

# Metodi Computazionali della Fisica

## Secondo Modulo: C++

### Settima Lezione



Utilizzo delle classi di ROOT

GUI con Python

# Una semplice ricetta

- ▶ prendete uno script di ROOT;
- ▶ aggiungete `int main()` all'inizio e `return 0;` prima dell'ultima parentesi;
- ▶ aggiungete gli header file del C++ che possono servire;
- ▶ aggiungete tutti gli header file corrispondenti alle classi di ROOT utilizzate (ad esempio, se usate `TCanvas`, `#include "TCanvas.h"`);
- ▶ aggiungete un "link" al sistema per aprire le finestre;
- ▶ compilate: `g++ nomefile.cpp 'root-config --lib' -I'root-config --incdir'`

## Un semplice esempio: lo script

```

{
  gROOT->Reset();

  Float_t xf[2000],dx[2000],fit[2000];
  Float_t xt[2000],train[2000],errorrt[2000];

  [...]

  TCanvas *Test = new TCanvas("Test", "TestPlot",800,600);
  TGraph *grTrain = new TGraphErrors(TrainPoints,xt,train,dx,errorrt);
  TGraph *grValid = new TGraphErrors(ValidPoints,xv,valid,dx,errorrv);
  TGraph *grFit = new TGraph(DataPoints,xf,fit);
  TF1 *orig = new TF1("sin","0.5+0.4*sin(6.28*x)");
  TH1 *Graph1 = new TH1F("Graph1",NULL,200,0.,1.0);

  [...]

  Test->Draw();
}

```

# Un semplice esempio: il main #1

```
#include <iostream>
#include <fstream>
#include <cmath>
#include "TSystem.h"
#include "TApplication.h"
#include "TCanvas.h"
#include "TF1.h"
#include "TGraph.h"
#include "TGraphErrors.h"
#include "TH1.h"
#include "TLegend.h"
#include "TLegendEntry.h"
#include "TColor.h"

using namespace std;

int main(){

    //gROOT->Reset();
```

## Un semplice esempio: il main #2

[...]

```
TApplication theApp("App", NULL, NULL);
```

```
Float_t xf[2000],dx[2000],fit[2000];
```

```
Float_t xt[2000],train[2000],error[2000];
```

[...]

```
Test->Draw();
```

```
//Test->Print("prova.png");
```

```
theApp.Run();
```

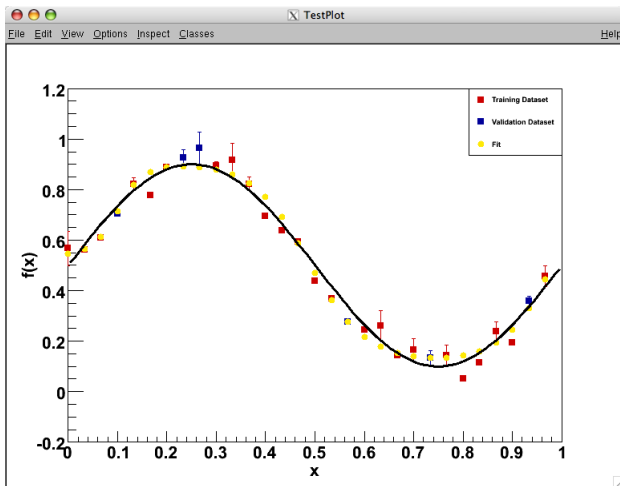
```
return 0;
```

```
}
```

## Un semplice esempio: note

- ▶ se serve solo l'output su file, scommentate la riga `Test->Print("prova.png");` (in questo caso non è necessario inizializzare una `TApplication`);
- ▶ prima di compilare, controllate i path per include e librerie con `root-config --libs` e `root-config --incdir`;

# Un semplice esempio: il risultato





# Introduzione

- ▶ Python è un linguaggio di scripting;
- ▶ Python è ObjectOriented;
- ▶ Python è multiplatforma;
- ▶ Python è semplice da usare;
- ▶ molti strumenti forniscono API per Python.

