



北京大学

ROOT的基本运行

王思广

北京大学物理学院

siguang@pku.edu.cn

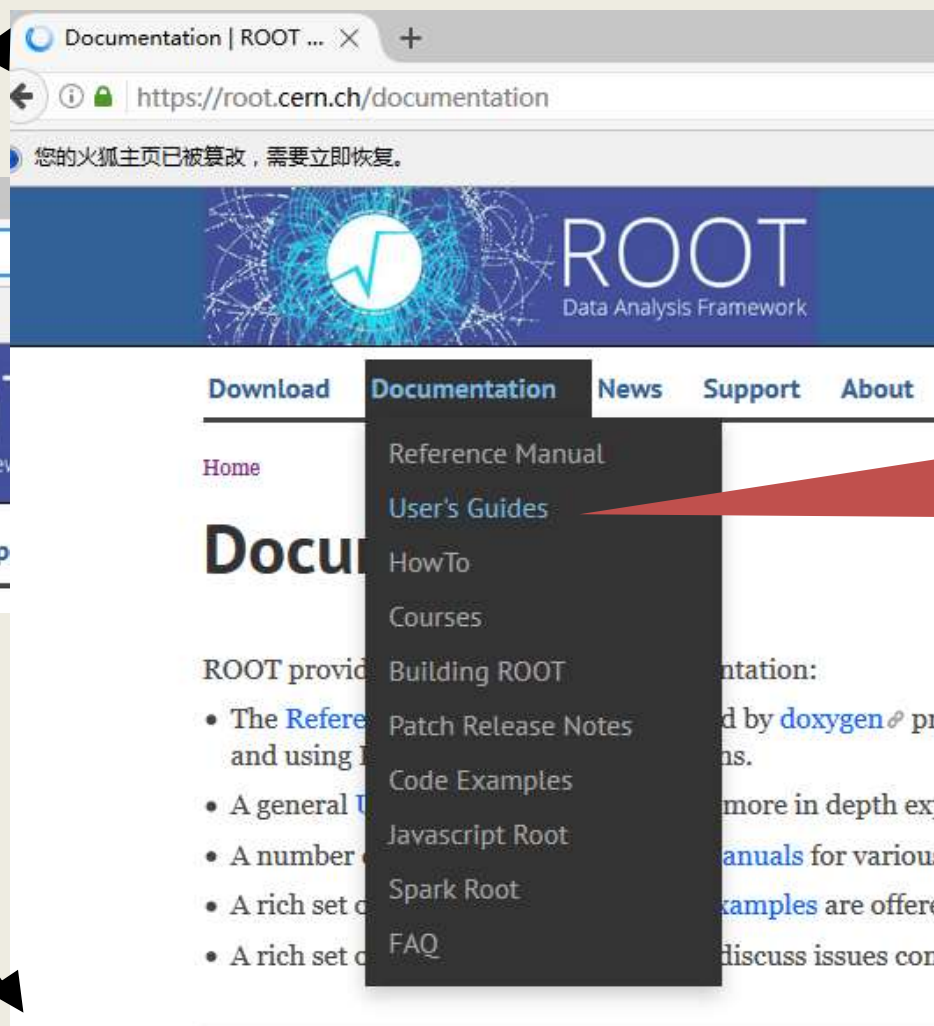
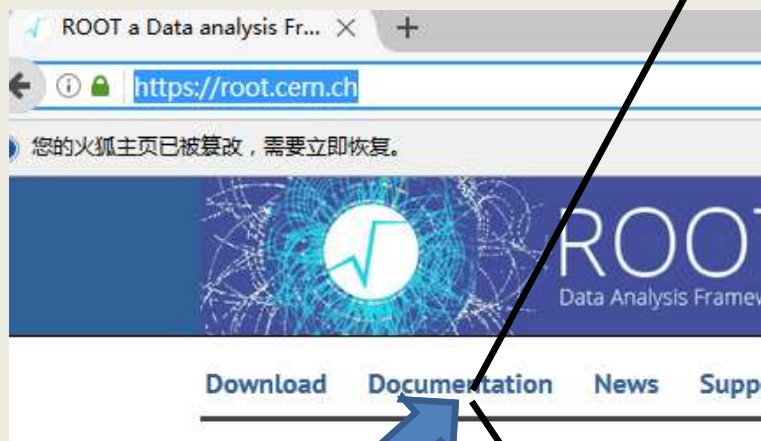
目录

- ROOT文档
- ROOT常用控制命令
- ROOT作为计算器
- ROOT作为函数画图器
- 写成代码文件运行

ROOT文档

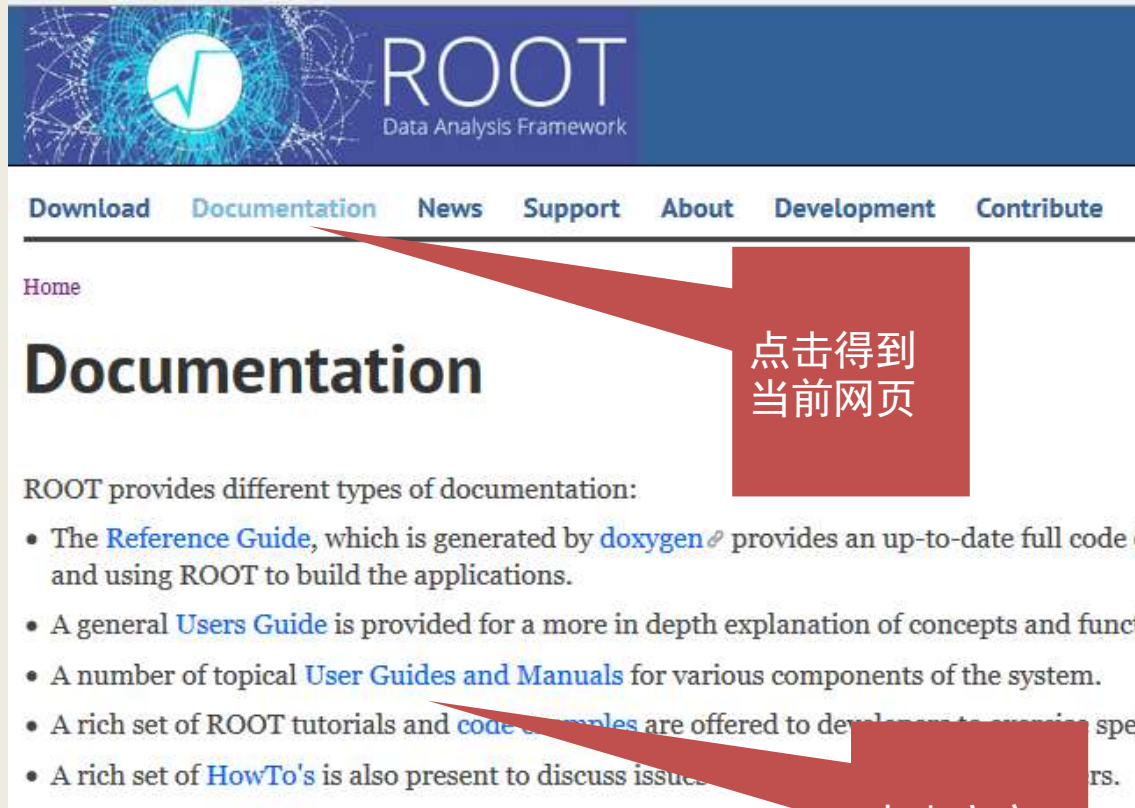
ROOT文档

■ <https://root.cern.ch/>



通过下拉菜单可以直接进入

<https://root.cern.ch/documentation>



点击得到
当前网页

点击高亮
字体进入

<https://root.cern.ch/root-user-guides-and-manuals>

ROOT
Data Analysis Framework

[Download](#) [Documentation](#) [News](#) [Support](#) [About](#) [Development](#) [Contribute](#)

[Home](#) » [Documentation](#)

ROOT User Guides and Manuals

ROOT Guides

Title	Quick Link	All Links
Reference Guide	head / 6.08	all releases
User's Guide	6 Series (html)	all formats and series
ROOT Primer	6 Series (html)	all formats and series
ROOT Primer 5	5 Series (pdf)	

ROOT5Primer.pdf

A ROOT Guide For Beginners

"Diving Into ROOT"

Abstract:

ROOT is a software framework for data analysis, a powerful tool to cope with the demanding tasks typical of state of the art scientific data analysis. Among its prominent features are an advanced graphical user interface, ideal for interactive analysis, an interpreter for the C++ programming language, for rapid and efficient prototyping and a persistency mechanism for C++ objects, used also to write every year petabytes of data recorded by the Large Hadron Collider experiments. This introductory guide illustrates the main features of ROOT, relevant for the typical problems of data analysis: input and plotting of data from measurements and fitting of analytical functions.

ROOT

An Object-Oriented
Data Analysis Framework



<https://d35c7d8c.web.cern.ch/sites/d35c7d8c.web.cern.ch/files/ROOT5Primer.pdf>

初学者以此作为参考书

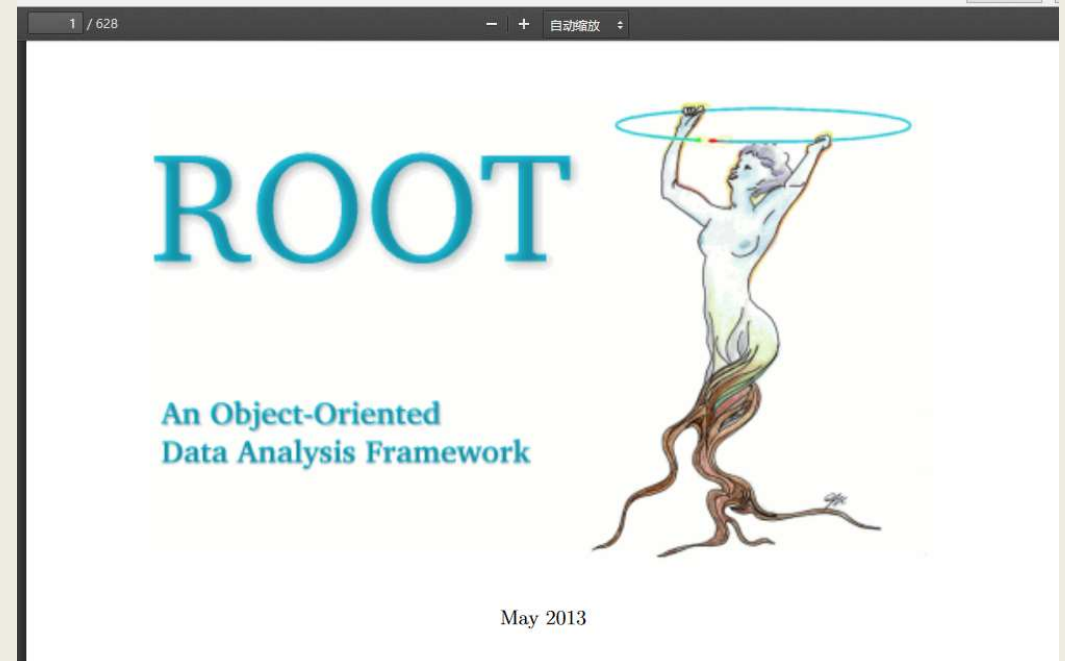
User's Guide

ROOT Old User's Guide (5.34)

ROOT User's Guide in PDF format

- [Complete User's Guide in A4 format.](#)
- [Complete User's Guide in Letter format.](#)
- Chapters:
 - [Introduction](#)
 - [Getting Started](#)
 - [Histograms](#)
 - [Graphs](#)
 - [Fitting Histograms](#)
 - [A Little C++](#)
 - [CINT](#)
 - [Object Ownership](#)
 - [Graphics](#)
 - [Folders and Tasks](#)
 - [Input/Output](#)
 - [Trees](#)
 - [Math Libraries](#)
 - [Linear Algebra](#)
 - [Adding a Class](#)

