

---

# **ROOT prompt usability features and the strict option**

**Yuka Takahashi - Princeton University, CERN**  
**Vasil Geogiev Vasilev - Princeton University**  
**Axel Naumann - CERN**



## ROOT Prompt

```
$ root <options>  
root [0]
```

--strict  
(Available in 6.16!)



To begin with,  
What are existing options right now?



# Agenda

1. Summary of older options - Available NOW
2. Summary of fairly new options - Available NOW
3. Strict option - Will be available in the next release

# Summary of older options

## Summary of older options

Available since 5.28  
Requires: Nothing :)

- **memstat** `$ root -memstat hsimple.root`

- Benchmark your codes' memory usage
- Records all calls to malloc and free

```
$ root -memstat tutorials/hsimple.C -q -l
Info in <TMemStatMng::Close>: Tree saved to file memstat_27174.root
Info in <TMemStatMng::Close>: Tree entries = 33558, file size =
0.957537
Info in <Memstat::TMemStatMng::~~TMemStatMng>: >>>
All free/malloc calls count: 321995
Info in <Memstat::TMemStatMng::~~TMemStatMng>: >>>
Unique BTIDs count: 63363
```

## Summary of older options

### ACLIC

```
$ root hsimple.C+
```

Available since long time ago  
Requires: External compiler

- Ensure the code is correct enough
- Code run in the speed of compiled C++

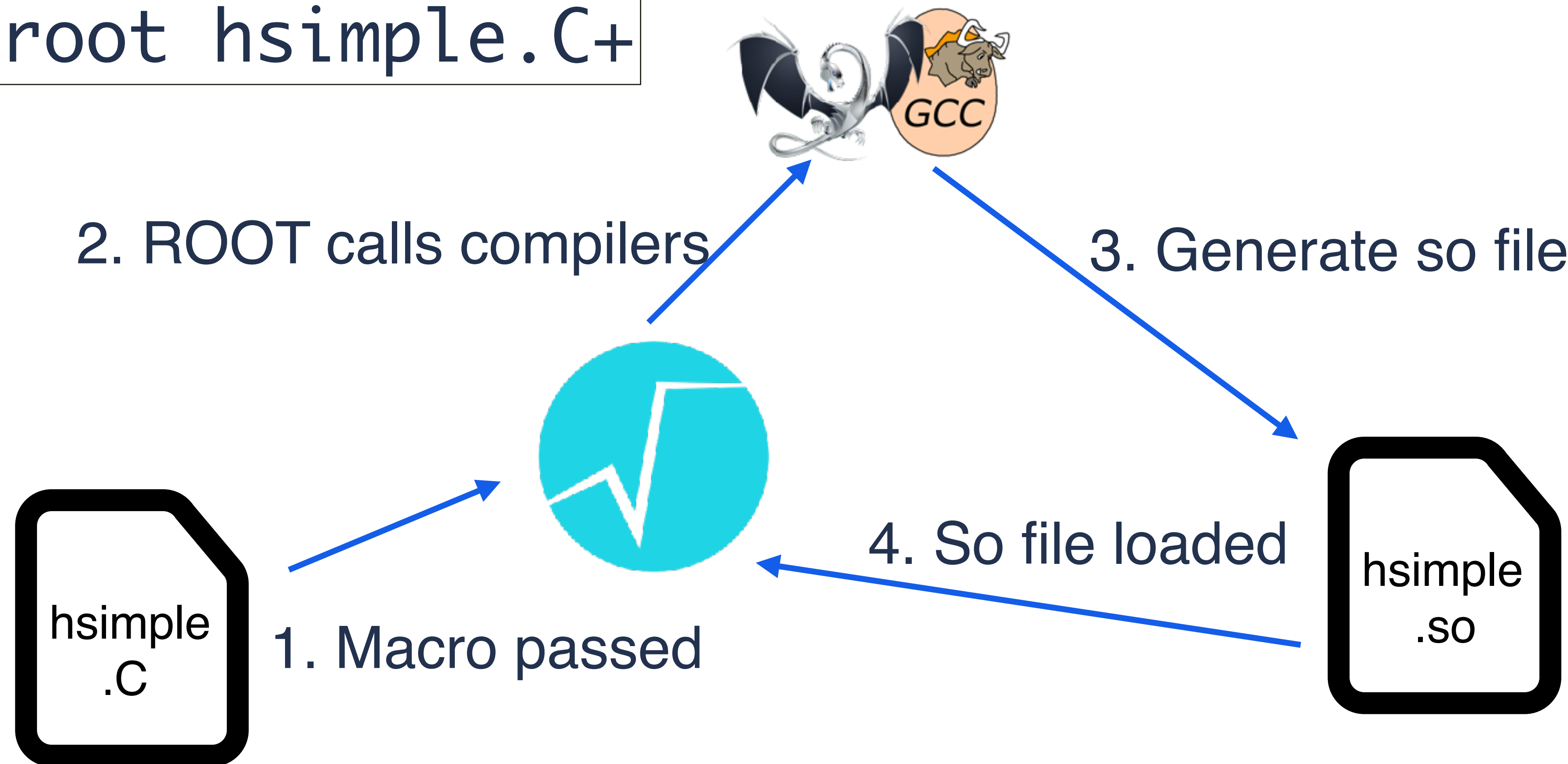


# Summary of older options

## ACLIC

Available since long time ago  
Requires: External compiler

```
$ root hsimple.C+
```



## Summary of older options

### - **config**

```
$ root -config
```

Available since long time ago  
Requires: Nothing :)

Use for debugging purpose to see cmake CACHE\_VARIABLES

```
$ root -config  
ROOT ./configure options:  
BLAS_Accelerate_LIBRARY=/System/Library/Frameworks/  
Accelerate.framework FFTW_INCLUDE_DIR=/usr/local/include  
FFTW_LIBRARY=/usr/local/lib/libfftw3.dylib  
GL2PS_INCLUDE_DIR=/usr/local/include GL2PS_LIBRARY=/usr/  
local/lib/libgl2ps.dylib JPEG_INCLUDE_DIR=/usr/local/  
include...
```



