



ROOT

Data Analysis Framework

Google Custom Search

Download

Documentation

News

Support

About

Development

Contribute

Download ROOT

All Releases

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

Releases

Pro Release 6.10/02 - 2017-07-06

Pro Release 6.10/06 - 2017-09-19

Old Release 6.08/06 - 2017-03-02

Version 6

Release 6.10/04 - 2017-07-28

Release 6.10/00 - 2017-06-13

Release 6.09/02 - 2017-03-08

Release 6.08/04 - 2017-01-13

Release 6.08/02 - 2016-12-02

Release 6.08/00 - 2016-11-04

Release 6.06/08 - 2016-09-01

Release 6.06/06 - 2016-07-06

Release 6.06/04 - 2016-05-03

Release 6.04/06 - 2015-10-13

Release 6.04/04 - 2015-10-08

Release 6.04/02 - 2015-07-14

Release 6.04/00 - 2015-06-02

Release 6.03/04 - 2015-04-22

Release 6.03/02 - 2015-01-27

Release 6.02/12 - 2015-06-24

Release 6.02/10 - 2015-05-29

Release 6.02/08 - 2015-04-13

Release 6.02/05 - 2015-02-00

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

Version 5

Release 5.99/06 - 2014-04-03

Release 5.99/05 - 2014-02-07

Release 5.99/04 - 2013-11-27

Release 5.34/36 - 2016-04-05

Release 5.34/34 - 2015-10-02

Release 5.34/32 - 2015-06-23

Release 5.34/30 - 2015-04-23

Release 5.34/28 - 2015-03-24

Release 5.34/26 - 2015-02-20

Release 5.34/25 - 2015-01-12

Release 5.34/24 - 2014-12-02

Release 5.34/23 - 2014-11-07

Release 5.34/22 - 2014-10-10

Release 5.34/21 - 2014-09-09

Release 5.34/20 - 2014-08-13

Release 5.34/19 - 2014-07-09

Release 5.34/18 - 2014-03-14

Release 5.32/00 - 2011-12-02

Release 5.30/06 - 2012-01-10

Release 5.30/05 - 2011-11-24

Release 5.30/04 - 2011-11-08

Release 5.30/03 - 2011-10-24

Release 5.30/02 - 2011-09-22

Release 5.30/01 - 2011-08-18

Release 5.30/00 - 2011-06-28

Release 5.29/02 - 2011-04-21

Release 5.28/00h - 2011-11-24

Release 5.28/00g - 2011-09-14

Release 5.28/00f - 2011-08-08

Release 5.28/00e - 2011-06-21

Release 5.28/00d - 2011-05-07

Release 5.28/00c - 2011-04-15

Release 5.28/00b - 2011-03-14

Release 5.28/00a - 2011-02-23



<https://root.cern.ch/content/release-53436>

Binary distributions

Windows Visual Studio 2010 (dbg)	root_v5.34.36.win32.vc10.debug.exe	92M
Windows Visual Studio 2010 (dbg)	root_v5.34.36.win32.vc10.debug.zip	140M
Windows Visual Studio 2010	root_v5.34.36.win32.vc10.exe	46M
Windows Visual Studio 2010	root_v5.34.36.win32.vc10.zip	63M
Windows Visual Studio 2012 (dbg)	root_v5.34.36.win32.vc11.debug.exe	98M
Windows Visual Studio 2012 (dbg)	root_v5.34.36.win32.vc11.debug.zip	151M
Windows Visual Studio 2012	root_v5.34.36.win32.vc11.exe	47M
Windows Visual Studio 2012	root_v5.34.36.win32.vc11.zip	64M
Windows Visual Studio 2013 (dbg)	root_v5.34.36.win32.vc12.debug.exe	99M
Windows Visual Studio 2013 (dbg)	root_v5.34.36.win32.vc12.debug.zip	152M
→ Windows Visual Studio 2013	root_v5.34.36.win32.vc12.exe	47M
Windows Visual Studio 2013	root_v5.34.36.win32.vc12.zip	64M

