北京大學

ROOT的基本运行

王思广
北京大学物理学院
siguang@pku.edu.cn

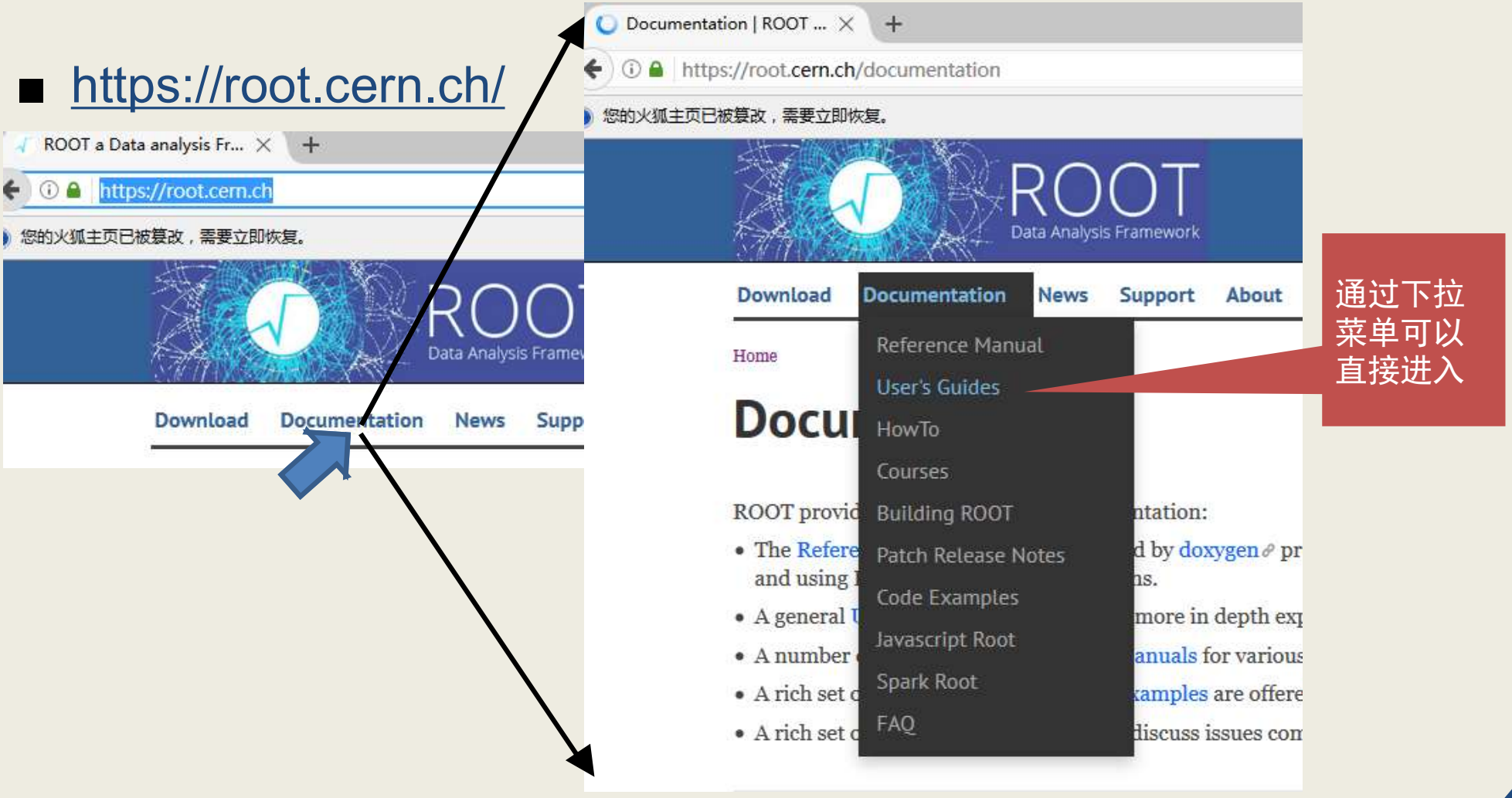
目录

- 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

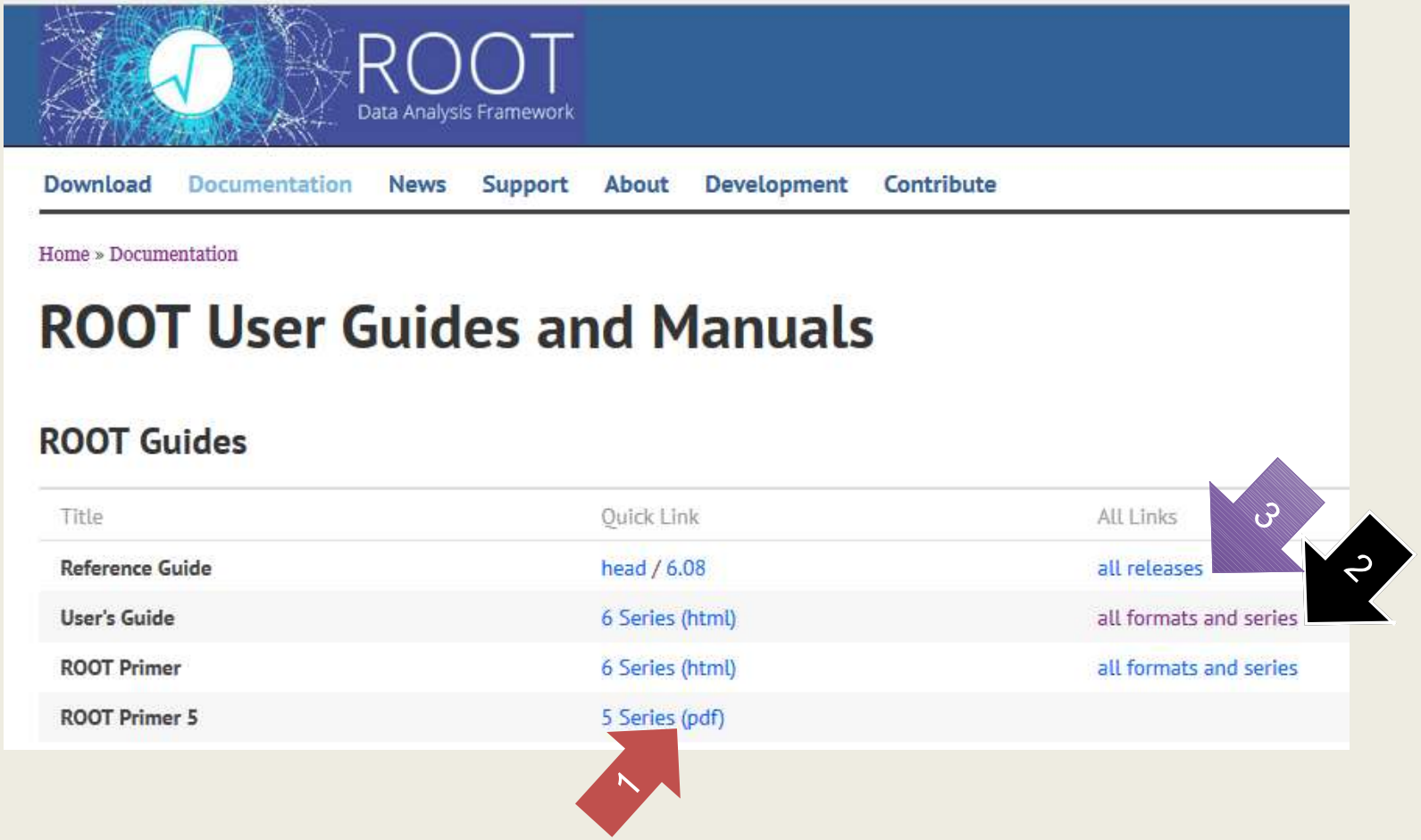
ROOT文档

- https://root.cern.ch/

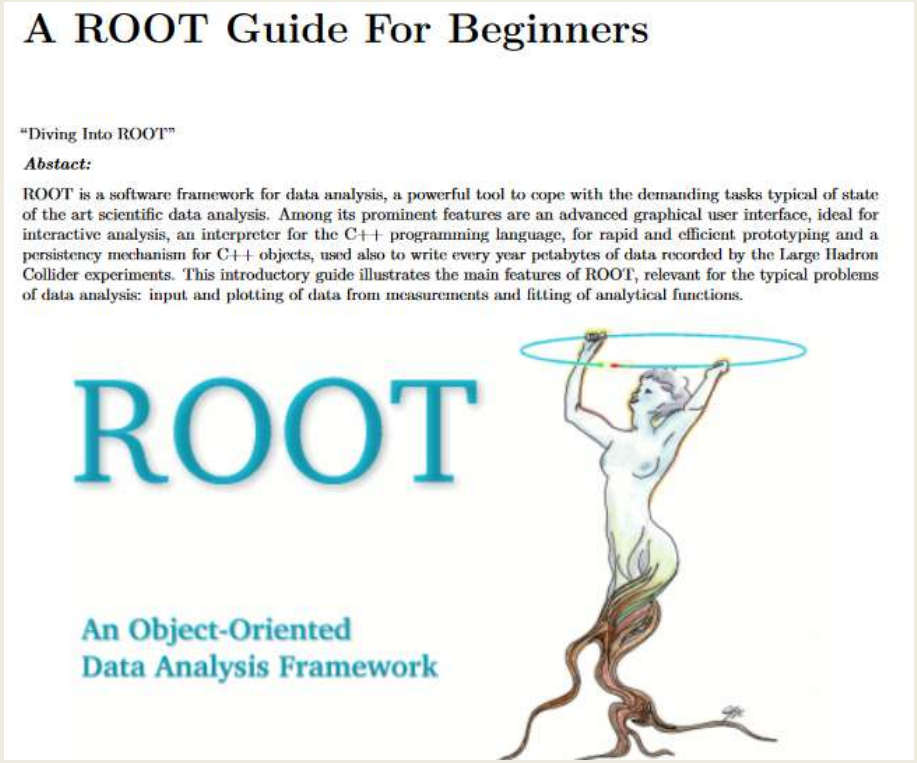


https://root.cern.ch/documentation

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



ROOT5Primer.pdf



A ROOT Guide For Beginners

"Diving Into ROOT"

Abstact:

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.