# Summary of fairly new options

# Summary of older options

## - - notebook

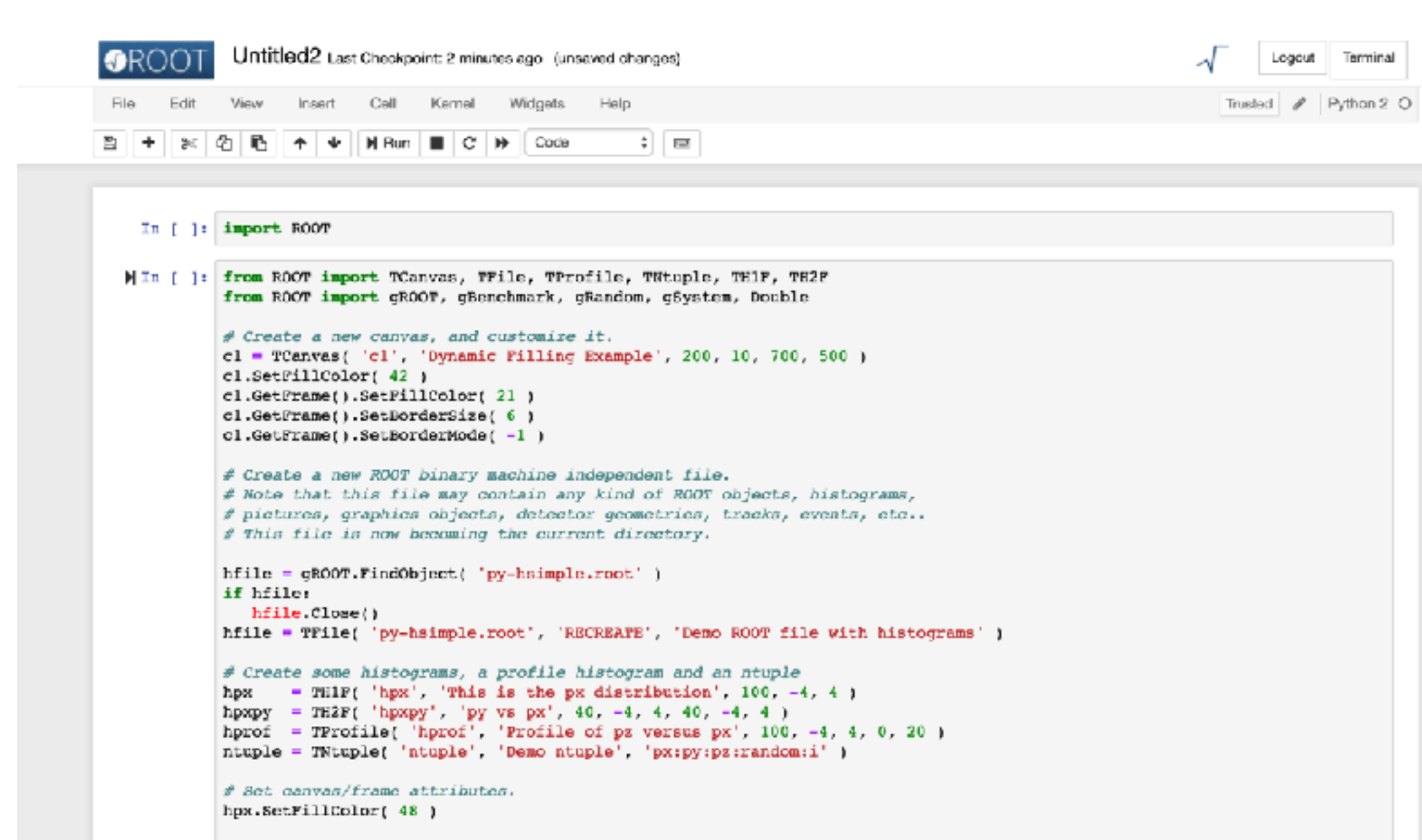
```
$ root --notebook
```

Available since 6.07.06

Requires: Jupyter to be installed

```
$ sudo pip install jupyter
```

## Activate both ROOT C++ and python



```
ROOT Untitled2 Last Checkpoint: 2 minutes ago (unsaved changes) Logout Terminal
File Edit View Insert Call Kernel Widgets Help Trusted Python 2
In [ ]: import ROOT
In [ ]: from ROOT import TCanvas, TFile, TProfile, TTuple, TH1F, TH2F
        from ROOT import gROOT, gBenchmark, gRandom, gSystem, Double

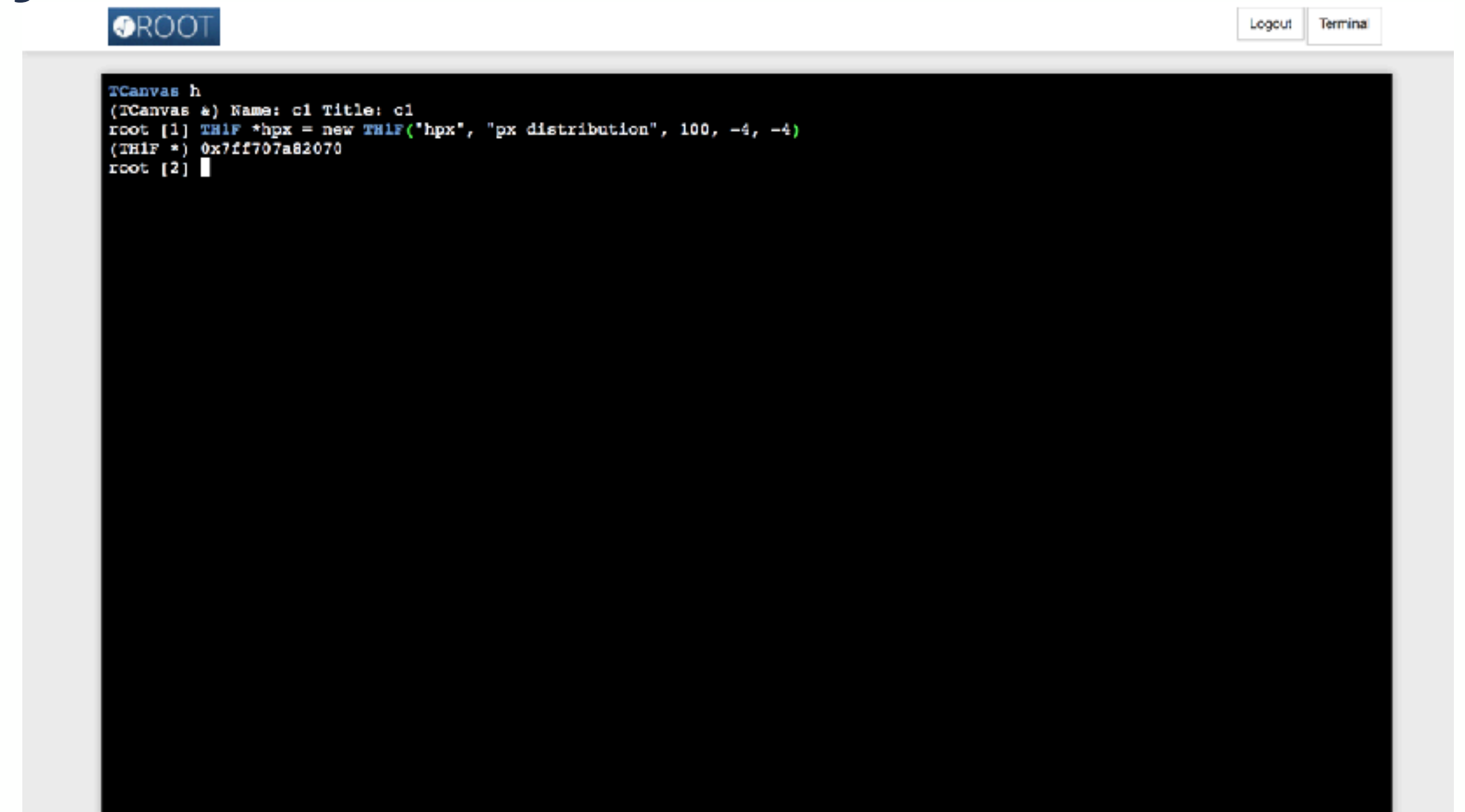
# Create a new canvas, and customize it.
c1 = TCanvas( 'c1', 'Dynamic Filling Example', 200, 10, 700, 500 )
c1.SetFillColor( 42 )
c1.GetFrame().SetFillColor( 21 )
c1.GetFrame().SetBorderSize( 6 )
c1.GetFrame().SetBorderMode( -1 )

# Create a new ROOT binary machine independent file.
# Note that this file may contain any kind of ROOT objects, histograms,
# pictures, graphics objects, detector geometries, tracks, events, etc..
# This file is now becoming the current directory.

hfile = gROOT.FindObject( 'py-hsimple.root' )
if hfile:
    hfile.Close()
hfile = TFile( 'py-hsimple.root', 'RECREATE', 'Demo ROOT file with histograms' )

# Create some histograms, a profile histogram and an ntuple
hpx = TH1F( 'hpx', 'This is the px distribution', 100, -4, 4 )
hpxy = TH2F( 'hpxy', 'py vs px', 40, -4, 4, 40, -4, 4 )
hproF = TProfile( 'hproF', 'Profile of pz versus px', 100, -4, 4, 0, 20 )
ntuple = TTuple( 'ntuple', 'Demo ntuple', 'px:py:pz:random:i' )

# Set canvas/frame attributes.
hpx.SetFillColor( 48 )
```



```
ROOT Logout Terminal
TCanvas h
(TCanvas *) Name: c1 Title: c1
root [1] TH1F *hpx = new TH1F("hpx", "px distribution", 100, -4, -4)
(TH1F *) 0x7ff707a82070
root [2]
```