ROOT

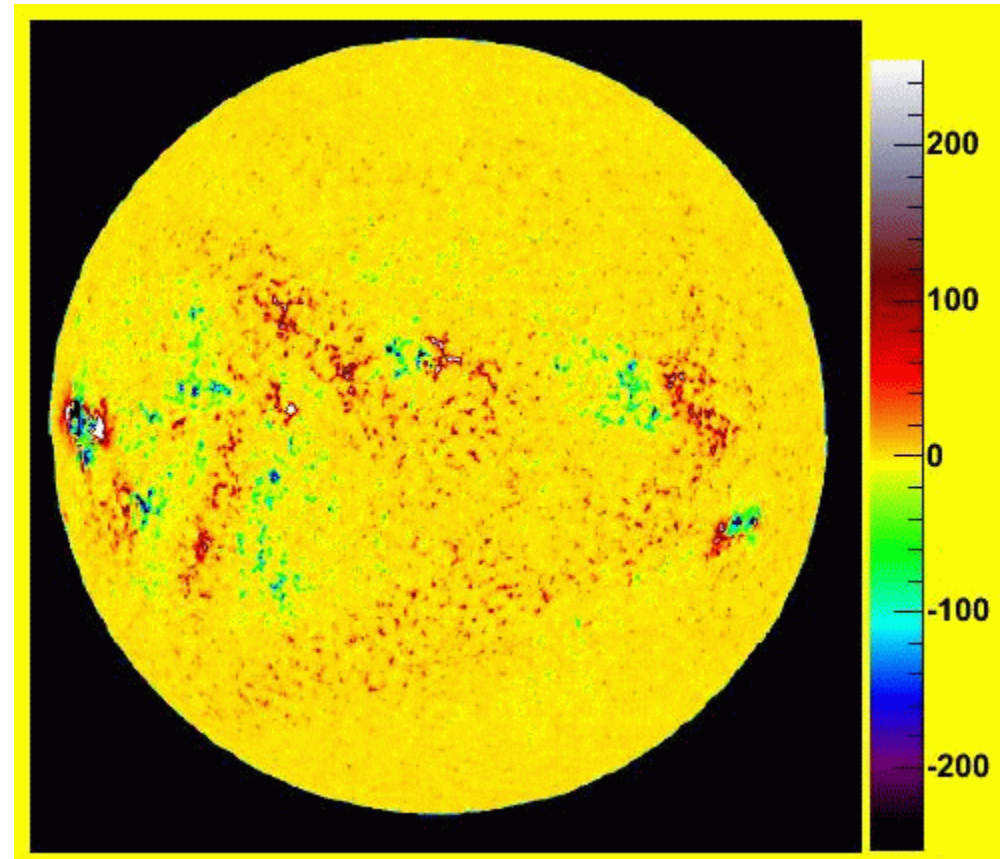
<http://root.cern.ch>

základní příkazy

`.L file.c` - načtení souboru s programem

`.U file.c` - uvolnění programem z paměti

`run()` - spuštění funkce `run`



ROOT

<http://root.cern.ch>

- **deklarace proměnných**

`int` - integer (celé číslo 4 B = 32 bitů)

`float` - reálné číslo (jednoduchá přesnost 4 B = 32 bitů)

`double` - reálné číslo (dvojitá přesnost 8 B = 64 bitů)

- **příklady**

```
int i;
```

```
double x,y;
```

```
int h[100];   pole 100 integerů
```

```
#define max 1000
```

```
double x[max],y[max];
```

```
dvě pole 1000 reálných čísel z dvojitou přesností
```



ROOT

<http://root.cern.ch>

- **přirazení & tisk**

```
double x;  
x=3.141;  
printf(“%lf“,x);
```

```
int i=0;  
i++;  
printf(“zvetseno na %d“,i);  
i--;  
printf(“zmenseno na %d“,i);
```



ROOT

<http://root.cern.ch>

• podmínka

```
if(Logický výraz)
{
    příkaz1;
    příkaz2;
    . . .
}
else
{
    příkaz1;
    příkaz2;
    . . .
}
```