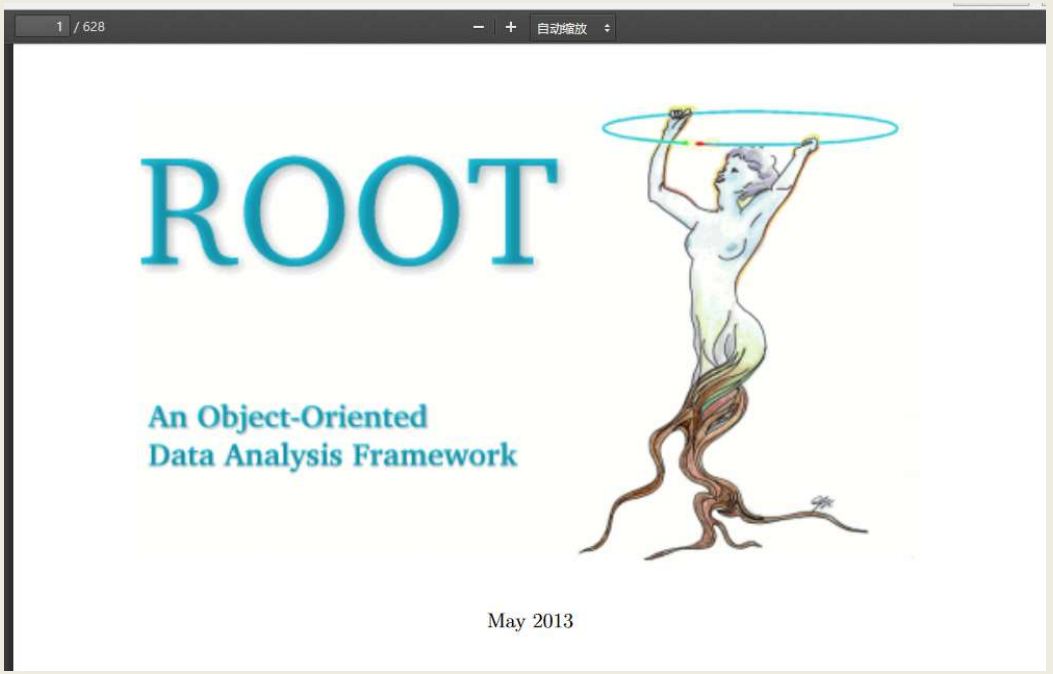
https://d35c7d8c.web.cern.ch/sites/d35c7d8c.web.cern.ch/files/ROOT5Primer.pdf 初学者以此作为参考书

User's Guide

ROOT Old User's Guide (5.34)

ROOT User's Guide in PDF format

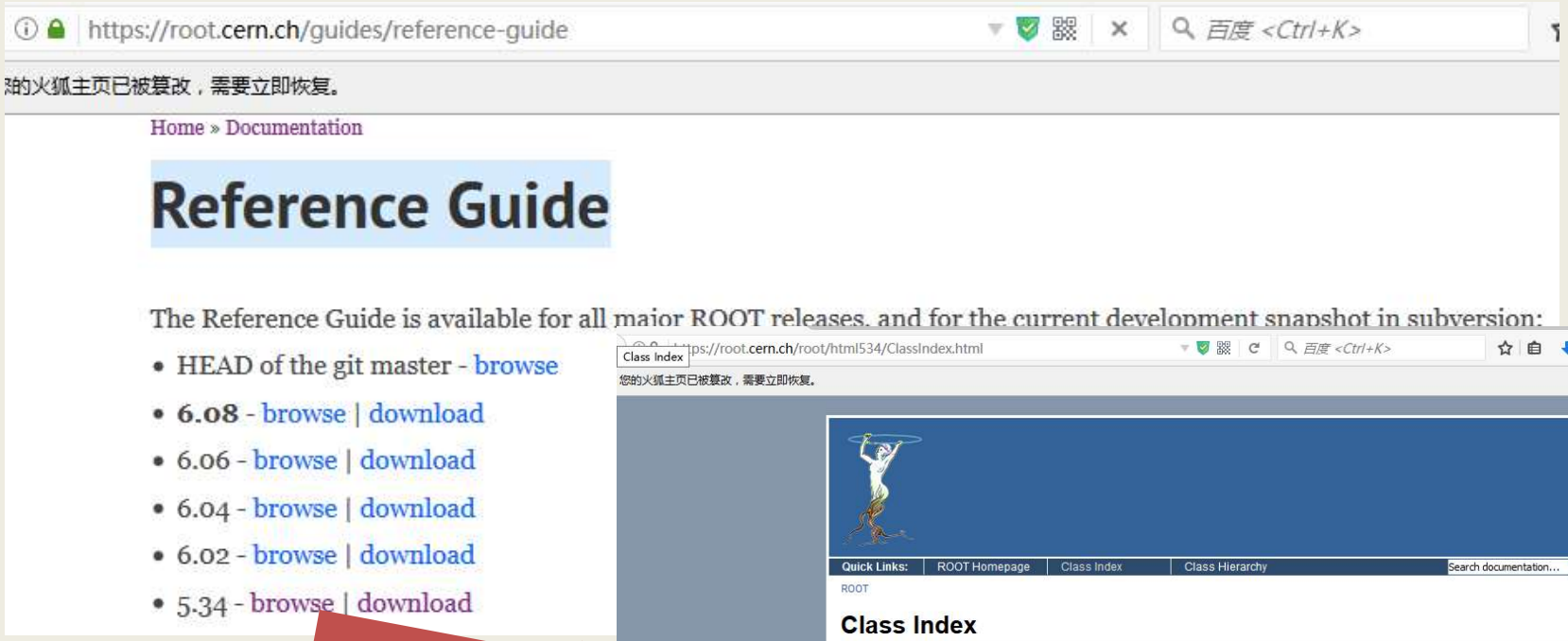
- Complete User's Guide in A4 format.
- Complete User's Guide in Letter format.
- Chapters:
  a. Introduction
  b. Getting Started
  c. Histograms
  d. Graphs
  e. Fitting Histograms
  f. A Little C++
  g. CINT
  h. Object Ownership
  i. Graphics
  j. Folders and Tasks
  k. Input/Output
  l. Trees
 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