## Un esempio: il codice #1

Utilizziamo le wxWidgets per Python per realizzare un pannello di controllo che compila un codice C++, scrive alcuni parametri di input su un file e lancia il fit:

```
#!/usr/bin/python

import wx,os,sys,string

class MyFrame(wx.Frame):

    def __init__(self, parent, title):
        wx.Frame.__init__(self, parent, -1, title,
                           pos=(150, 150), size=(400, 800))

        # Create the menubar
        menuBar = wx.MenuBar()

        # and a menu
        menu = wx.Menu()
```

## Un esempio: il codice #2

```
[...]  
    # add an item to the menu, using \tKeyName automatically  
    # creates an accelerator, the third param is some help text  
    # that will show up in the statusbar  
    menu.Append(wx.ID_EXIT, "E&xit\tAlt-X", "Exit this simple sample")  
  
    # bind the menu event to an event handler  
    self.Bind(wx.EVT_MENU, self.OnTimeToClose, id=wx.ID_EXIT)  
  
    # and put the menu on the menubar  
    menuBar.Append(menu, "&File")  
    self.SetMenuBar(menuBar)  
  
    self.CreateStatusBar()  
  
    # Now create the Panel to put the other controls on.  
    panel = wx.Panel(self)  
[...]
```

## Un esempio: il codice #3

```
[...]  
makebtn = wx.Button(panel, -1, "Make")  
exebtn = wx.Button(panel, -1, 'Execute')  
closebtn = wx.Button(panel, -1, "Close")  
  
actfuncList = ['Linear', 'Sigmoid', 'Sinusoidal']  
  
radioactfunc = wx.RadioButton(  
    panel, -1, "Activation function for internal layers:",  
    wx.DefaultPosition, wx.DefaultSize,  
    actfuncList, 1, wx.RA_SPECIFY_COLS  
)  
  
# bind the button events to handlers  
self.Bind(wx.EVT_BUTTON, self.OnMakeButton, makebtn)  
self.Bind(wx.EVT_BUTTON, self.OnRunButton, exebtn)  
self.Bind(wx.EVT_BUTTON, self.OnTimeToClose, closebtn)  
[...]
```

## Un esempio: il codice #4

[...]

```
sizer = wx.BoxSizer(wx.VERTICAL)
sizer.Add(text, 0, wx.ALL, 10)
sizer.Add(text1, 0, wx.ALL, 10)
sizer.Add(chlayers, 0, wx.ALL, 10)
sizer.Add(textn, 0, wx.ALL, 10)
sizer.Add(textnodes, 0, wx.ALL, 10)
sizer.Add(radioactfunc, 0, wx.ALL, 10)
sizer.Add(radioactfunclast, 0, wx.ALL, 10)
sizer.Add(makebtn, 0, wx.ALL, 10)
sizer.Add(exebtn, 0, wx.ALL, 10)
sizer.Add(closebtn, 0, wx.ALL, 10)
```

[...]

## Un esempio: il codice #5

```
[...]  
    panel.SetSizer(sizer)  
    panel.Layout()  
  
def OnRunButton(self, evt):  
    """Event handler for the button click."""  
    print "Running the main ..."  
    out_file = open("init/mlp.par","w")  
    out_file.write(MyFrame.nlayers + " !number_of_layers\n")  
    out_file.write(MyFrame.nnodes + " !nodes_on_int_layers\n")  
    out_file.write(repr(MyFrame.actfunc) + " !af\n")  
    out_file.write(repr(MyFrame.actfunclast) + " !af_last_layer\n")  
    out_file.close()  
    os.system('./main')  
  
[...]
```

## Un esempio: il codice #6

```
[...]  
class MyApp(wx.App):  
    def OnInit(self):  
        frame = MyFrame(None, "Simple wxPython App")  
        self.SetTopWindow(frame)  
  
        print "Print statements go to this stdout window by default."  
  
        frame.Show(True)  
        return True  
  
app = MyApp(redirect=True)  
app.MainLoop()  
[...]
```

# Un esempio: il risultato

**Control Panel for Fitting**

Total number of layers:

4

Number of nodes on internal layers:

10 7

Activation function for internal layers:

Linear

Sigmoid

Sinusoidal

Activation function for the last layer:

Linear

Sigmoid

Sinusoidal

Make

Execute

Close