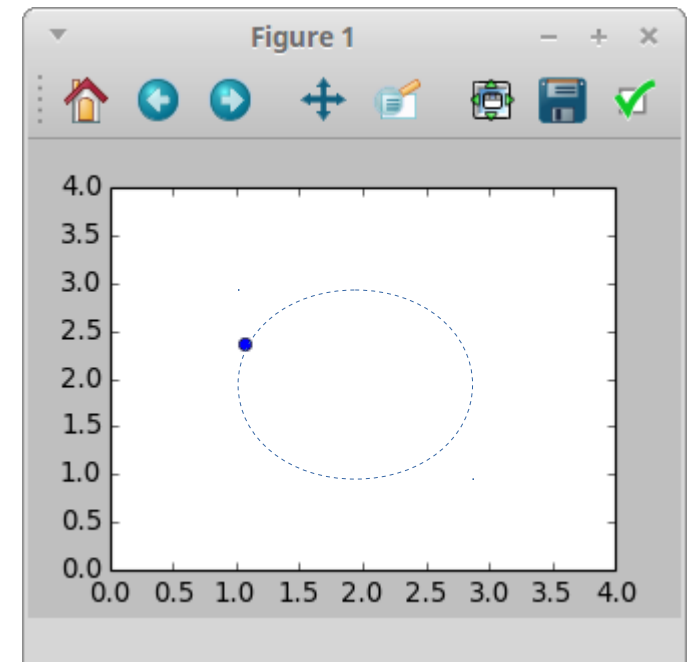
```
def animate(i):
    x[0],y[0] = 2+sin(6.28*i/100.0) , 2+cos(6.28*i/100.0)
    line.set_xdata(x)
    line.set_ydata(y)
    return line,
```

```
def init():
    line.set_ydata([0])
    return line,
```

```
fig = plt.figure()
ax = fig.add_subplot(111)
```

```
x,y = [2],[2]
line, = ax.plot(x,y,'o')
ax.axis([0,4,0,4])
```

```
ani = animation.FuncAnimation(fig,animate,100,interval=50,blit=False)
plt.show()
```



Exemple, Sinus:

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
```

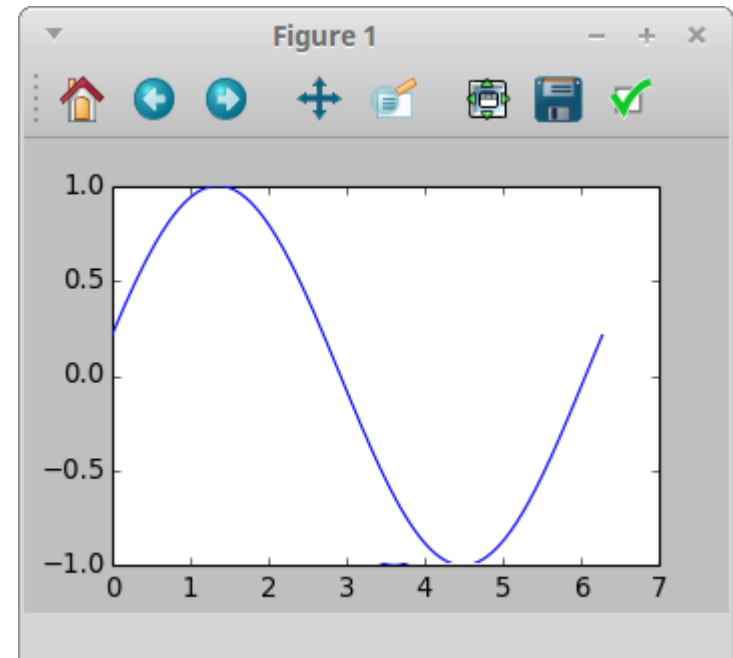
```
def animate(i):
    line.set_ydata(np.sin(x+i/10.0))
    return line,
```

```
def init():
    line.set_ydata(np.ma.array(x, mask=True))
    return line,
```