主要的参考

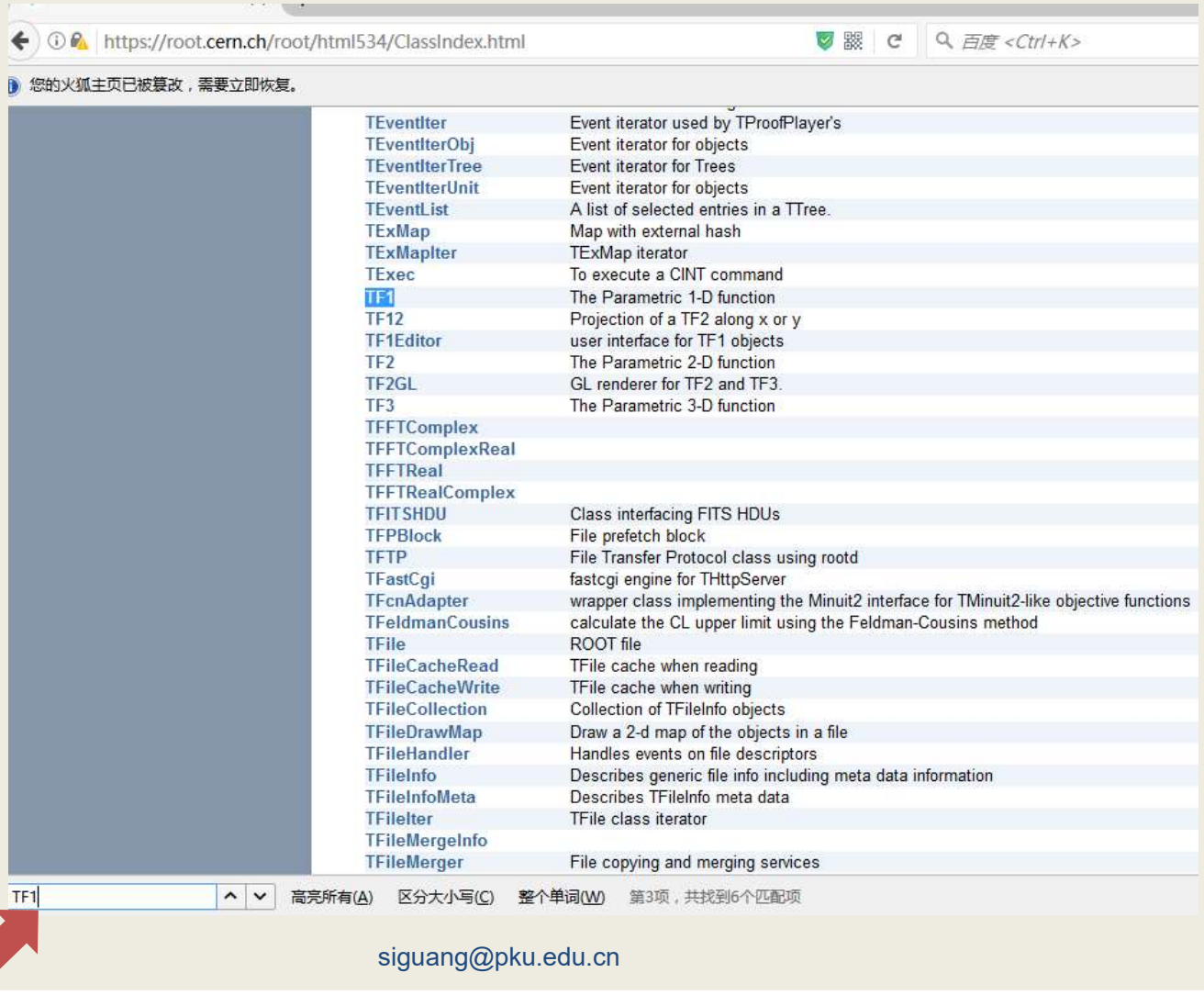
https://root.cern.ch/guides/users-guide

Reference Guide



siguang@pku.edu.cn

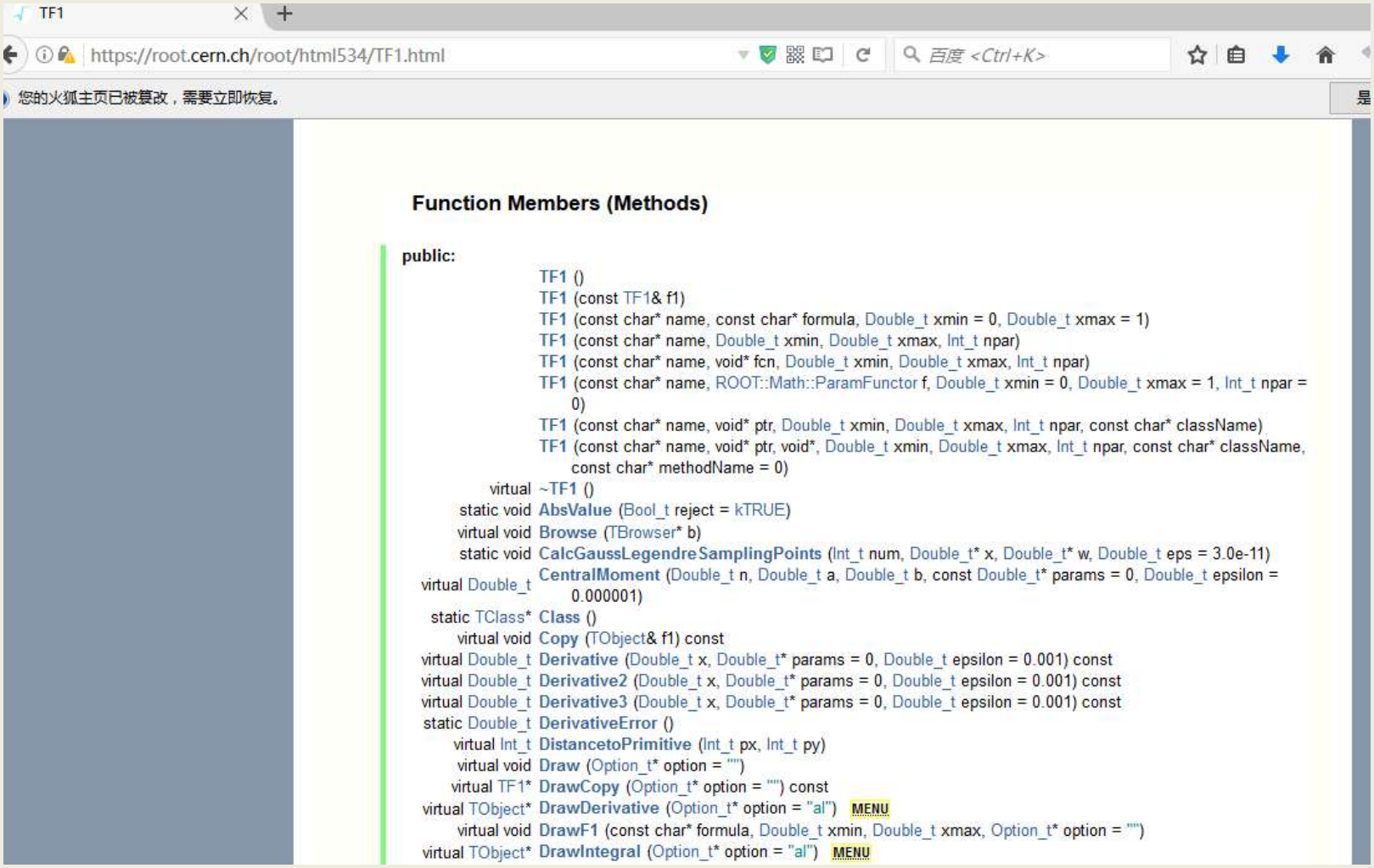
TF1函数定义网页



CTRL+F 搜索：

siguang@pku.edu.cn

TF1函数定义网页



Function Members (Methods)

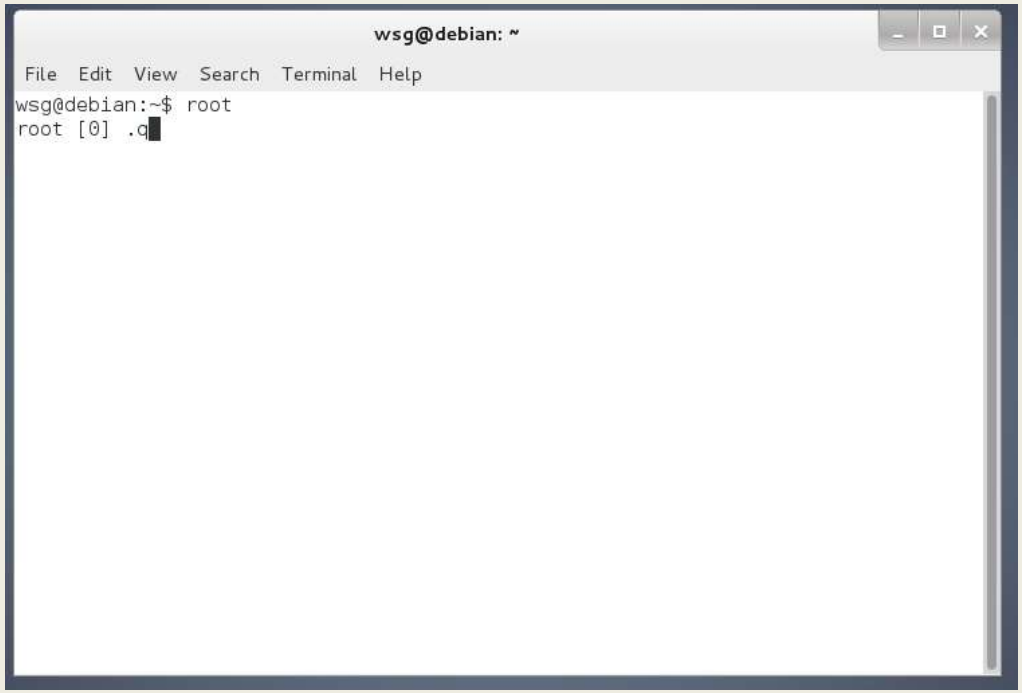
public:

```
                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::ParamFunctor 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

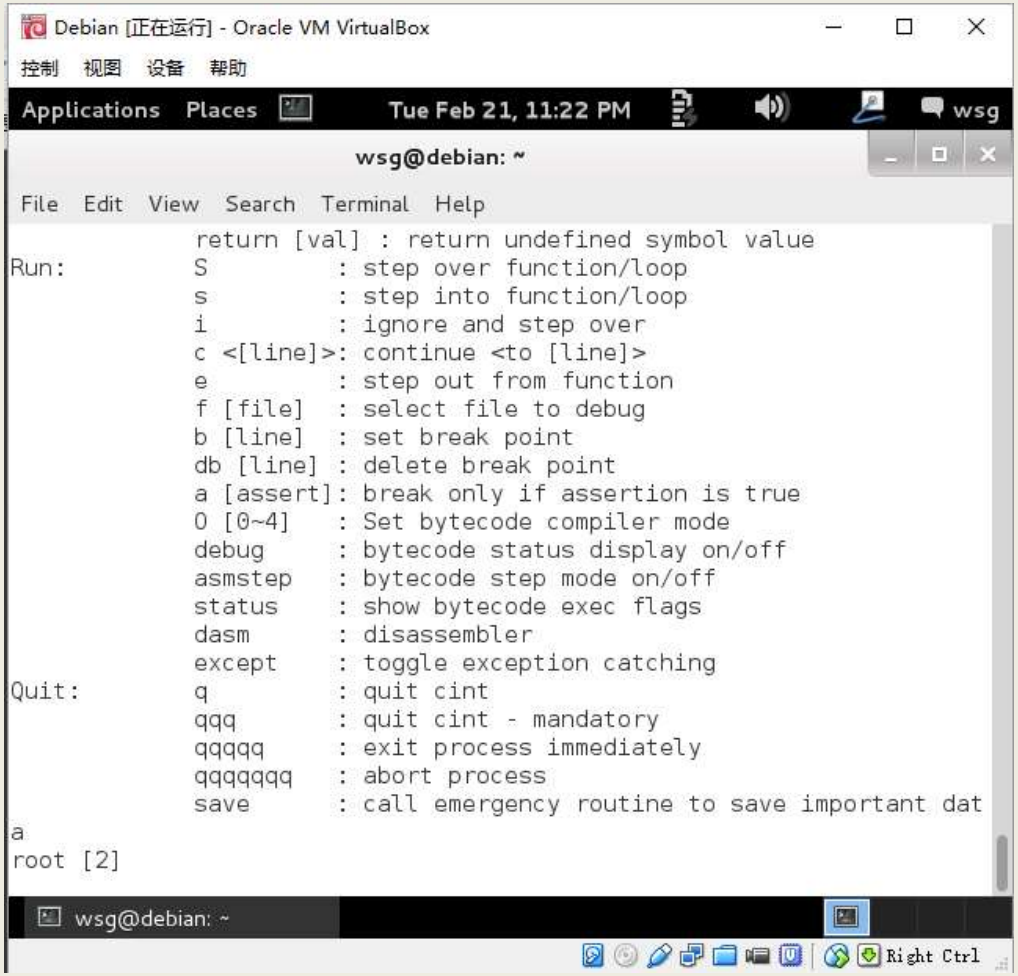
启动退出ROOT

- 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



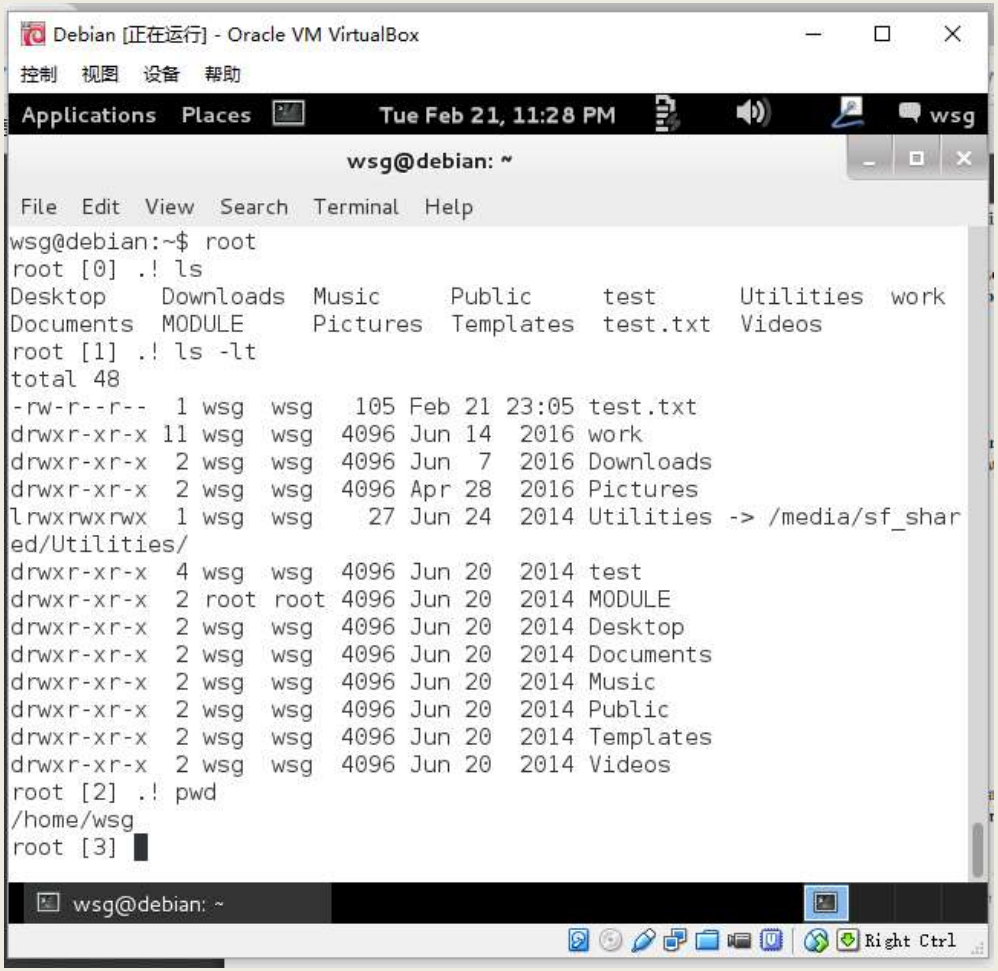
ROOT常用控制命令(.?将所有的命令列出来）

- Root[].?



ROOT常用控制命令(.! shell 命令 ）

- Root[].! ls



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)1.0000000000000000e+00
root [2] sqrt(3)
(const double)1.7320508075688719e+00
root [3] 1 > 2
(const int)0
root [4] TMath::Pi()
(Double_t)3.1415926535897932e+00
root [5] TMath::Erf(.2)
(Double_t)2.2270258921047844e-01
```