<https://root.cern.ch/guides/users-guide>



主要的参考

siguang@pku.edu.cn

Reference Guide

https://root.cern.ch/guides/reference-guide

您的火狐主页已被篡改，需要立即恢复。

Home » Documentation

Reference Guide

The Reference Guide is available for all major ROOT releases, and for the current development snapshot in subversion:


- HEAD of the git master - [browse](#)
- 6.08 - [browse](#) | [download](#)
- 6.06 - [browse](#) | [download](#)
- 6.04 - [browse](#) | [download](#)
- 6.02 - [browse](#) | [download](#)
- 5.34 - [browse](#) | [download](#)

Class Index

ps://root.cern.ch/root/html534/ClassIndex.html

您的火狐主页已被篡改，需要立即恢复。

是(R)



ROOT

Quick Links: ROOT Homepage Class Index Class Hierarchy Search documentation... Search

Class Index

Modules

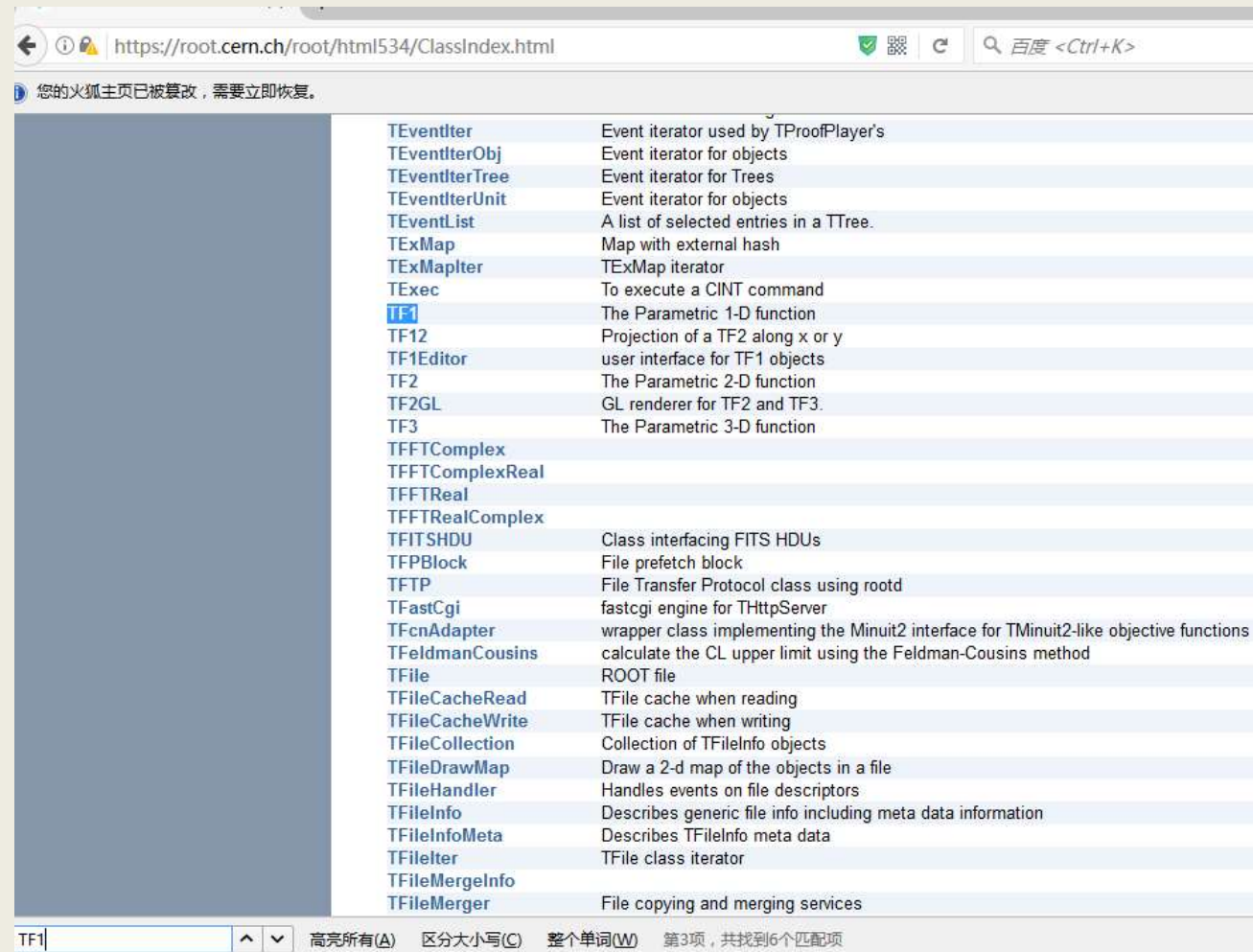
BINDINGS CINT CORE GEOM GRAF2D GRAF3D GUI HIST HTML IO MATH MISC MONTECARLO
NET PROOF ROOFIT SQL TEST TMVA TREE

Jump to

C ROOT: ROOT::Math: ROOT::Math::L ROOT::Math::P ROOT::Math::S ROOT::Math::SMatrix<f
ROOT::Math::T ROOT::T ROOT::TS Roo1 RooC RooCh RooG RooH RooS RooSt RooStats: RooU
T TB TC TD TEv TEvG TEvQ TEvW TG TGH TGL TGLP TGLW TGR TGU TGeo
TGeoN TGeoT TGu TI TMV TMVA: TMVA::V TMe TO TP TPo TPY TR TS TSp TSu TU
TV

ColorStruct_t
CpuInfo_t CPU load information.
Event_t
FileSaver

TF1函数定义网页



您的火狐主页已被篡改，需要立即恢复。

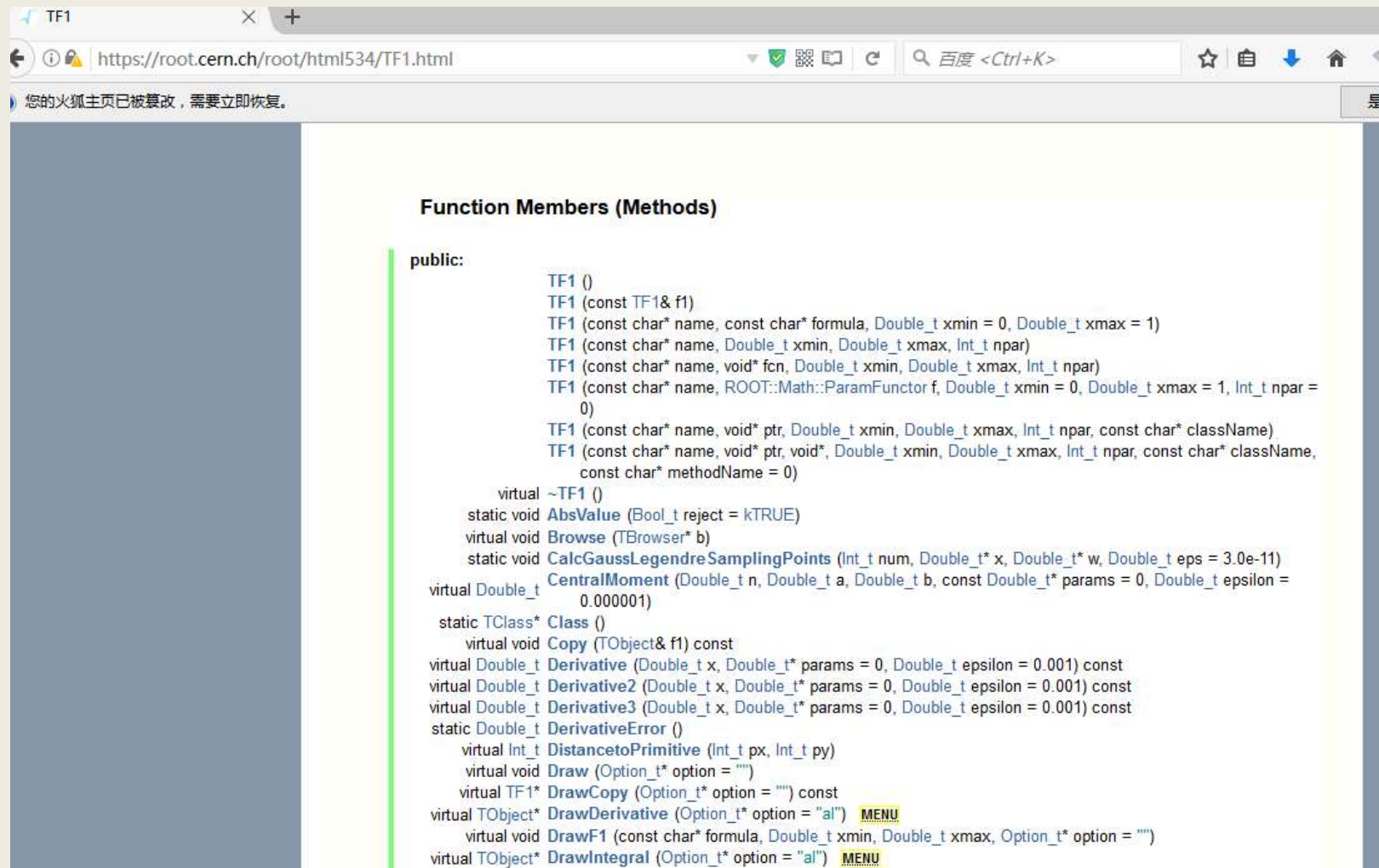
TEventIter	Event iterator used by TProofPlayer's
TEventIterObj	Event iterator for objects
TEventIterTree	Event iterator for Trees
TEventIterUnit	Event iterator for objects
TEventList	A list of selected entries in a TTree.
TExMap	Map with external hash
TExMapIter	TExMap iterator
TExec	To execute a CINT command
TF1	The Parametric 1-D function
TF12	Projection of a TF2 along x or y
TF1Editor	user interface for TF1 objects
TF2	The Parametric 2-D function
TF2GL	GL renderer for TF2 and TF3.
TF3	The Parametric 3-D function
TFFTComplex	
TFFTComplexReal	
TFFTReal	
TFFTRealComplex	
TFITSHDU	Class interfacing FITS HDUs
TFPBlock	File prefetch block
TFTP	File Transfer Protocol class using rootd
TFastCgi	fastcgi engine for THttpServer
TFcnAdapter	wrapper class implementing the Minuit2 interface for TMinuit2-like objective functions
TFeldmanCousins	calculate the CL upper limit using the Feldman-Cousins method
TFile	ROOT file
TFileCacheRead	TFile cache when reading
TFileCacheWrite	TFile cache when writing
TFileCollection	Collection of TFileInfo objects
TFileDrawMap	Draw a 2-d map of the objects in a file
TFileHandler	Handles events on file descriptors
TFileInfo	Describes generic file info including meta data information
TFileInfoMeta	Describes TFileInfo meta data
TFileIter	TFile class iterator
TFileMergeInfo	
TFileMerger	File copying and merging services