• logické operátory

== - rovná se

!= - nerovná se

> - je větší

< - je menší

>= - je větší nebo rovno

<= - je menší nebo rovno

! - logický operátor NOT

&& - logický operátor AND

|| - logický operátor OR



ROOT

<http://root.cern.ch>

• podmínka

```
if(Logický výraz)
{
    příkaz1;
    příkaz2;
    . . .
}
else
{
    příkaz1;
    příkaz2;
    . . .
}
```

• příklady

```
if(a!=0) x=b/a;      test na dělení nulou
```

funkce na výpočet absolutní hodnoty

```
double absolutni_hodnota(double x)
{
    if(x<0)return(-x);
    else return(x);
}
```



ROOT

<http://root.cern.ch>

- **cyklus for**

```
for(počáteční výraz; podmínka; krok)
{
    příkaz1;
    příkaz2;
    . . .
}
```

- **příklad** výpočet součtu pole a součtu kvadrátů pole

```
suma=suma2=0.0;
for(i=0; i<max; i++)
{
    suma=suma+h[i];
    suma2=suma2+pow(h[i],2);
}
```



ROOT

<http://root.cern.ch>

- **cyklus while**

```
while(podmínka)
{
    příkaz1;
    příkaz2;
    . . .
}
```

- **příklad** výpočet součtu pole a součtu kvadrátů pole

```
suma=suma2=0.0;
i=0;
while(i<max)
{
    suma=suma+h[i];
    suma2=suma2+pow(h[i],2);
    i++;
}
```



ROOT

<http://root.cern.ch>

- **cyklus do ... while**

```
do {  
    příkaz1;  
    příkaz2;  
    . . .  
} while (podmínka);
```

- **příklad** výpočet součtu pole a součtu kvadrátů pole

```
suma=suma2=0.0;  
i=0;  
do {  
    suma=suma+h[i];  
    suma2=suma2+pow(h[i],2);  
    i++;  
} while (i<max);
```



ROOT

<http://root.cern.ch>

• tvorba grafů

```
TCanvas *c = new TCanvas("c1","navez",10,10,600,600);
```

vytvoření okna pro vykreslení grafu,

poloha levého horního rohu okna: 10, 10 px, velikost okna: 600, 600 px

1D graf

```
TGraph *g = new TGraph(n,x,y);
```

vytvoření 1D grafu, z polí x , y o velikosti n

```
g->Draw("AP");
```

 vykreslení grafu

2D graf

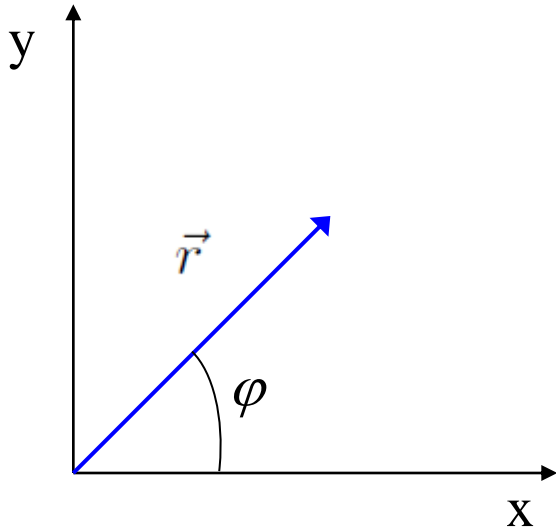
```
TGraph2D *g = new TGraph2D(n,x,y,z);
```