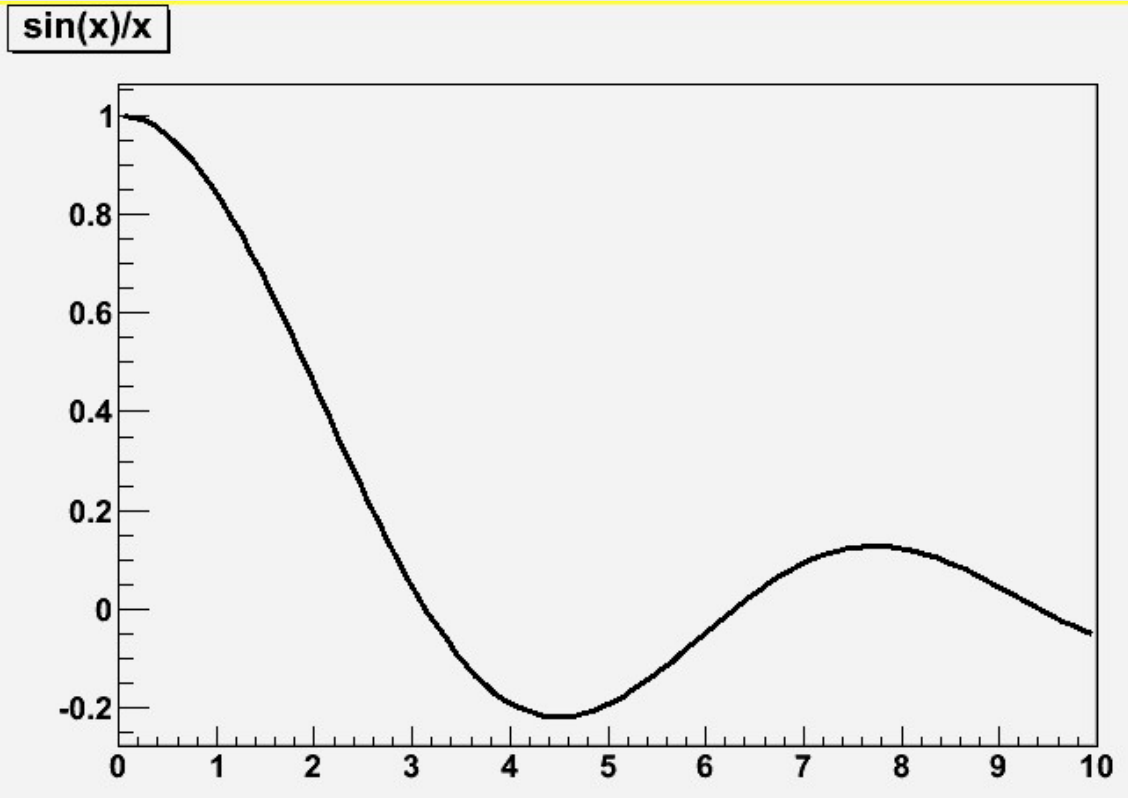
ROOT作为计算器 [续]

```
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();
```

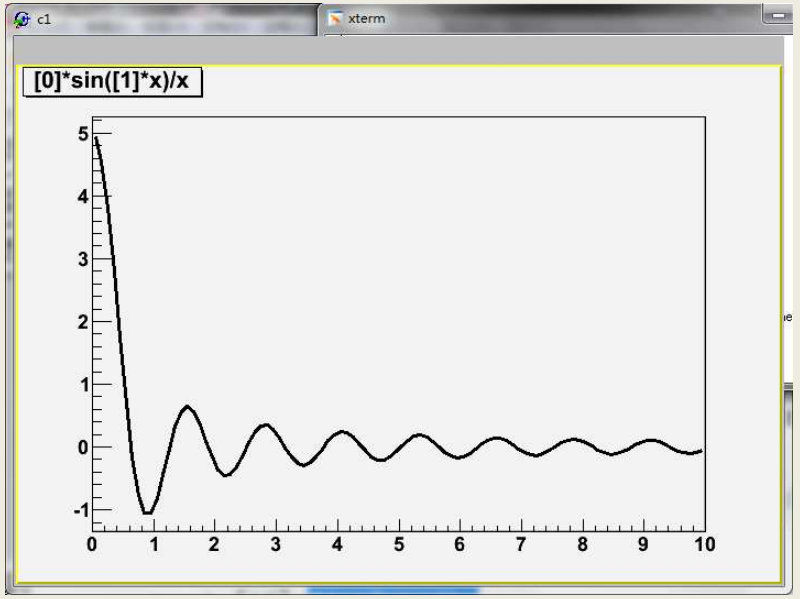


siguang@pku.edu.cn

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()