TF1 高亮所有(A) 区分大小写(C) 整个单词(W) 第3项, 共找到6个匹配项

CTRL+F 搜索:

siguang@pku.edu.cn

TF1函数定义网页



```
TF1

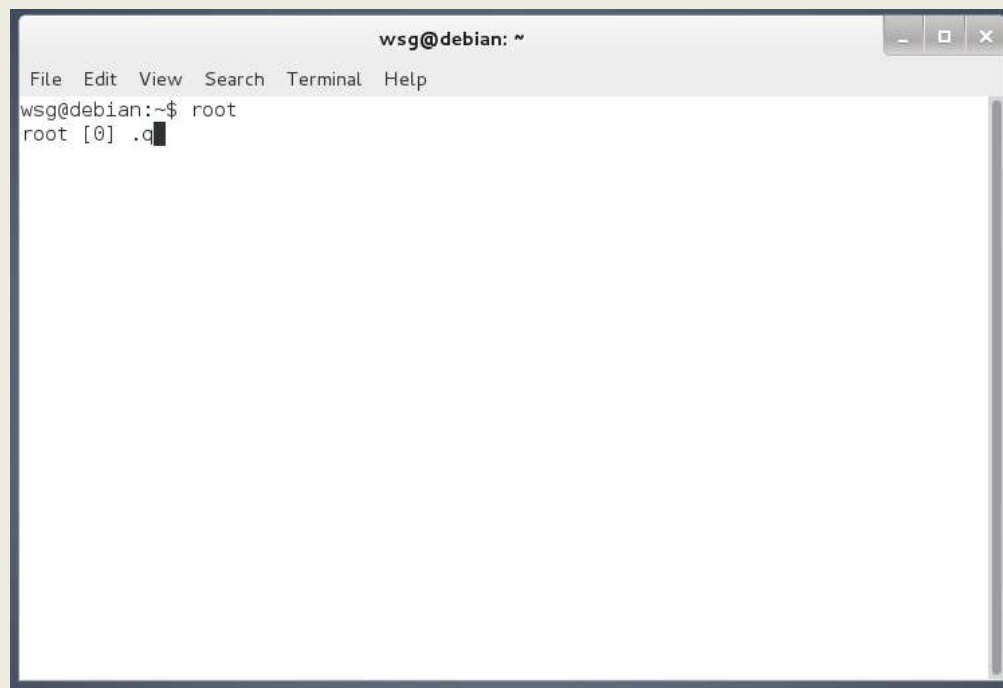
Function Members (Methods)

public:
    TF1 ()
    TF1 (const TF1& f1)
    TF1 (const char* name, const char* formula, Double_t xmin = 0, Double_t xmax = 1)
    TF1 (const char* name, Double_t xmin, Double_t xmax, Int_t npar)
    TF1 (const char* name, void* fcn, Double_t xmin, Double_t xmax, Int_t npar)
    TF1 (const char* name, ROOT::Math::ParamFuncor f, Double_t xmin = 0, Double_t xmax = 1, Int_t npar = 0)
    TF1 (const char* name, void* ptr, Double_t xmin, Double_t xmax, Int_t npar, const char* className)
    TF1 (const char* name, void* ptr, void*, Double_t xmin, Double_t xmax, Int_t npar, const char* className, const char* methodName = 0)
    virtual ~TF1 ()
    static void AbsValue (Bool_t reject = kTRUE)
    virtual void Browse (TBrowser* b)
    static void CalcGaussLegendreSamplingPoints (Int_t num, Double_t* x, Double_t* w, Double_t eps = 3.0e-11)
    virtual Double_t CentralMoment (Double_t n, Double_t a, Double_t b, const Double_t* params = 0, Double_t epsilon = 0.000001)
    static TClass* Class ()
    virtual void Copy (TObject& f1) const
    virtual Double_t Derivative (Double_t x, Double_t* params = 0, Double_t epsilon = 0.001) const
    virtual Double_t Derivative2 (Double_t x, Double_t* params = 0, Double_t epsilon = 0.001) const
    virtual Double_t Derivative3 (Double_t x, Double_t* params = 0, Double_t epsilon = 0.001) const
    static Double_t DerivativeError ()
    virtual Int_t DistancetoPrimitive (Int_t px, Int_t py)
    virtual void Draw (Option_t* option = "")
    virtual TF1* DrawCopy (Option_t* option = "") const
    virtual TObject* DrawDerivative (Option_t* option = "al") MENU
    virtual void DrawF1 (const char* formula, Double_t xmin, Double_t xmax, Option_t* option = "")
    virtual TObject* DrawIntegral (Option_t* option = "al") MENU
```

ROOT常用控制命令

启动退出ROOT

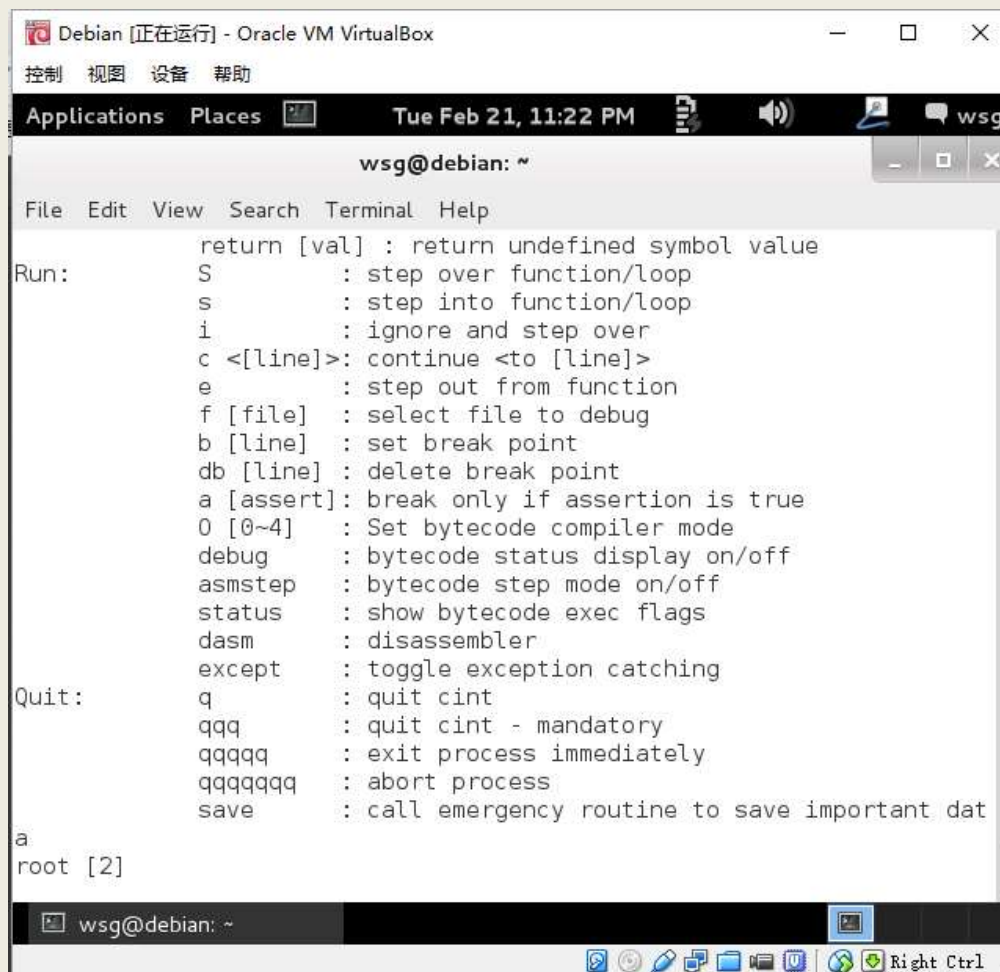
- 启动ROOT: 在任意的终端提示符下键入`root`并按回车
- 退出ROOT: `root[]`.q

A screenshot of a terminal window titled 'wsg@debian: ~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows the command 'root' being entered at the 'wsg@debian:~\$' prompt, which results in a new prompt 'root [0]'. Below this, the command '.q' is entered at the 'root [0]' prompt, and the cursor is positioned at the end of the command.

```
wsg@debian: ~  
File Edit View Search Terminal Help  
wsg@debian:~$ root  
root [0] .q
```

ROOT常用控制命令(.?将所有的命令列出来)

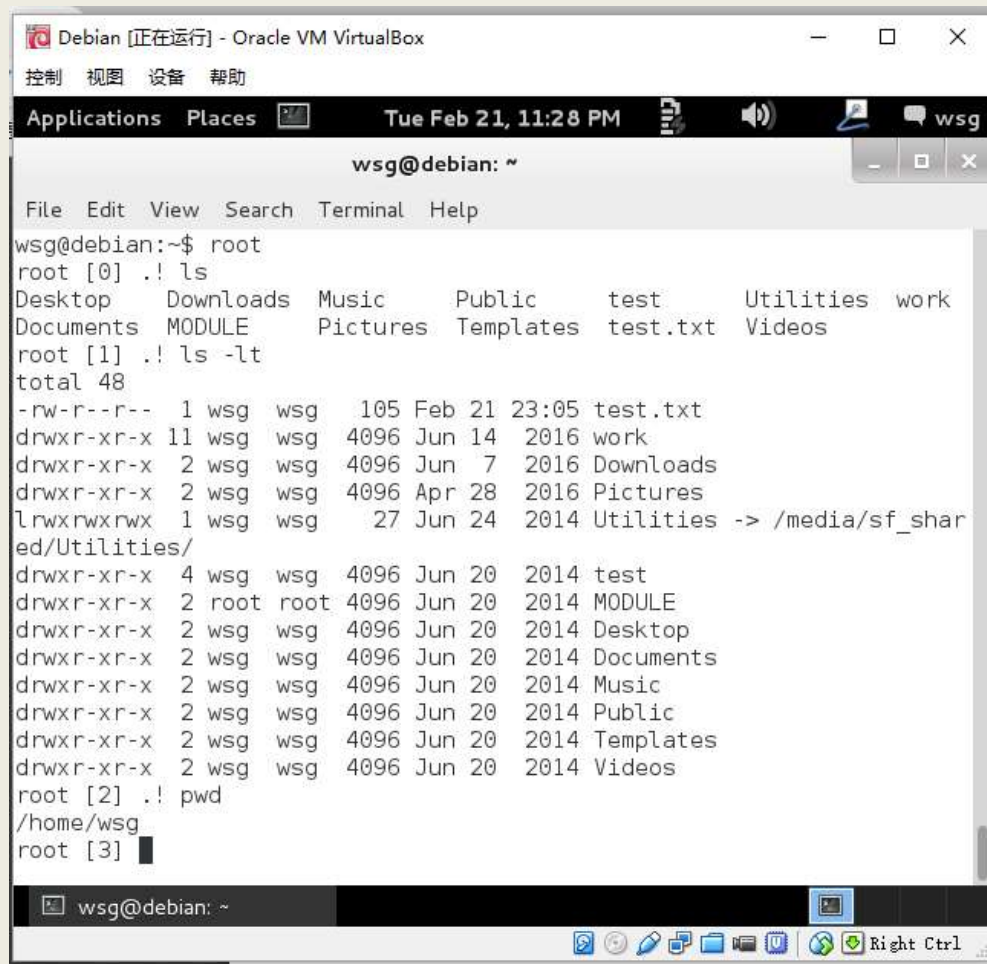
■ Root[.].?



```
Debian [正在运行] - Oracle VM VirtualBox
控制 视图 设备 帮助
Applications Places Tue Feb 21, 11:22 PM wsg
wsg@debian: ~
File Edit View Search Terminal Help
Run:      return [val] : return undefined symbol value
          S           : step over function/loop
          s           : step into function/loop
          i           : ignore and step over
          c <[line]>: continue <to [line]>
          e           : step out from function
          f [file]   : select file to debug
          b [line]   : set break point
          db [line]  : delete break point
          a [assert] : break only if assertion is true
          0 [0~4]    : Set bytecode compiler mode
          debug      : bytecode status display on/off
          asmstep    : bytecode step mode on/off
          status     : show bytecode exec flags
          dasm       : disassembler
          except     : toggle exception catching
Quit:     q         : quit cint
          qq         : quit cint - mandatory
          qqqq       : exit process immediately
          qqqqqq     : abort process
          save       : call emergency routine to save important dat
a
root [2]
```

ROOT常用控制命令(! shell 命令)

■ Root[.]! ls



The screenshot shows a terminal window titled "Debian [正在运行] - Oracle VM VirtualBox". The user "wsg" is logged in. The terminal shows the following commands and output:

```
wsg@debian:~$ root
root [0] .! ls
Desktop    Downloads Music      Public     test       Utilities  work
Documents  MODULE   Pictures  Templates test.txt   Videos
root [1] .! ls -lt
total 48
-rw-r--r--  1 wsg  wsg   105 Feb 21  23:05 test.txt
drwxr-xr-x 11 wsg  wsg  4096 Jun 14  2016 work
drwxr-xr-x  2 wsg  wsg  4096 Jun  7  2016 Downloads
drwxr-xr-x  2 wsg  wsg  4096 Apr 28  2016 Pictures
lrwxrwxrwx  1 wsg  wsg    27 Jun 24  2014 Utilities -> /media/sf_shar
ed/Utilities/
drwxr-xr-x  4 wsg  wsg  4096 Jun 20  2014 test
drwxr-xr-x  2 root root 4096 Jun 20  2014 MODULE
drwxr-xr-x  2 wsg  wsg  4096 Jun 20  2014 Desktop
drwxr-xr-x  2 wsg  wsg  4096 Jun 20  2014 Documents
drwxr-xr-x  2 wsg  wsg  4096 Jun 20  2014 Music
drwxr-xr-x  2 wsg  wsg  4096 Jun 20  2014 Public
drwxr-xr-x  2 wsg  wsg  4096 Jun 20  2014 Templates
drwxr-xr-x  2 wsg  wsg  4096 Jun 20  2014 Videos
root [2] .! pwd
/home/wsg
root [3]
```