```
fig = plt.figure()
ax = fig.add_subplot(111)
```

```
x = np.arange(0, 2*np.pi, 0.01)
line, = ax.plot(x, np.sin(x))
```

```
ani = animation.FuncAnimation(fig, animate, 200, init_func=init,
    interval=50, blit=True)
plt.show()
```



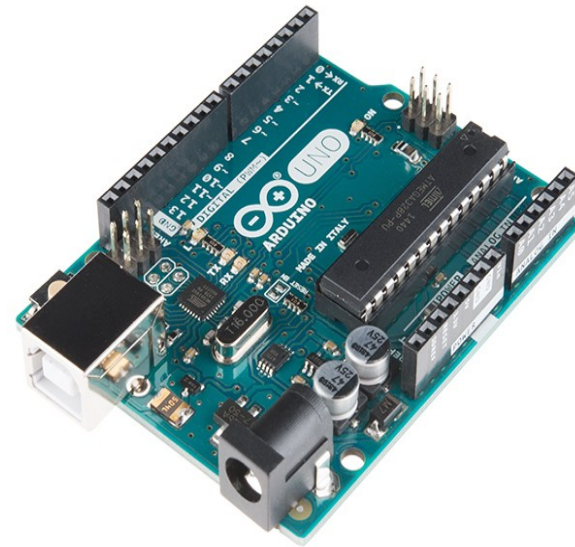
6.- Dades dinàmiques. Connexió amb una font externa.

Usarem una placa Arduino UNO per entrar dades externes al PC.

Ens fa falta un driver i el software de programació.

www.arduino.cc

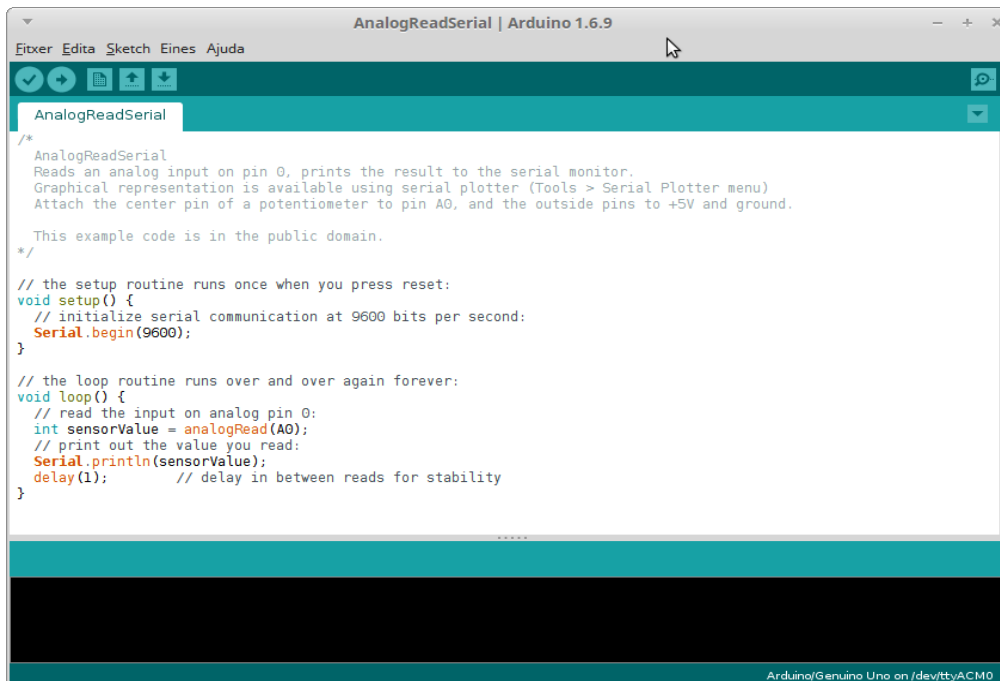
Download, windows zip for non admin
(windows, linux, Mac)



6.- Dades dinàmiques. Connexió amb una font externa.

Prova bàsica: programa d'exemple: Basics → blink (canvieu els temps)

Prova avançada: programa d'exemple: Basics → AnalogReadSerial

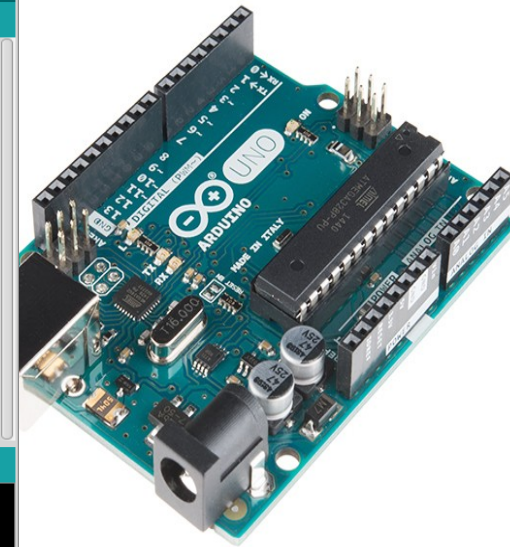
A screenshot of the Arduino IDE interface. The title bar reads 'AnalogReadSerial | Arduino 1.6.9'. The menu bar includes 'Fitxer', 'Edita', 'Sketch', 'Eines', and 'Ajuda'. The toolbar shows icons for opening, saving, and running. The main text area contains the following code:

```
/*
  AnalogReadSerial
  Reads an analog input on pin 0, prints the result to the serial monitor.
  Graphical representation is available using serial plotter (Tools > Serial Plotter menu)
  Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.