siguang@pku.edu.cn

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

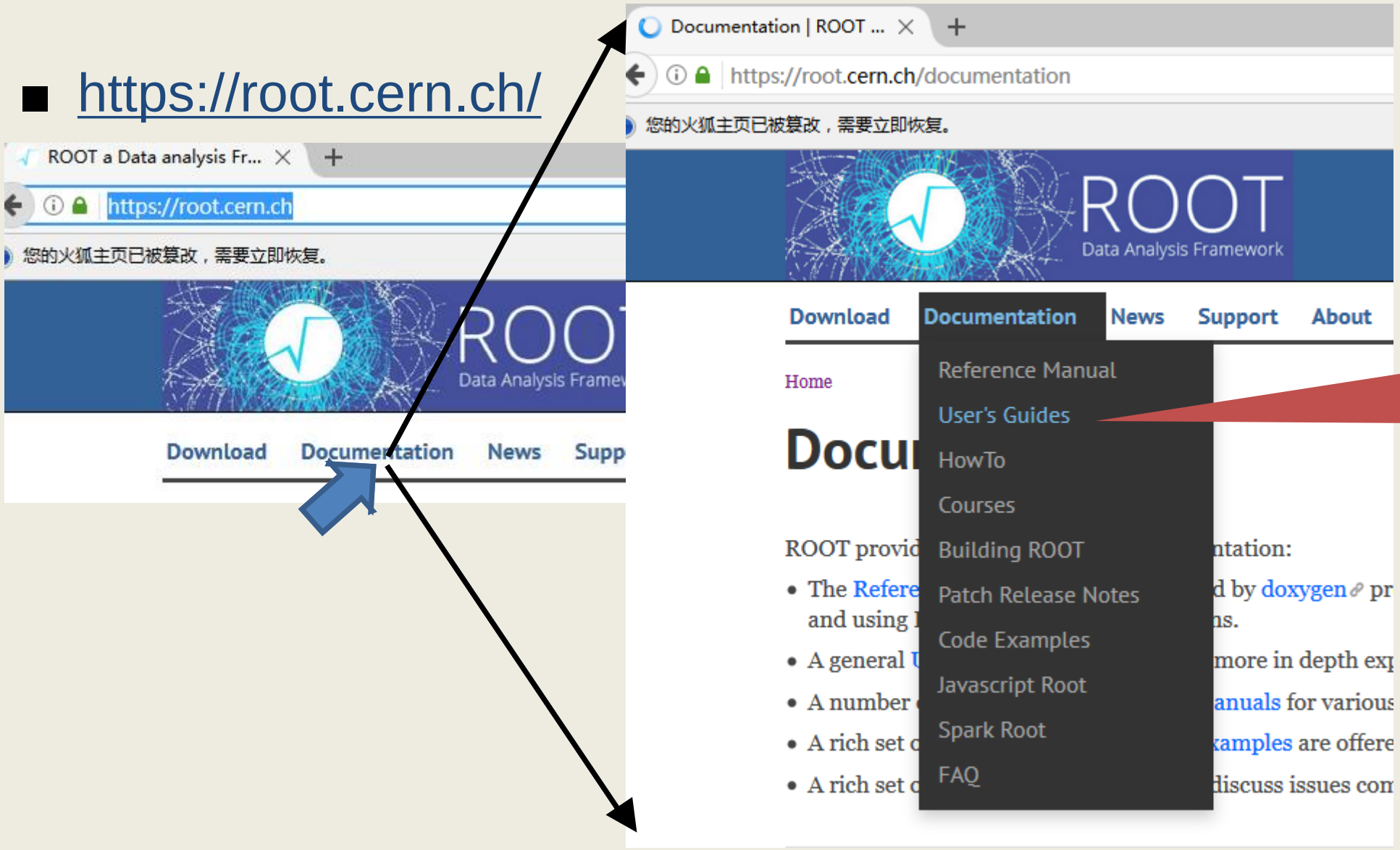
目录

- 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

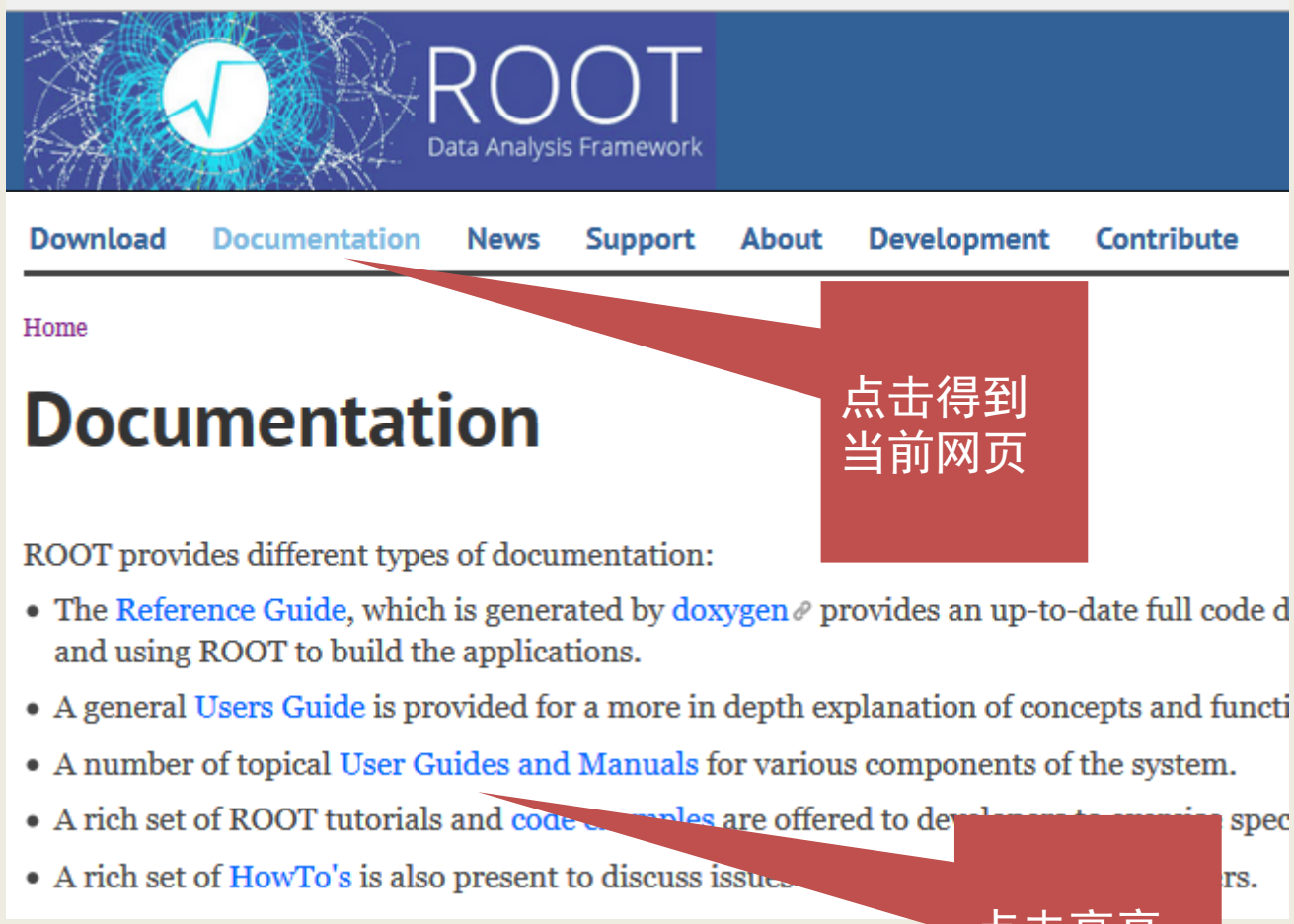
ROOT文档

ROOT文档

- https://root.cern.ch/

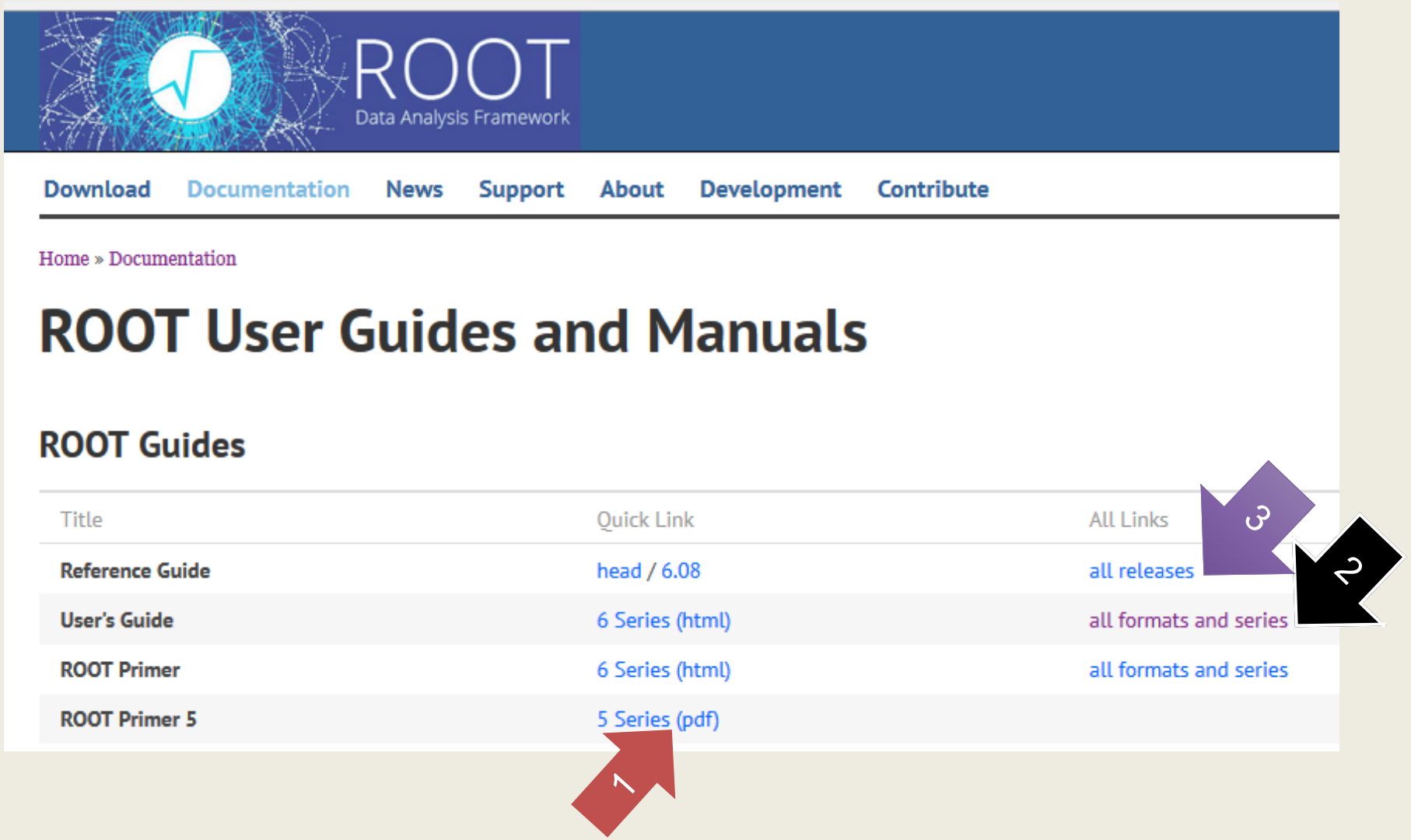


https://root.cern.ch/documentation



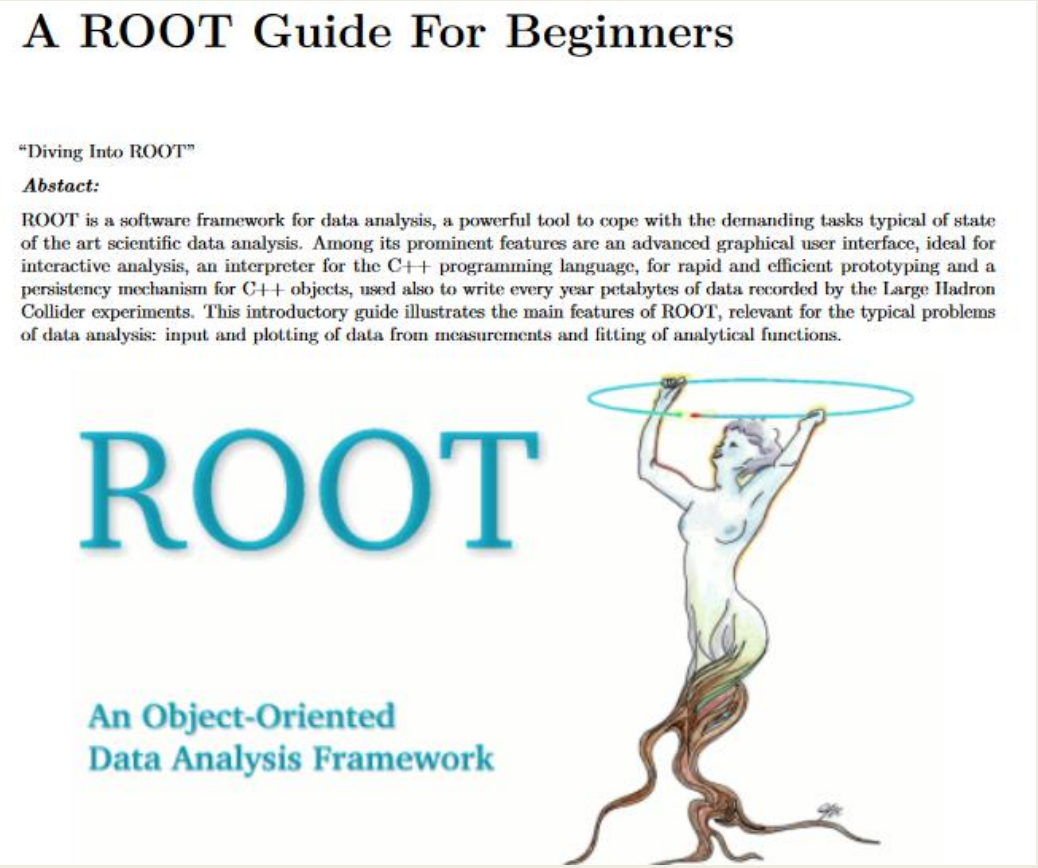
siguang@pku.edu.cn

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



siguang@pku.edu.cn

ROOT5Primer.pdf



A ROOT Guide For Beginners

"Diving Into ROOT"

Abstact:

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.