ROOT运行常用命令（直接运行、.L 及.x 执行root程序）

以 \$ROOTSYS/tutorials/roofit/rf208_convolution.C 为例子
进入roofit目录 >cd \$ROOTSYS/tutorials/roofit

启动root的同时执行程序

- a)直接运行(解释): >root rf208_convolution.C
- b)直接运行(编译): >root rf208_convolution.C+
- c)直接运行(编译): >root rf208_convolution.C++

说明:

文件名后有1个+: 先查看当前是不是已有有编译过的库, 如果有, 将代码的最后编辑时间与库进行比较, 如果库比较新, 不编译, 直接运行; 如果代码比较新, 重新编译并运行。

文件名后有++: 直接编译并运行, 不进行与是否有编译过的库的检查。

ROOT运行常用命令（直接运行、.L 及.x 执行root程序）

以 \$ROOTSYS/tutorials/roofit/rf208_convolution.C 为例子
进入roofit目录 >`cd $ROOTSYS/tutorials/roofit`

启动root的后用.x执行程序

- a).x运行(解释): `root[] .x rf208_convolution.C`
- b).x运行(编译) : `root[] .x rf208_convolution.C+`
- c).x运行(编译) : `root[] .x rf208_convolution.C++`

ROOT运行常用命令（直接运行、.L 及.x 执行root程序）

以 \$ROOTSYS/tutorials/roofit/rf208_convolution.C 为例子
进入roofit目录 >`cd $ROOTSYS/tutorials/roofit`

启动root的后用.L执行程序

- a).L运行(解释): `root[] .L rf208_convolution.C`
`root[] rf208_convolution()`
- b).L运行(编译): `root[] .L rf208_convolution.C+`
`root[] rf208_convolution()`
- c).L运行(编译): `root[] .L rf208_convolution.C++`
`root[] rf208_convolution()`

说明:

每步运行需要两步，第一步将程序引入内存（有+或++的进行编译后引入内存），第二步调用内存中的函数。

ROOT作为计算器

ROOT作为计算器

```
root [0] 1+1
(const int)2
root [1] 2*(4+2)/12.
(const double)1.000000000000000000e+00
root [2] sqrt(3)
(const double)1.73205080756887719e+00
root [3] 1 > 2
(const int)0
root [4] TMath::Pi()
(Double_t)3.14159265358979312e+00
root [5] TMath::Erf(.2)
(Double_t)2.22702589210478447e-01
```

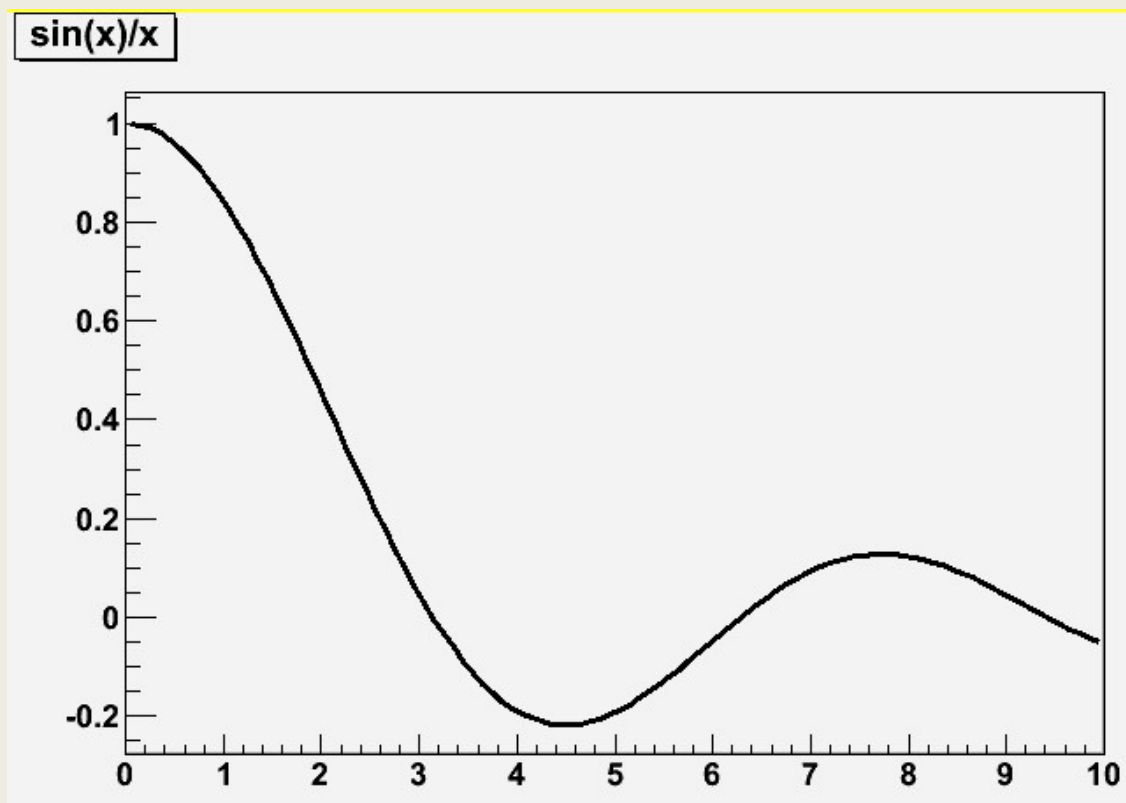
ROOT作为计算器 [续]

```
root [6] double x=.5
root [7] int N=30
root [8] double geom_series=0
root [9] for (int i=0;i<N;++i)geom_series+=TMath::Power(x,i)
root [10] TMath::Abs(geom_series - (1-TMath::Power(x,N-1))/(1-x))
(Double_t)1.86264514923095703e-09
```

ROOT作为函数画图器

ROOT作为函数画图器

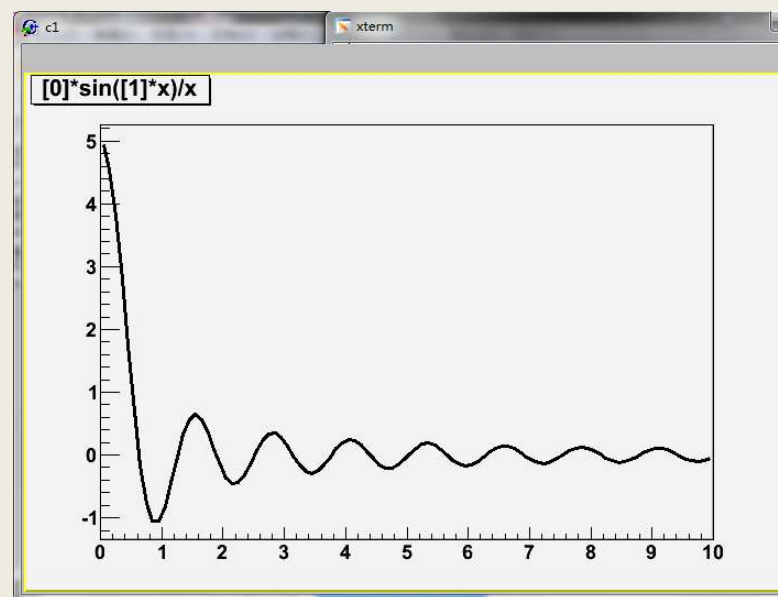
```
root [11] TF1 *f1 = new TF1("f1","sin(x)/x",0.,10.);  
root [12] f1->Draw();
```



ROOT作为函数画图器[续]

```
root [13] TF1 *f1 = new TF1("f2","[0]*sin([1]*x)/x",0.,10.);  
root [14] f1->SetParameter(0,1);  
root [15] f1->SetParameter(1,1);  
root [16] f1->Draw();
```

```
root [] f1->SetParameter(1,5);  
root [] f1->Draw()
```





北京大学

ROOT的基本运行

王思广

北京大学物理学院

siguang@pku.edu.cn

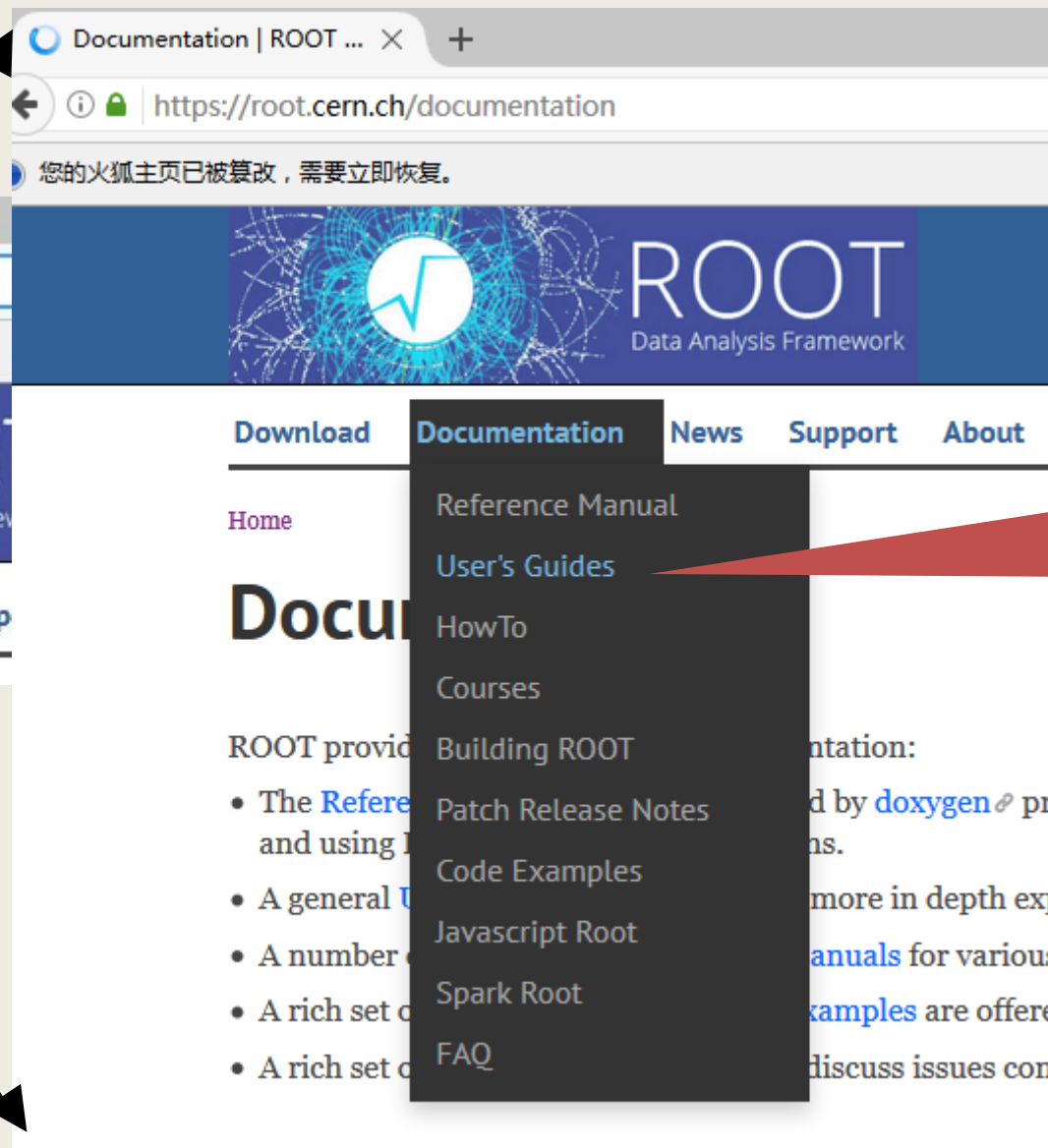
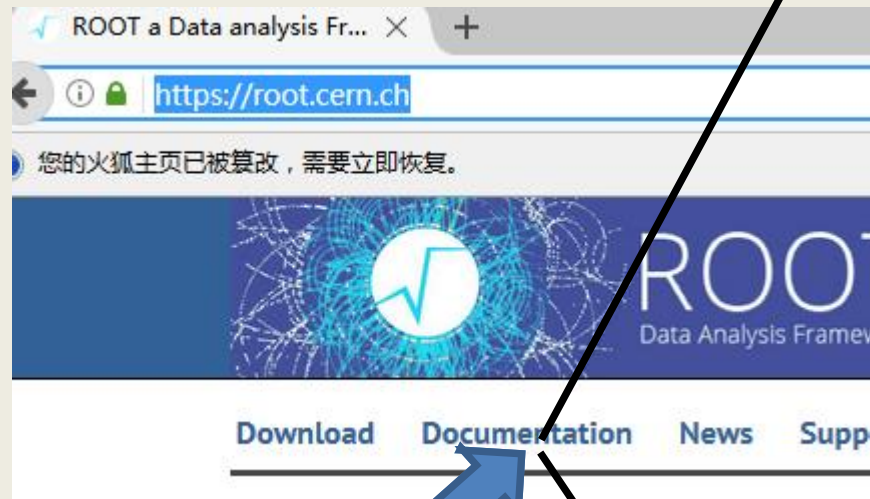
目录