Refer to [https://root.cern.ch/notebooks/HowTos/HowTo\\_ROOT-Notebooks.html](https://root.cern.ch/notebooks/HowTos/HowTo_ROOT-Notebooks.html) for more information.



## Summary of fairly new options

**-t**

```
$ root -t hsimple.C
```

Available since 6.10

Requires: -Dimt=ON (by default)

```
$ cmake ../root -Dimt=ON
```

Equivalent to

```
$ root  
root [0] ROOT::EnableImplicitMT()
```

Enable multi-threading in

- RDataFrame
- TTree read and write
- TMVA training
- Fitting

Refer to [Danilo's talk](#) for more information

# Summary of fairly new options

- - web

Available since 6.14

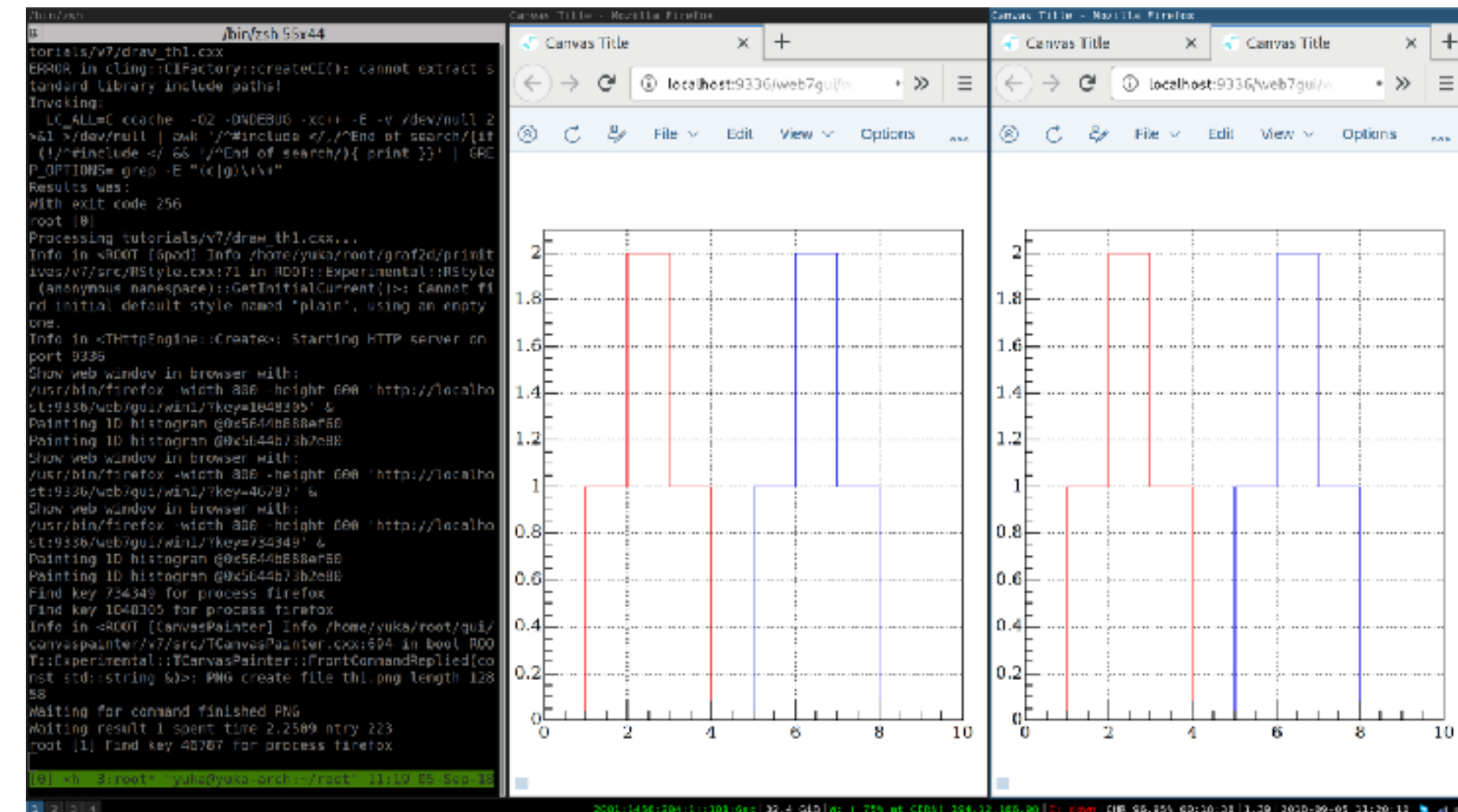
Requires: C++14 (root 7)

```
$ cmake ../root -Dcxx14=ON
```

```
$ root --web firefox draw_th1.cxx
```

- Specify the browser to display graphics
- The default is chrome

Refer to [Sergey's talk](#) for more information



# Strict option

## Strict option

- - **strict**

Work in progress!

Will be available in 6.16 (or 6.18)

Requires: Nothing :)