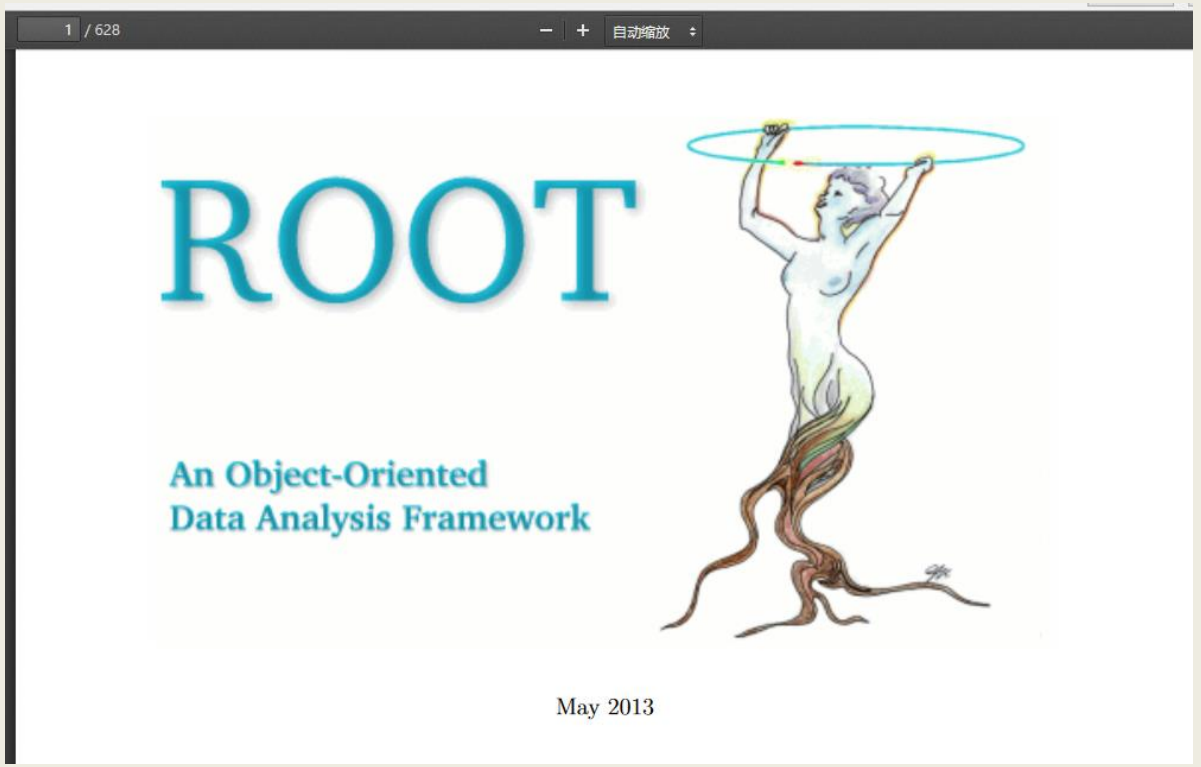
https://d35c7d8c.web.cern.ch/sites/d35c7d8c.web.cern.ch/files/ROOT5Primer.pdf 初学者以此作为参考书