- ROOT文档
- ROOT常用控制命令
- ROOT作为计算器
- ROOT作为函数画图器
- 写成代码文件运行

ROOT文档


ROOT文档

■ <https://root.cern.ch/>



通过下拉菜单可以直接进入

<https://root.cern.ch/documentation>



The screenshot shows the ROOT Data Analysis Framework documentation page. The header features the ROOT logo and navigation links: Download, Documentation, News, Support, About, Development, and Contribute. The 'Documentation' link is highlighted. Below the header, the word 'Home' is visible, followed by the title 'Documentation'. A red callout box points to the 'Documentation' link in the header with the text '点击得到当前网页'. Below the title, a paragraph states 'ROOT provides different types of documentation:'. A list of links follows: 'The Reference Guide, which is generated by doxygen', 'A general Users Guide', 'A number of topical User Guides and Manuals', 'A rich set of ROOT tutorials and code examples', and 'A rich set of HowTo's'. A second red callout box points to the 'code examples' link in the list with the text '点击高亮字体进入'.

ROOT Data Analysis Framework

Download Documentation News Support About Development Contribute

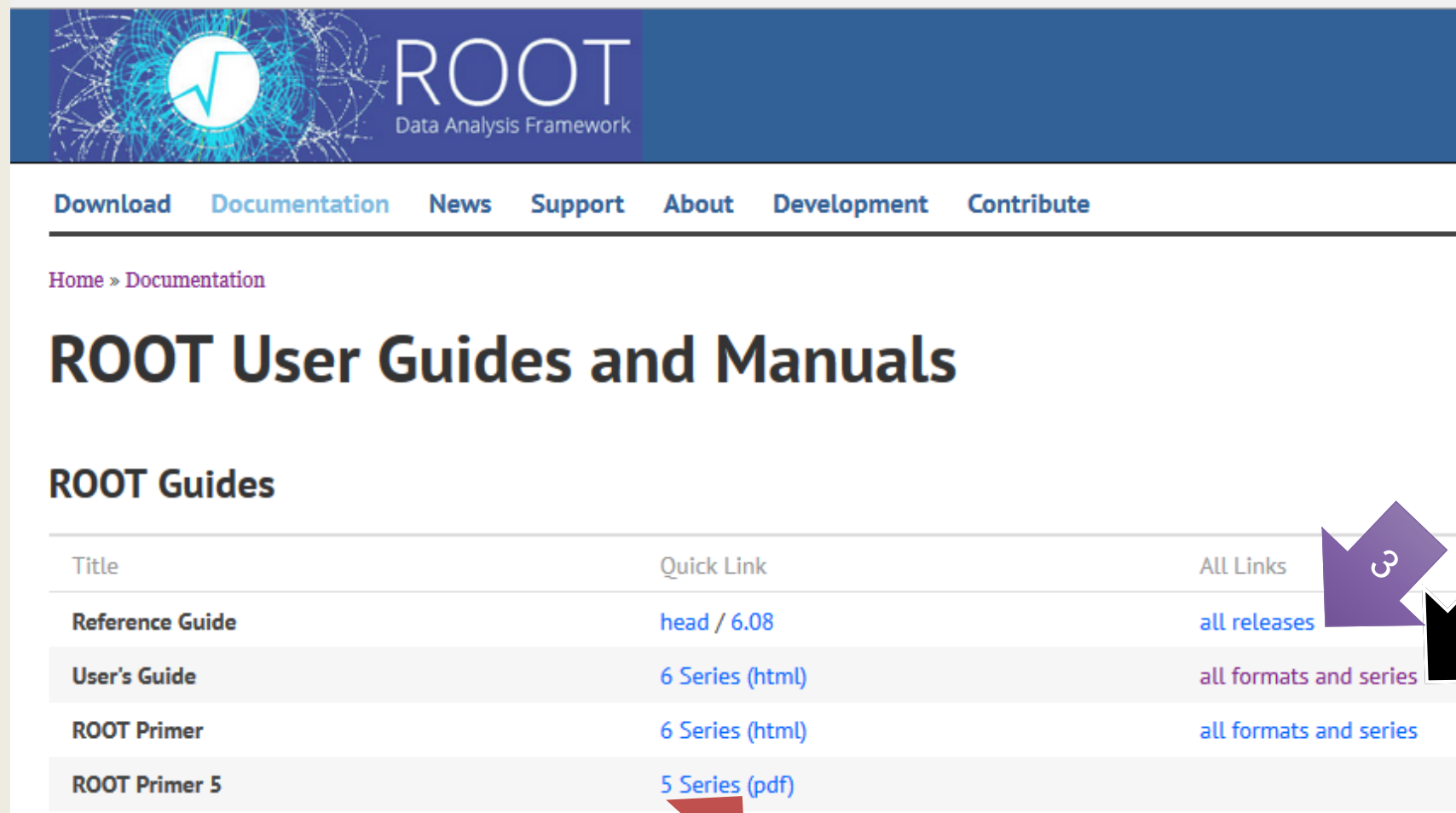
Home

Documentation

ROOT provides different types of documentation:

- The [Reference Guide](#), which is generated by [doxygen](#) provides an up-to-date full code d and using ROOT to build the applications.
- A general [Users Guide](#) is provided for a more in depth explanation of concepts and functi
- A number of topical [User Guides and Manuals](#) for various components of the system.
- A rich set of ROOT tutorials and [code examples](#) are offered to develop your own spec
- A rich set of [HowTo's](#) is also present to discuss issues

https://root.cern.ch/root-user-guides-and-manuals



The screenshot shows the ROOT Data Analysis Framework website. The header includes the ROOT logo and navigation links: Download, Documentation, News, Support, About, Development, and Contribute. The breadcrumb trail is Home » Documentation. The main heading is 'ROOT User Guides and Manuals'. Below this is a section titled 'ROOT Guides' containing a table with three columns: Title, Quick Link, and All Links. The table lists five items: Reference Guide, User's Guide, ROOT Primer, and ROOT Primer 5 (each with 6 series), and ROOT Primer 5 (with 5 series). Numbered arrows are overlaid on the image: a red arrow labeled '1' points to the '5 Series (pdf)' link for ROOT Primer 5; a purple arrow labeled '3' points to the 'all releases' link; and a black arrow labeled '2' points to the 'all formats and series' link for the User's Guide.

Title	Quick Link	All Links
Reference Guide	head / 6.08	all releases
User's Guide	6 Series (html)	all formats and series
ROOT Primer	6 Series (html)	all formats and series
ROOT Primer 5	5 Series (pdf)	

ROOT5Primer.pdf

A ROOT Guide For Beginners

“Diving Into ROOT”

Abstract:

ROOT is a software framework for data analysis, a powerful tool to cope with the demanding tasks typical of state of the art scientific data analysis. Among its prominent features are an advanced graphical user interface, ideal for interactive analysis, an interpreter for the C++ programming language, for rapid and efficient prototyping and a persistency mechanism for C++ objects, used also to write every year petabytes of data recorded by the Large Hadron Collider experiments. This introductory guide illustrates the main features of ROOT, relevant for the typical problems of data analysis: input and plotting of data from measurements and fitting of analytical functions.

ROOT

An Object-Oriented
Data Analysis Framework



<https://d35c7d8c.web.cern.ch/sites/d35c7d8c.web.cern.ch/files/ROOT5Primer.pdf>

初学者以此作为参考书

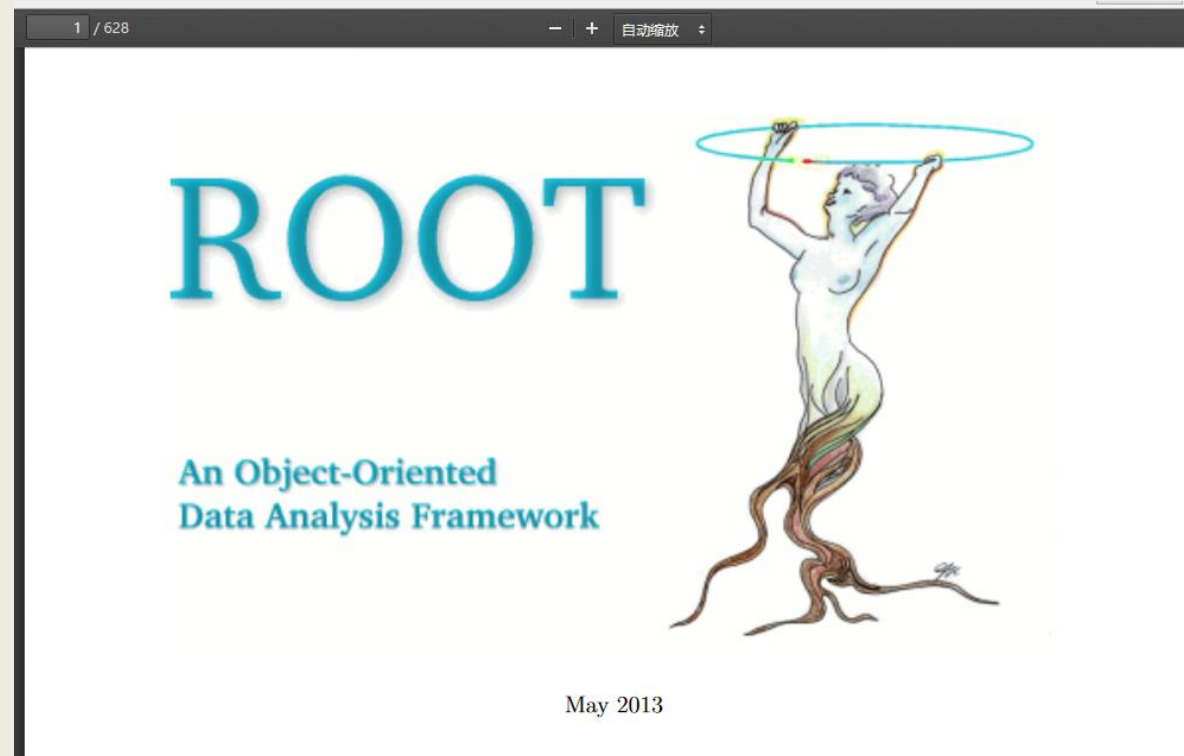
User's Guide

ROOT Old User's Guide (5.34)