vytvoření 2D grafu, z polí x , y , z o velikosti n

```
g->Draw("P");
```

 vykreslení grafu

Kruhový pohyb



polární souřadnice

$$r(t) = r$$

$$\varphi(t) = \omega t$$

ω - úhlová rychlost

$$T = \frac{2\pi}{\omega} \quad \text{- perioda}$$

kartézské souřadnice

$$x(t) = r \cos \varphi = r \cos(\omega t)$$

$$y(t) = r \sin \varphi = r \sin(\omega t)$$

Kruhový pohyb

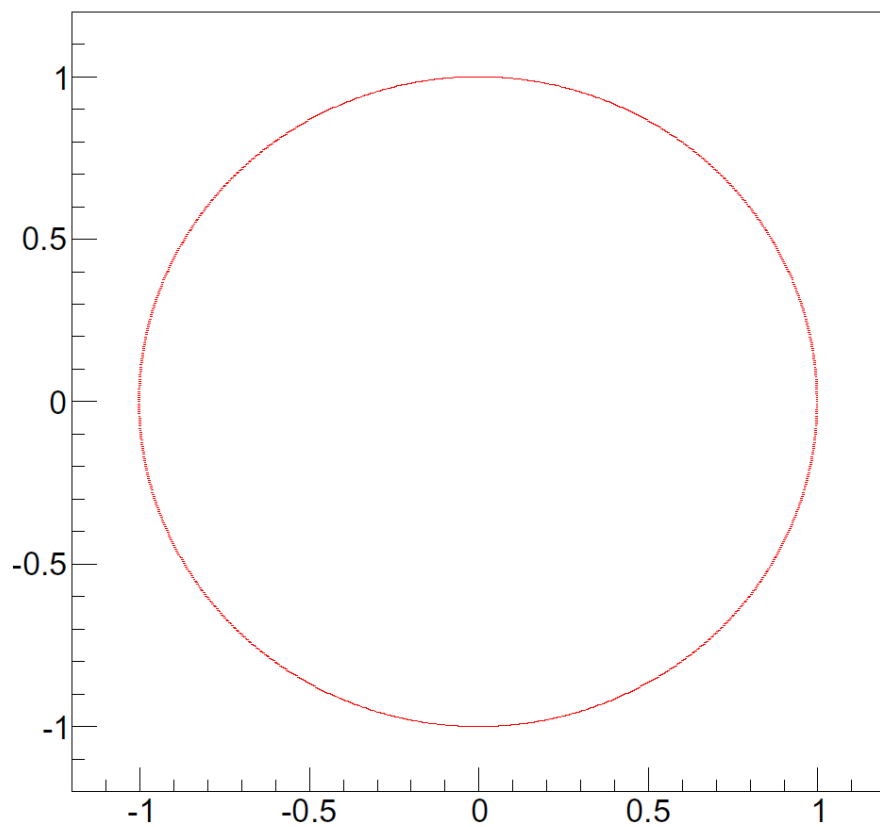
```
//trajektorie kruhoveho pohybu
#define n 1000 //pocet bodu na 1 otocku
#define pi 3.1415926535897932384626433832795
int run()
{
    int i;
    double t[n]; //cas
    double dt; // casovy krok
    double r=1; //radius
    double omega=2*pi; //uhlova rychlost 1 otocka/s
    double x[n],y[n];
    dt=1.0/n; //n bodu na 1 otocku
    for(i=0;i<n;i++)
    {
        t[i]=i*dt;
        x[i]=r*cos(omega*t[i]);
        y[i]=r*sin(omega*t[i]);
    }
    TGraph *gr1 = new TGraph(n,x,y); //vytvoreni grafu
    TCanvas *c1 = new TCanvas("c1","trajektorie kruhoveho pohybu",10,10,600,600);
//vytvoreni okna
    gr1->SetMarkerColor(2); //nastaveni barvy
    gr1->SetTitle("trajektorie kruhoveho pohybu");
    gr1->Draw("AP"); //vykresleni grafu

    TGraph *gr2 = new TGraph(n,t,x); //vytvoreni grafu
    TGraph *gr3 = new TGraph(n,t,y); //vytvoreni grafu
    TCanvas *c2 = new TCanvas("c2","casova zavislost souradnic",100,100,600,600);
//vytvoreni okna
    gr2->SetMarkerColor(3); //nastaveni barvy symbolu
    gr3->SetMarkerColor(4); //nastaveni barvy symbolu
    gr2->SetTitle("casova zavislost souradnic");
    gr2->Draw("AP"); //vykresleni grafu
    gr3->Draw("P"); //vykresleni grafu

    return(0);
}
```

Kruhový pohyb

trajektorie



časová závislost souřadnic