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

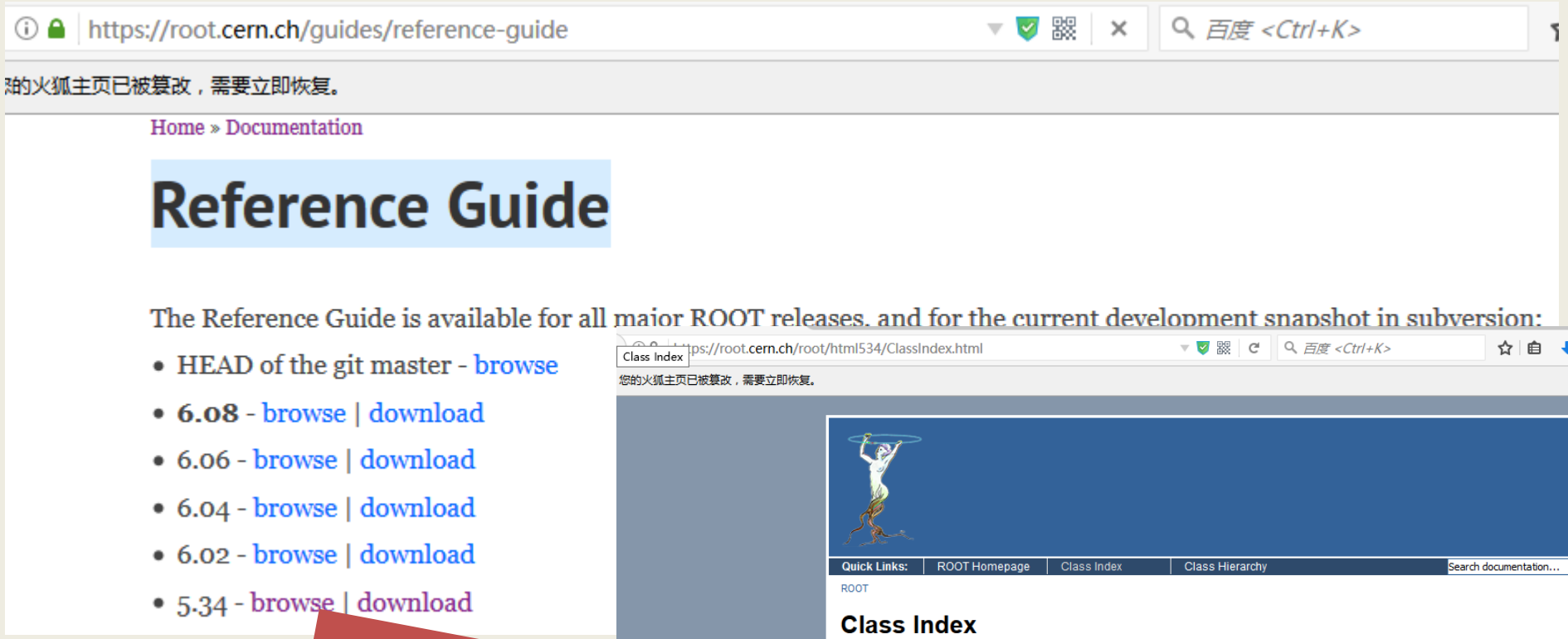


主要的参考

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

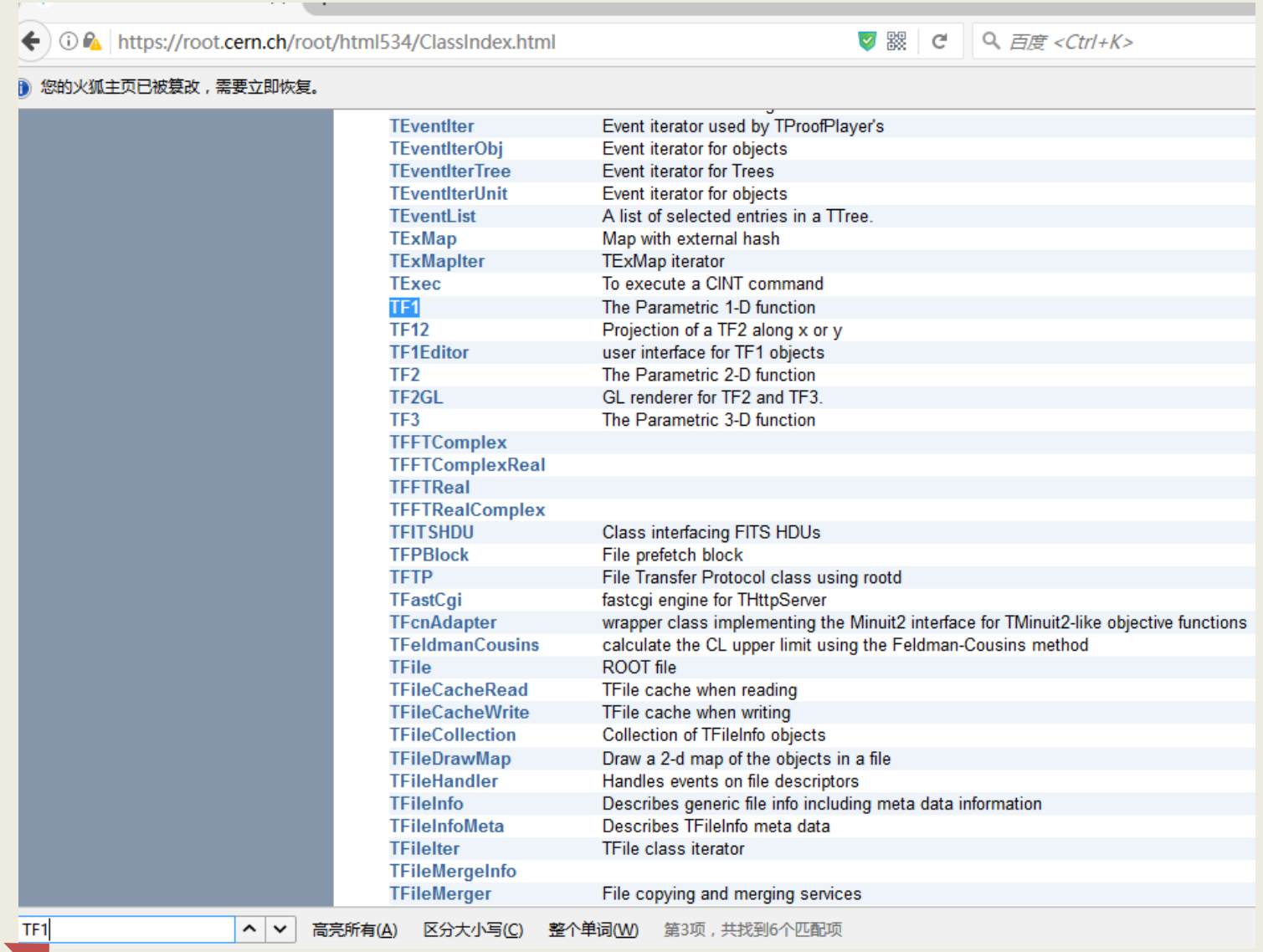
siguang@pku.edu.cn

Reference Guide



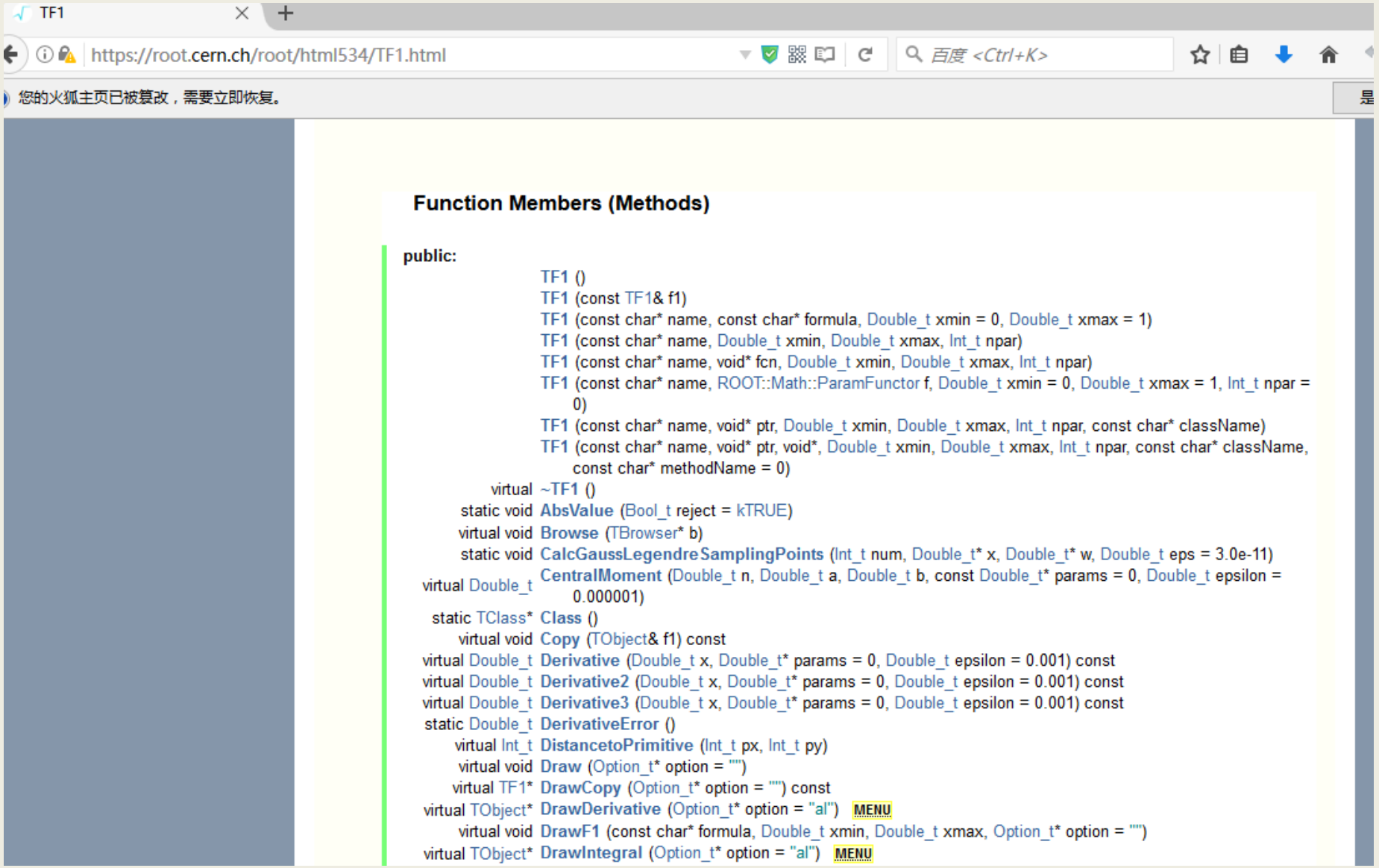
siguang@pku.edu.cn

TF1函数定义网页



CTRL+F 搜索：

TF1函数定义网页



Function Members (Methods)

public:

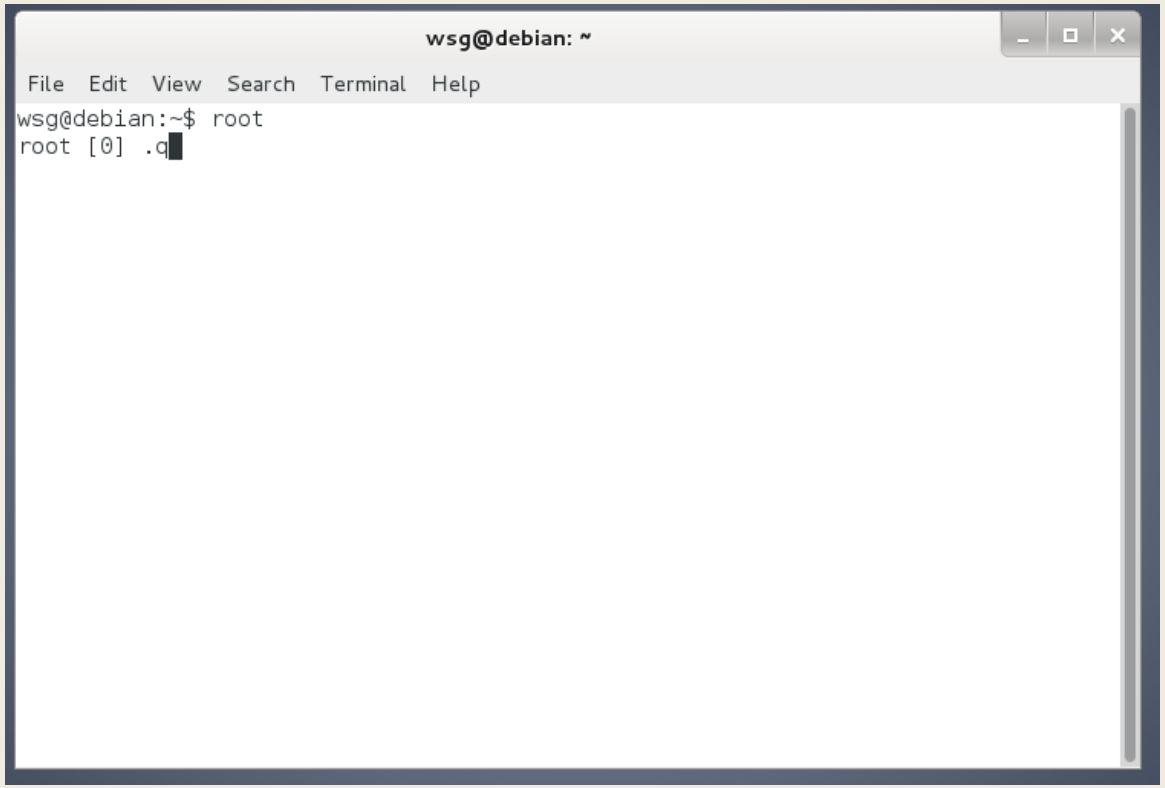
```
                TF1 ()
                TF1 (const TF1& f1)
                TF1 (const char* name, const char* formula, Double_t xmin = 0, Double_t xmax = 1)
                TF1 (const char* name, Double_t xmin, Double_t xmax, Int_t npar)
                TF1 (const char* name, void* fcn, Double_t xmin, Double_t xmax, Int_t npar)
                TF1 (const char* name, ROOT::Math::ParamFunctor 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
```