ROOT User's Guide in PDF format

- [Complete User's Guide in A4 format.](#)
- [Complete User's Guide in Letter format.](#)
- Chapters:
 - a. [Introduction](#)
 - b. [Getting Started](#)
 - c. [Histograms](#)
 - d. [Graphs](#)
 - e. [Fitting Histograms](#)
 - f. [A Little C++](#)
 - g. [CINT](#)
 - h. [Object Ownership](#)
 - i. [Graphics](#)
 - j. [Folders and Tasks](#)
 - k. [Input/Output](#)
 - l. [Trees](#)
 - m. [Math Libraries](#)
 - n. [Linear Algebra](#)
 - o. [Adding a Class](#)

<https://root.cern.ch/guides/users-guide>



主要的参考

Reference Guide

您的火狐主页已被篡改，需要立即恢复。

[Home](#) » [Documentation](#)

Reference Guide

The Reference Guide is available for all major ROOT releases, and for the current development snapshot in subversion:

- **HEAD** of the git master - [browse](#)
- **6.08** - [browse](#) | [download](#)
- **6.06** - [browse](#) | [download](#)
- **6.04** - [browse](#) | [download](#)
- **6.02** - [browse](#) | [download](#)
- **5.34** - [browse](#) | [download](#)

Class Index

您的火狐主页已被篡改，需要立即恢复。

ROOT

Class Index

Modules

BINDINGS CINT CORE GEOM GRAF2D GRAF3D GUI HIST HTML IO MATH MISC MONTECARLO
NET PROOF ROOFIT SQL TEST TMVA TREE

Jump to

C ROOT: ROOT::Math: ROOT::Math::L ROOT::Math::P ROOT::Math::S ROOT::Math::SMatrix<f
ROOT::Math::T ROOT::T ROOT::TS Roo1 RooC RooCh RooG RooN RooS RooSt RooStats: RooU
T TB TC TD TEv TEveG TEveQ TEveW TG TGH TGL TGLP TGLW TGR TGU TGeo
TGeoN TGeoT TGu TI TMV TMVA: TMVA::V TMe TO TP TPo TPpy TR TS TSp TSu TU
TW

ColorStruct_t
CpuInfo_t CPU load information.
Event_t
FileSect_t

TF1函数定义网页

您的火狐主页已被篡改，需要立即恢复。

TEventIter	Event iterator used by TProofPlayer's
TEventIterObj	Event iterator for objects
TEventIterTree	Event iterator for Trees
TEventIterUnit	Event iterator for objects
TEventList	A list of selected entries in a TTree.
TExMap	Map with external hash
TExMapIter	TExMap iterator
TExec	To execute a CINT command
TF1	The Parametric 1-D function
TF12	Projection of a TF2 along x or y
TF1Editor	user interface for TF1 objects
TF2	The Parametric 2-D function
TF2GL	GL renderer for TF2 and TF3.
TF3	The Parametric 3-D function
TFFTComplex	
TFFTComplexReal	
TFFTReal	
TFFTRealComplex	
TFITSHDU	Class interfacing FITS HDUs
TFPBlock	File prefetch block
TFTP	File Transfer Protocol class using rootd
TFastCgi	fastcgi engine for THttpServer
TFcnAdapter	wrapper class implementing the Minuit2 interface for TMinuit2-like objective functions
TFeldmanCousins	calculate the CL upper limit using the Feldman-Cousins method
TFile	ROOT file
TFileCacheRead	TFile cache when reading
TFileCacheWrite	TFile cache when writing
TFileCollection	Collection of TFileInfo objects
TFileDrawMap	Draw a 2-d map of the objects in a file
TFileHandler	Handles events on file descriptors
TFileInfo	Describes generic file info including meta data information
TFileInfoMeta	Describes TFileInfo meta data
TFileIter	TFile class iterator
TFileMergeInfo	
TFileMerger	File copying and merging services

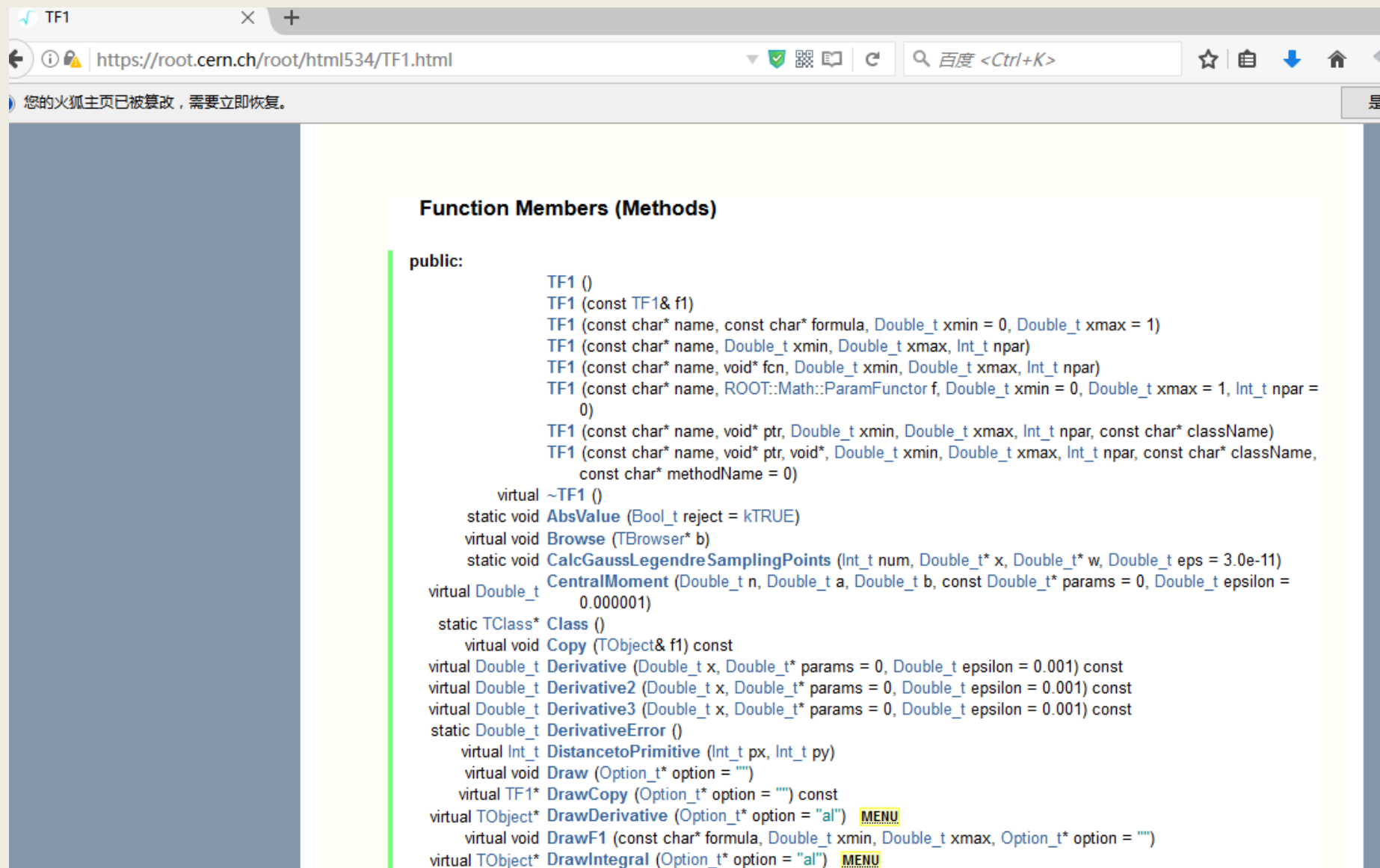
TF1

高亮所有(A) 区分大小写(C) 整个单词(W) 第3项, 共找到6个匹配项

CTRL+F 搜索:



TF1函数定义网页



```
TF1

Function Members (Methods)

public:

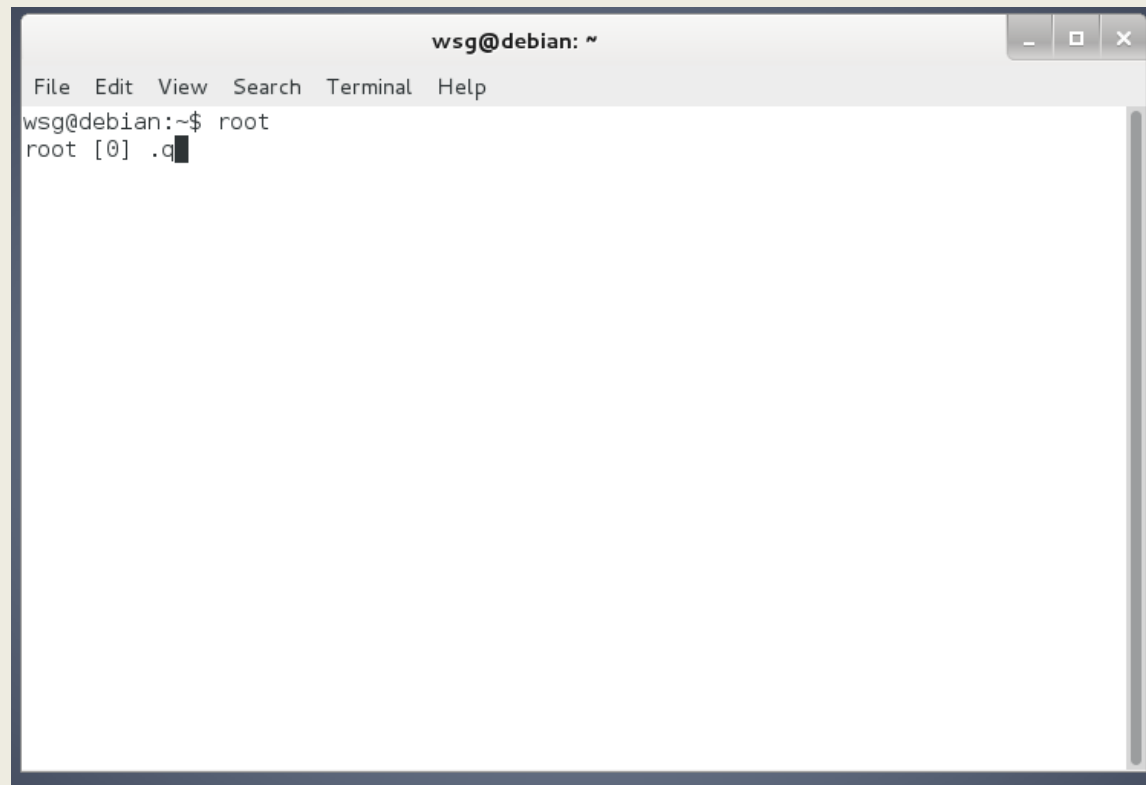
    TF1 ()
    TF1 (const TF1& f1)
    TF1 (const char* name, const char* formula, Double_t xmin = 0, Double_t xmax = 1)
    TF1 (const char* name, Double_t xmin, Double_t xmax, Int_t npar)
    TF1 (const char* name, void* fcn, Double_t xmin, Double_t xmax, Int_t npar)
    TF1 (const char* name, ROOT::Math::ParamFuncor f, Double_t xmin = 0, Double_t xmax = 1, Int_t npar = 0)
    TF1 (const char* name, void* ptr, Double_t xmin, Double_t xmax, Int_t npar, const char* className)
    TF1 (const char* name, void* ptr, void*, Double_t xmin, Double_t xmax, Int_t npar, const char* className, const char* methodName = 0)

    virtual ~TF1 ()
    static void AbsValue (Bool_t reject = kTRUE)
    virtual void Browse (TBrowser* b)
    static void CalcGaussLegendreSamplingPoints (Int_t num, Double_t* x, Double_t* w, Double_t eps = 3.0e-11)
    virtual Double_t CentralMoment (Double_t n, Double_t a, Double_t b, const Double_t* params = 0, Double_t epsilon = 0.000001)
    static TClass* Class ()
    virtual void Copy (TObject& f1) const
    virtual Double_t Derivative (Double_t x, Double_t* params = 0, Double_t epsilon = 0.001) const
    virtual Double_t Derivative2 (Double_t x, Double_t* params = 0, Double_t epsilon = 0.001) const
    virtual Double_t Derivative3 (Double_t x, Double_t* params = 0, Double_t epsilon = 0.001) const
    static Double_t DerivativeError ()
    virtual Int_t DistancetoPrimitive (Int_t px, Int_t py)
    virtual void Draw (Option_t* option = "")
    virtual TF1* DrawCopy (Option_t* option = "") const
    virtual TObject* DrawDerivative (Option_t* option = "al") MENU
    virtual void DrawF1 (const char* formula, Double_t xmin, Double_t xmax, Option_t* option = "")
    virtual TObject* DrawIntegral (Option_t* option = "al") MENU
```