  This example code is in the public domain.
  */

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
```

The bottom of the window shows a black serial monitor area.

6.- Dades dinàmiques. Connexió amb una font externa.

AnalogReadSerial: Què fa?

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}
```

```
// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);      // delay in between reads for stability
}
```

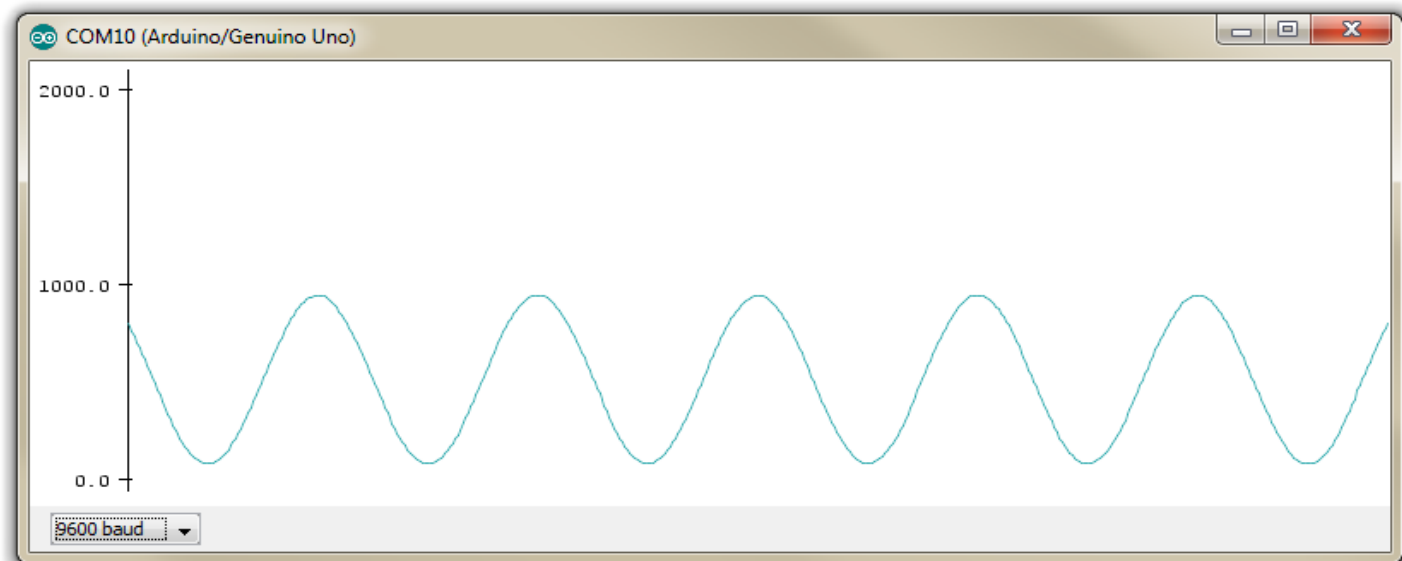
6.- Dades dinàmiques. Connexió amb una font externa.

AnalogReadSerial: Què posem a A0 ??

- Muntarem un divisor de tensió (R-R) i l'entrarem al Pin A0 d'Arduino.

AnalogReadSerial: Com ho veiem ??

- Usem la Eina → Serial Plotter.



6.- Dades dinàmiques. Connexió amb una font externa.

Què ens falta ara? Connectar la línia sèrie a Python.

Programa:

```
import serial

Port = serial.Serial( 'COM5', 9600)
Dada = Port.readline( )
print Dada
```

6.- Dades dinàmiques. Connexió amb una font externa.

Què ens falta ara? Connectar la línia sèrie a Python.

Programa:

```
import serial
```

```
Port = serial.Serial( 'COM  
Dada = Port.readline()  
print Dada
```



NO VA!!

6.- Dades dinàmiques. Connexió amb una font externa.

Instaleu el mòdul python de línia sèrie.

Obriu un terminal: >> pip2.7 install pyserial

Programa:

```
import serial

Port = serial.Serial( 'COM5', 9600)
while (1):
    Dada = Port.readline( )
    print Dada
```

Exercici:

Representar amb Matplotlib.pyplot les dades de la línia sèrie!