siguang@pku.edu.cn

ROOT常用控制命令

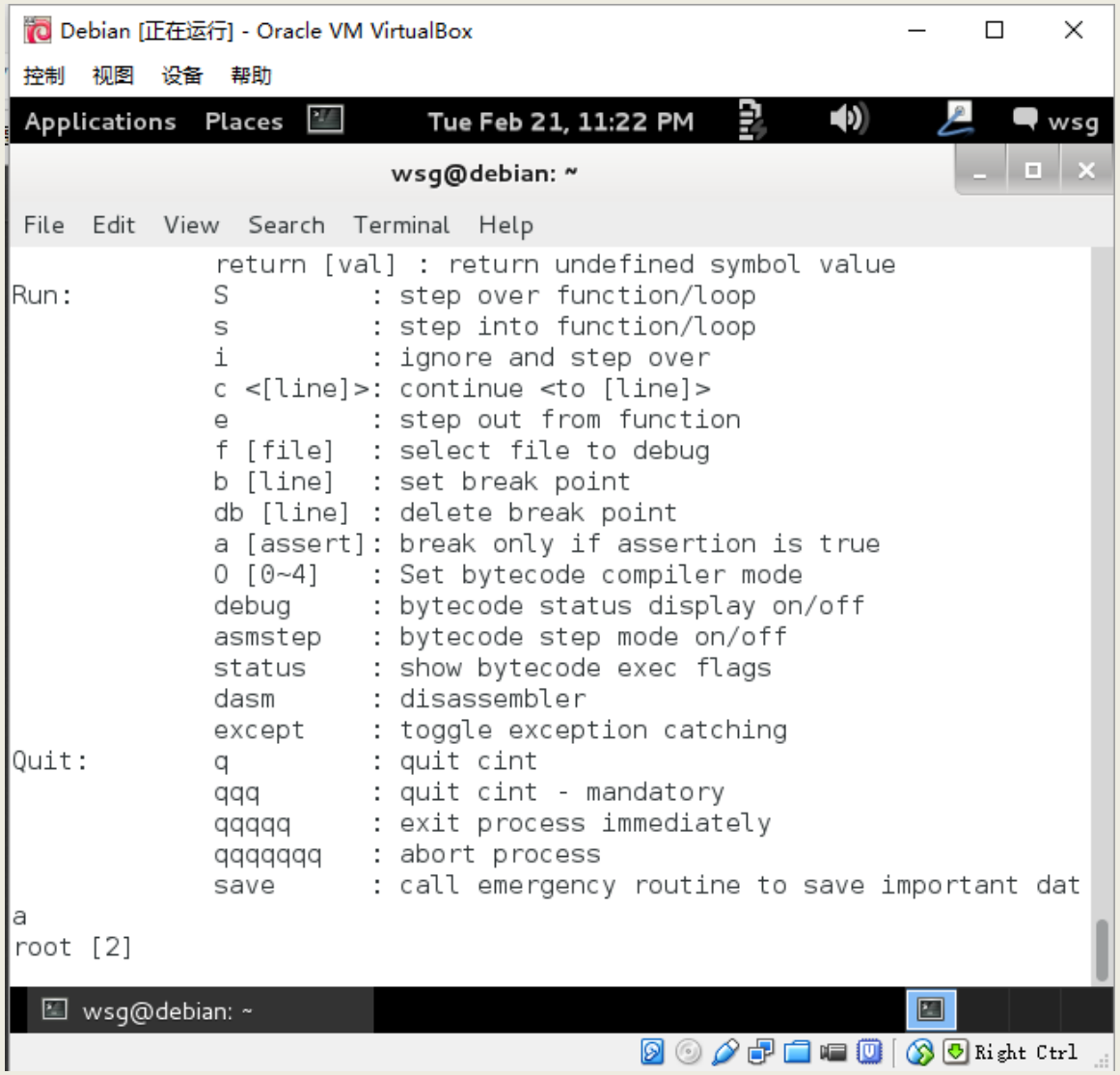
启动退出ROOT

- 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



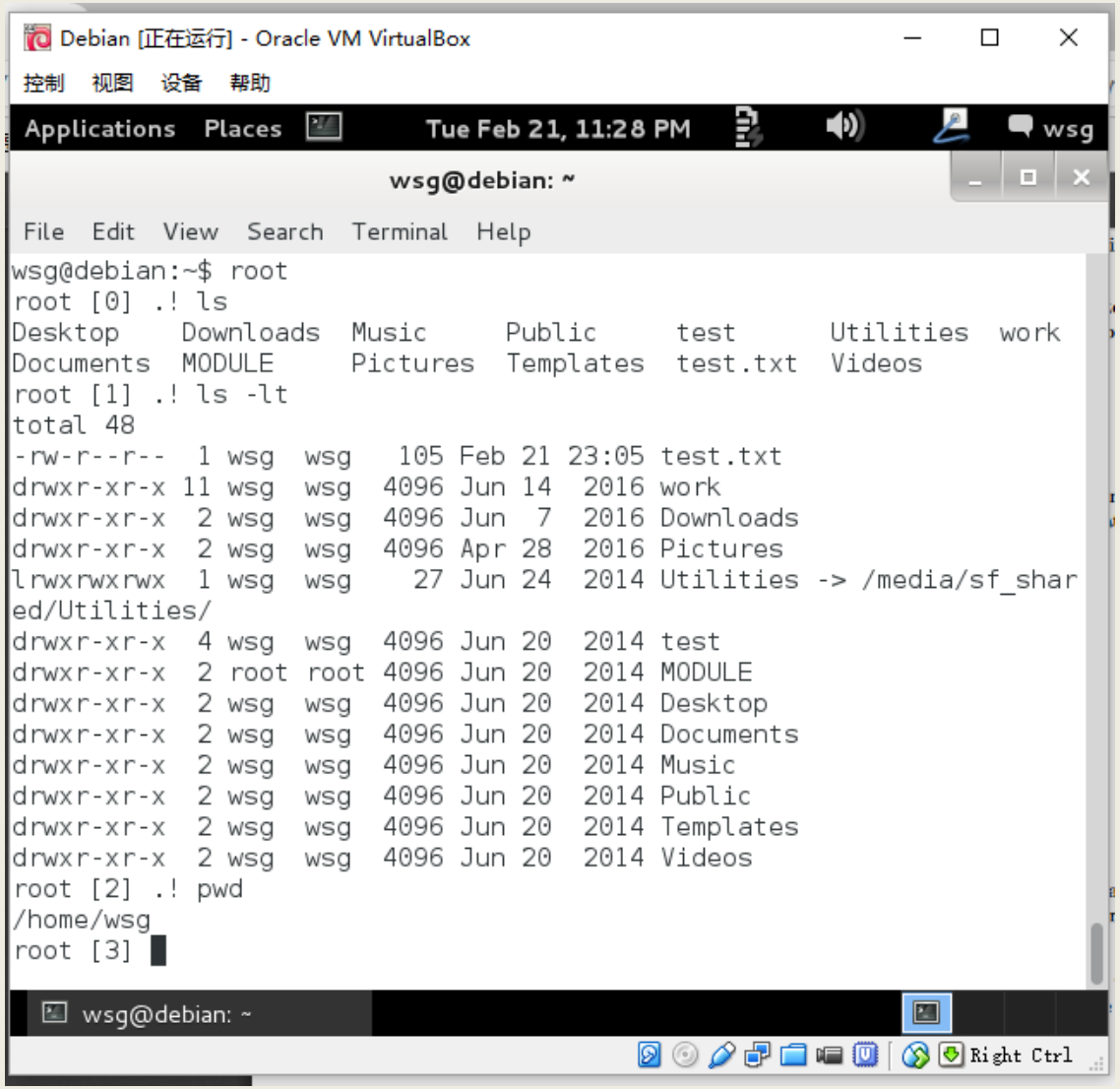
ROOT常用控制命令(.?将所有的命令列出来)

- Root[].?



ROOT常用控制命令(.! shell 命令 ）

- Root[].! ls



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)1.00000000000000000e+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
```

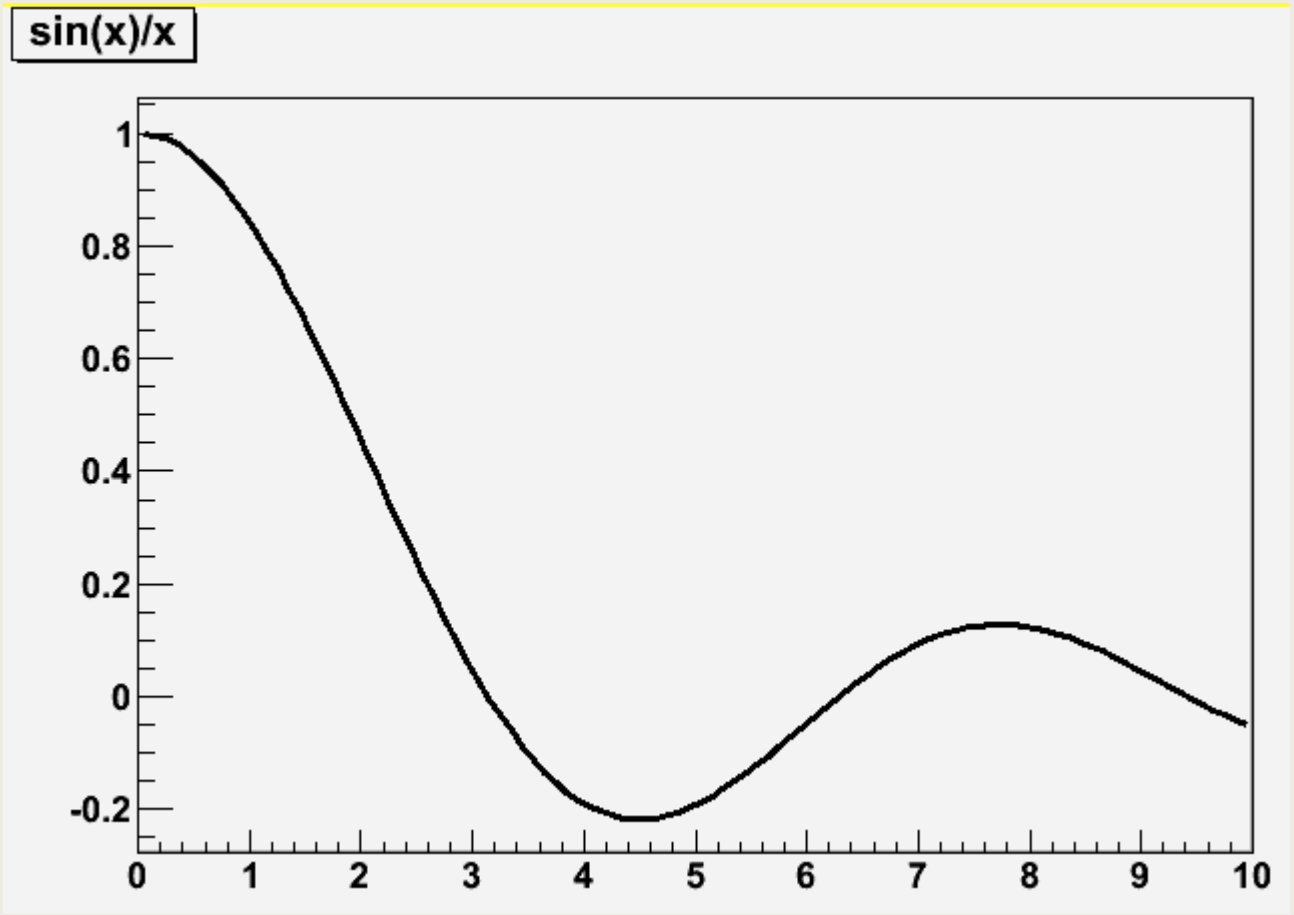
ROOT作为计算器 [续]

```
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)1.8626451492309570 3e-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();
```

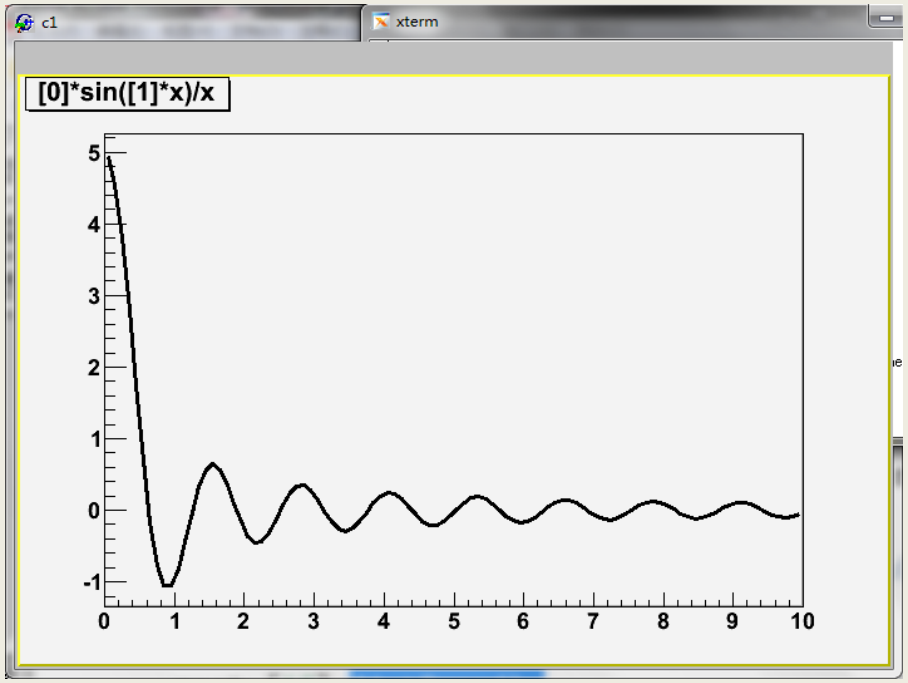


siguang@pku.edu.cn

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()



siguang@pku.edu.cn

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);
```

siguang@pku.edu.cn

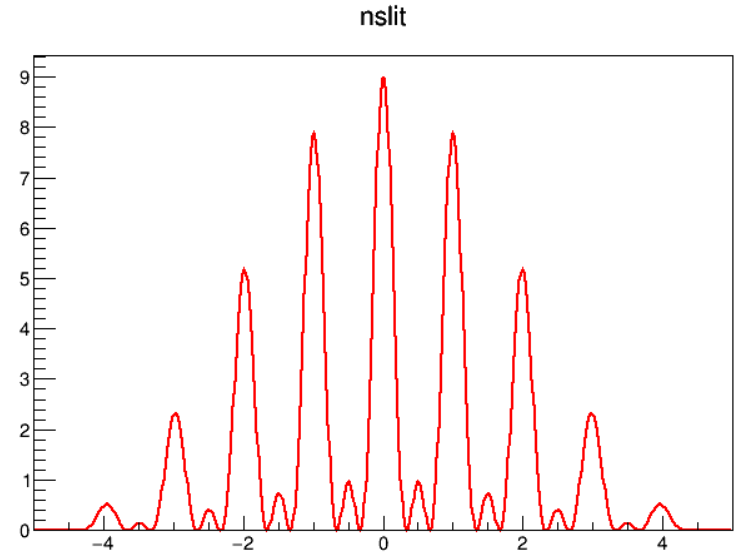
slits.C [续]

```
  // 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。。。。。