ROOT常用控制命令

启动退出ROOT

- 启动ROOT：在任意的终端提示符下键入`root`并按回车
- 退出ROOT：`root[]`.q

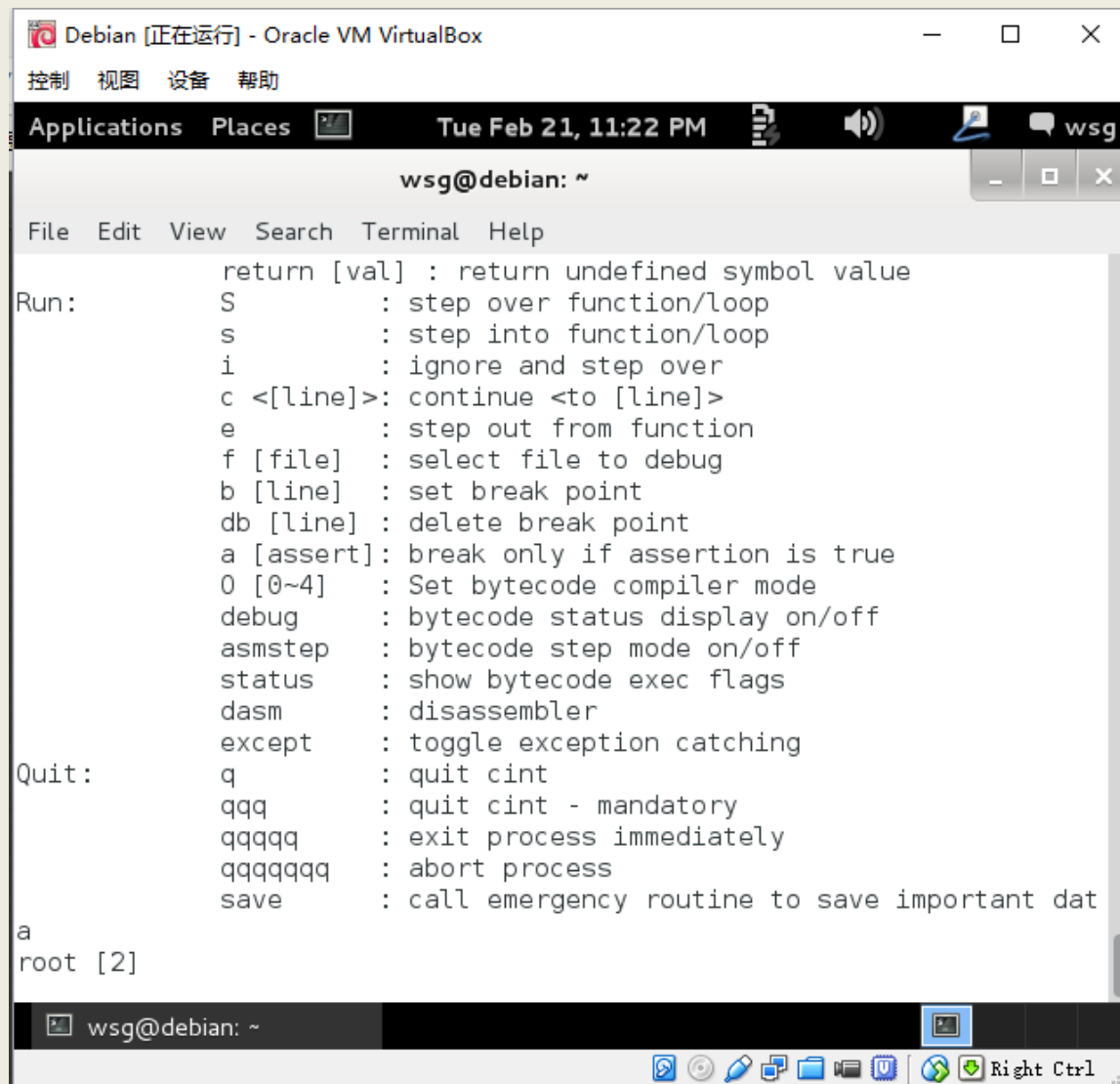


A terminal window titled "wsg@debian: ~" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command "root" being entered and executed, resulting in a prompt change from "wsg@debian:~\$" to "root [0] .q".

```
wsg@debian: ~  
File Edit View Search Terminal Help  
wsg@debian:~$ root  
root [0] .q
```

ROOT常用控制命令(.?将所有的命令列出来)

■ Root[.].?

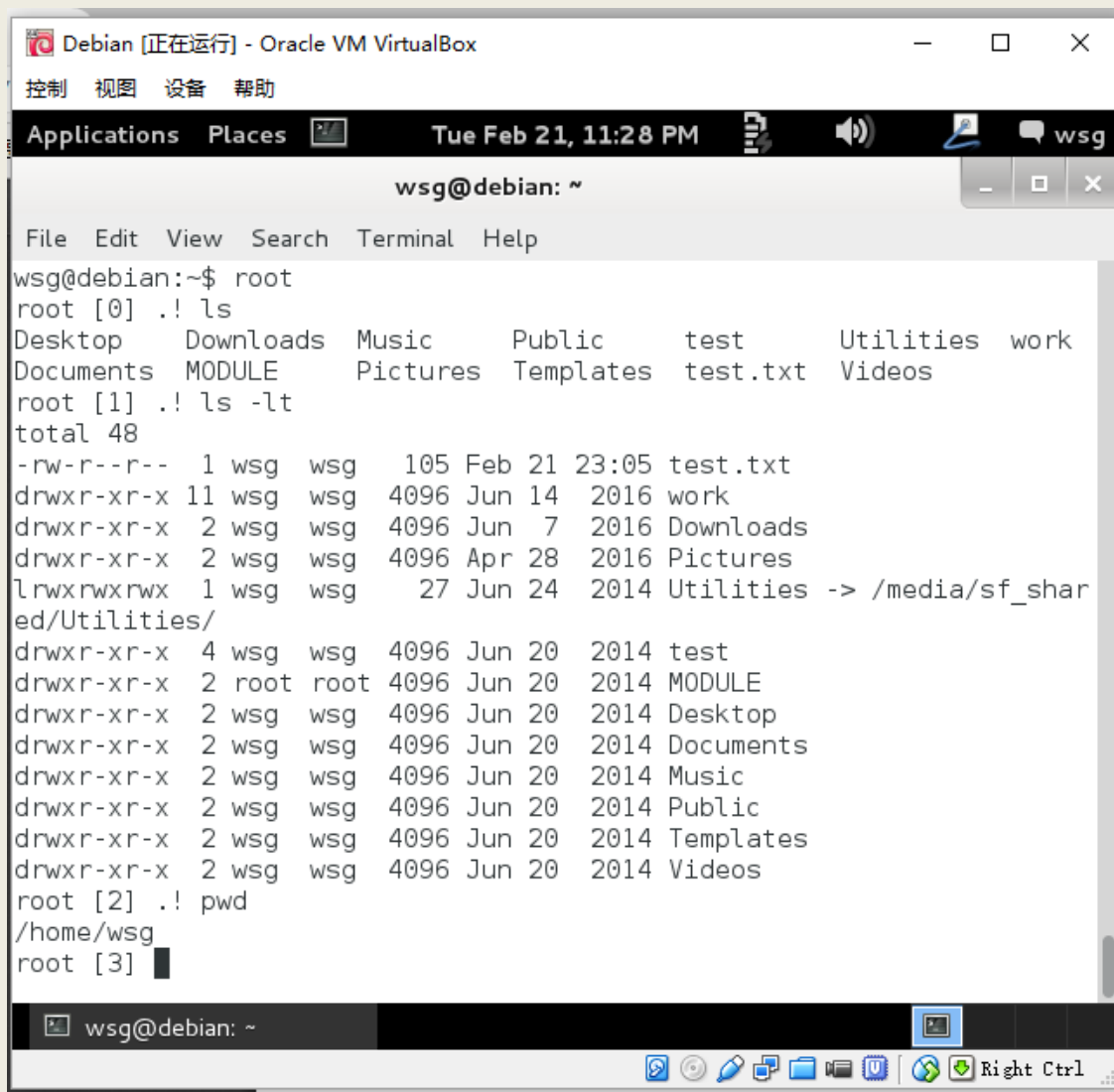


The screenshot shows a terminal window titled "Debian [正在运行] - Oracle VM VirtualBox". The window has a menu bar with "控制", "视图", "设备", and "帮助". Below the menu bar is a status bar showing "Applications", "Places", "Tue Feb 21, 11:22 PM", and a user icon labeled "wsg". The terminal itself has a title bar "wsg@debian: ~" and a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content displays the help text for the "root" command, listing various debugging and control options. At the bottom of the terminal, the prompt "a root [2]" is visible.

```
return [val] : return undefined symbol value
Run:
  S          : step over function/loop
  s          : step into function/loop
  i          : ignore and step over
  c <[line]> : continue <to [line]>
  e          : step out from function
  f [file]   : select file to debug
  b [line]   : set break point
  db [line]  : delete break point
  a [assert] : break only if assertion is true
  O [0~4]    : Set bytecode compiler mode
  debug      : bytecode status display on/off
  asmstep    : bytecode step mode on/off
  status     : show bytecode exec flags
  dasm       : disassembler
  except     : toggle exception catching
Quit:
  q          : quit cint
  qq         : quit cint - mandatory
  qqqq      : exit process immediately
  qqqqqq    : abort process
  save      : call emergency routine to save important dat
a
root [2]
```

ROOT常用控制命令(! shell 命令)

■ Root[.]! ls



```
Debian [正在运行] - Oracle VM VirtualBox
控制 视图 设备 帮助
Applications Places Tue Feb 21, 11:28 PM wsg
wsg@debian: ~
File Edit View Search Terminal Help
wsg@debian:~$ root
root [0] .! ls
Desktop Downloads Music Public test Utilities work
Documents MODULE Pictures Templates test.txt Videos
root [1] .! ls -lt
total 48
-rw-r--r-- 1 wsg wsg 105 Feb 21 23:05 test.txt
drwxr-xr-x 11 wsg wsg 4096 Jun 14 2016 work
drwxr-xr-x 2 wsg wsg 4096 Jun 7 2016 Downloads
drwxr-xr-x 2 wsg wsg 4096 Apr 28 2016 Pictures
lrwxrwxrwx 1 wsg wsg 27 Jun 24 2014 Utilities -> /media/sf_shar
ed/Utilities/
drwxr-xr-x 4 wsg wsg 4096 Jun 20 2014 test
drwxr-xr-x 2 root root 4096 Jun 20 2014 MODULE
drwxr-xr-x 2 wsg wsg 4096 Jun 20 2014 Desktop
drwxr-xr-x 2 wsg wsg 4096 Jun 20 2014 Documents
drwxr-xr-x 2 wsg wsg 4096 Jun 20 2014 Music
drwxr-xr-x 2 wsg wsg 4096 Jun 20 2014 Public
drwxr-xr-x 2 wsg wsg 4096 Jun 20 2014 Templates
drwxr-xr-x 2 wsg wsg 4096 Jun 20 2014 Videos
root [2] .! pwd
/home/wsg
root [3]
```

ROOT运行常用命令（直接运行、.L 及.x 执行root程序）

以 \$ROOTSYS/tutorials/roofit/rf208_convolution.C 为例子
进入roofit目录 >`cd $ROOTSYS/tutorials/roofit`

启动root的同时执行程序

- a)直接运行(解释): >`root rf208_convolution.C`
- b)直接运行(编译): >`root rf208_convolution.C+`
- c)直接运行(编译): >`root rf208_convolution.C++`

说明:

文件名后有1个+: 先查看当前是不是已有有编译过的库, 如果有, 将代码的最后编辑时间与库进行比较, 如果库比较新, 不编译, 直接运行; 如果代码比较新, 重新编译并运行。

文件名后有++: 直接编译并运行, 不进行与是否有编译过的库的检查。

ROOT运行常用命令（直接运行、.L 及.x 执行root程序）

以 \$ROOTSYS/tutorials/roofit/rf208_convolution.C 为例子
进入roofit目录 >`cd $ROOTSYS/tutorials/roofit`

启动root的后用.x执行程序

- a).x运行(解释): `root[] .x rf208_convolution.C`
- b).x运行(编译) : `root[] .x rf208_convolution.C+`
- c).x运行(编译) : `root[] .x rf208_convolution.C++`

ROOT运行常用命令（直接运行、.L 及.x 执行root程序）

以 \$ROOTSYS/tutorials/roofit/rf208_convolution.C 为例子
进入roofit目录 >`cd $ROOTSYS/tutorials/roofit`

启动root的后用.L执行程序

- a).L运行(解释): `root[] .L rf208_convolution.C`
`root[] rf208_convolution()`
- b).L运行(编译): `root[] .L rf208_convolution.C+`
`root[] rf208_convolution()`
- c).L运行(编译): `root[] .L rf208_convolution.C++`
`root[] rf208_convolution()`

说明:

每步运行需要两步，第一步将程序引入内存（有+或++的进行编译后引入内存），第二步调用内存中的函数。

ROOT作为计算器

ROOT作为计算器

```
root [0] 1+1
(const int)2
root [1] 2*(4+2)/12.
(const double)1.000000000000000000e+00
root [2] sqrt(3)
(const double)1.73205080756887719e+00
root [3] 1 > 2
(const int)0
root [4] TMath::Pi()
(Double_t)3.14159265358979312e+00
root [5] TMath::Erf(.2)
(Double_t)2.22702589210478447e-01
```

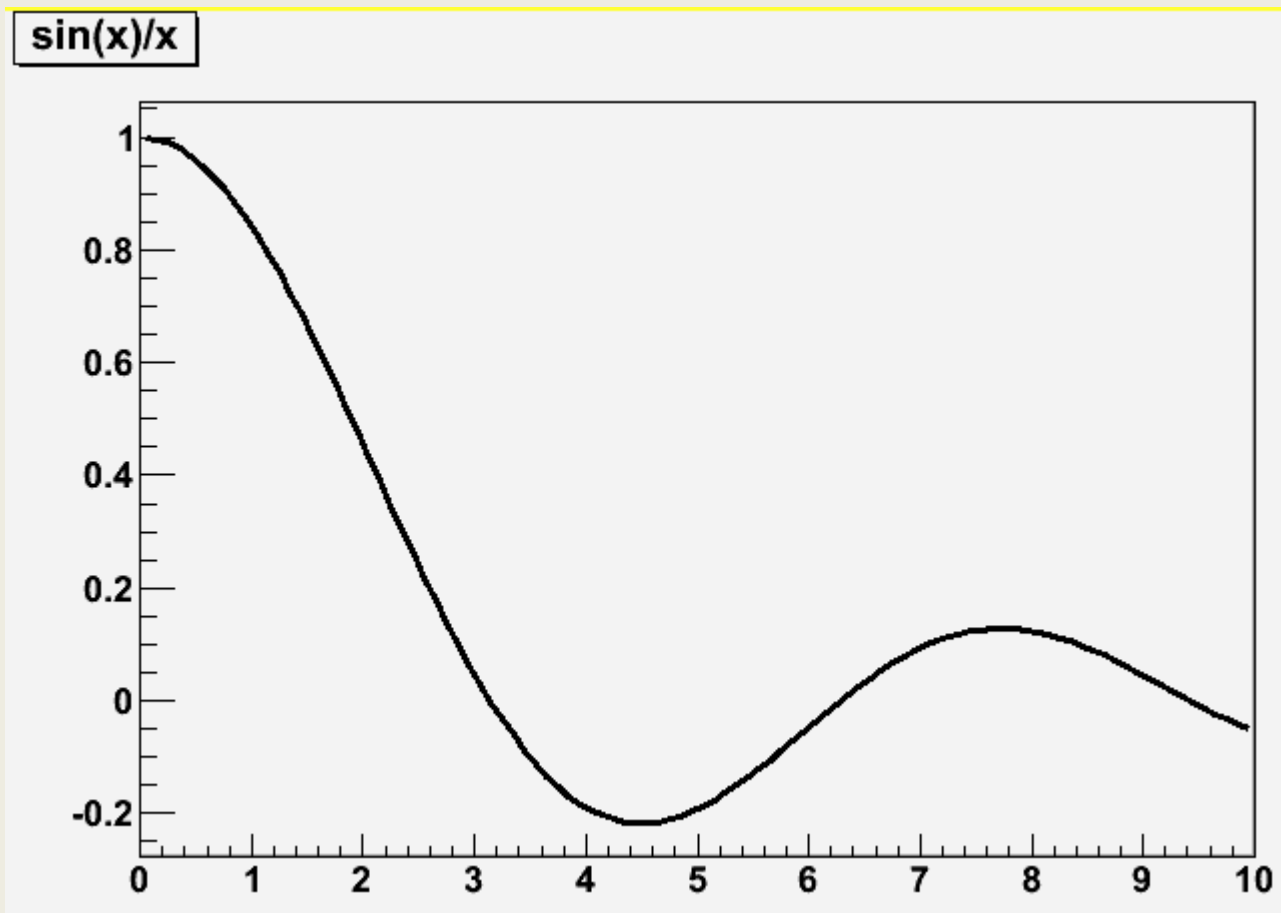
ROOT作为计算器 [续]

```
root [6] double x=.5
root [7] int N=30
root [8] double geom_series=0
root [9] for (int i=0;i<N;++i)geom_series+=TMath::Power(x,i)
root [10] TMath::Abs(geom_series - (1-TMath::Power(x,N-1))/(1-x))
(Double_t)1.86264514923095703e-09
```

ROOT作为函数画图器

ROOT作为函数画图器

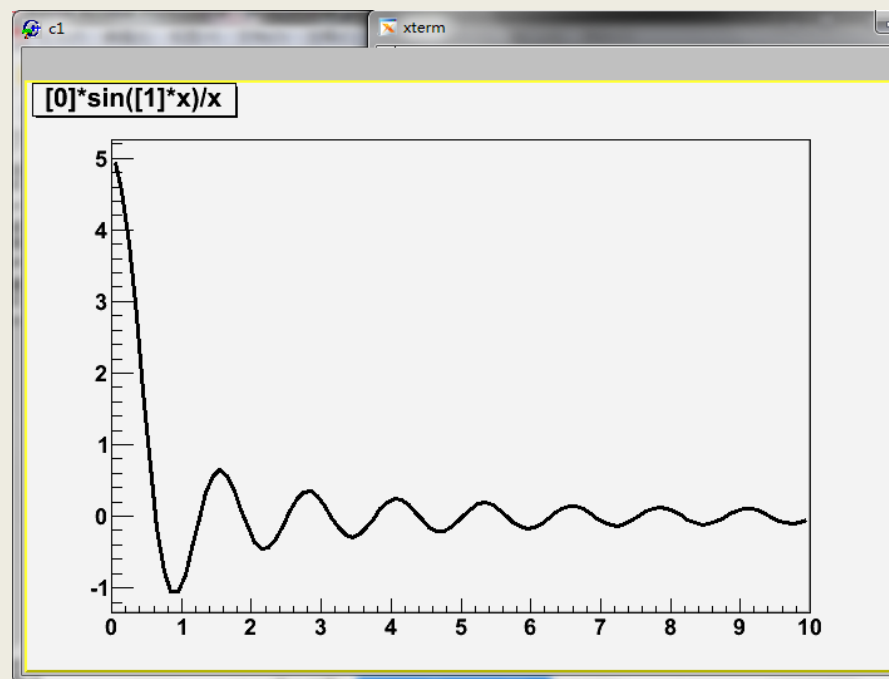
```
root [11] TF1 *f1 = new TF1("f1","sin(x)/x",0.,10.);  
root [12] f1->Draw();
```



ROOT作为函数画图器[续]

```
root [13] TF1 *f1 = new TF1("f2","[0]*sin([1]*x)/x",0.,10.);  
root [14] f1->SetParameter(0,1);  
root [15] f1->SetParameter(1,1);  
root [16] f1->Draw();
```

```
root [] f1->SetParameter(1,5);  
root [] f1->Draw()
```



写成代码文件运行

编辑并保存任意名字，例如slits.C

```
// Example drawing the interference pattern of light
// falling on a grid with n slits and ratio r of slit
// width over distance between slits.

// function code in C
double single(double *x, double *par) {
    double const pi=4*atan(1.);
    return pow(sin(pi*par[0]*x[0])/(pi*par[0]*x[0]),2);
}

double nslit0(double *x,double *par){
    double const pi=4*atan(1.);
    return pow(sin(pi*par[1]*x[0])/sin(pi*x[0]),2);
}

double nslit(double *x, double *par){
    return single(x,par) * nslit0(x,par);
}

// This is the main program
void slits() {
    float r,ns;
    // request user input
    cout << "slit width: r=0.2? ";
    scanf("%f",&r);
    cout << "# of slits ns=2?";
    scanf("%f",&ns);
    cout <<"interference pattern for "<< ns
        <<" slits, width/distance: "<<r<<endl;

    // define function and set options
    TF1 *Fnslit = new TF1("Fnslit",nslit,-5.001,5.,2);
    Fnslit->SetNpx(500);
```

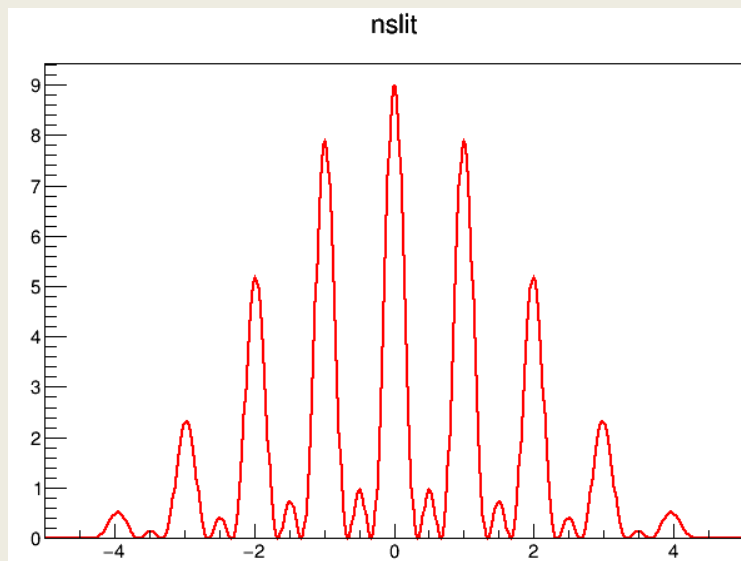
slits.C [续]

```
// set parameters, as read in above
Fnslit->SetParameter(0,r);
Fnslit->SetParameter(1,ns);

// draw the interference pattern for a grid with n slits
Fnslit->Draw();
}
```

主函数名字如果与文件名字同，
直接终端提示符下运行：root slits.C
或
Root[].x slits.C
程序自动运行slits.C中的slits()函数

```
root [0] .x slits.C
slit width: r=0.2? 0.2
# of slits ns=2?3
interference pattern for 3 slits, width/distance: 0.2
Info in <TCanvas::MakeDefCanvas>: created default TCanvas with name c1
root [1] █
```



练习：

- 1) 在上面画箭头然后保存成c1.C文件；保存成.eps 等文件
- 2) 增加坐标轴的标题
- 3) 在TCanvas上用鼠标右键看看有啥反应？！
可以设线，点的属性，坐标轴的Title等。。。。。