Use **prompt** with **proper C++!**



## Strict option

- - **strict**

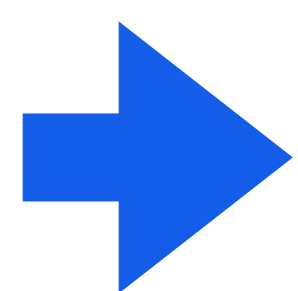
Work in progress!

Will be available in 6.16 (or 6.18)

Requires: Nothing :)

ROOT backend interpreter supports a superset of C++

```
$ root  
root [0] vector<int> v = {1, 2, 3}  
(std::vector<int> &) { 1, 2, 3 }
```



This is done by an effort of C++ interpreter

**It's not valid C++!**

## Strict option

- - **strict**

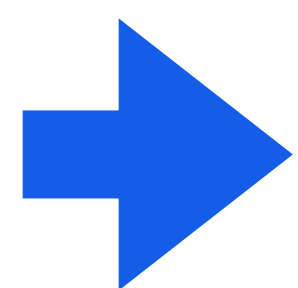
Work in progress!

Will be available in 6.16 (or 6.18)

Requires: Nothing :)

ROOT backend interpreter supports a superset of C++

```
$ root  
root [0] vector<int> v = {1, 2, 3}  
(std::vector<int> &) { 1, 2, 3 }
```



This superset support is a nice feature.. but  
**May mess up your code when debugging**

## Strict option

- - **strict**

Work in progress!

Will be available in 6.16 (or 6.18)

Requires: Nothing :)

--strict **disables** C++ superset supports

```
$ root --strict
```

```
root [0] int a = 1; ← Valid in C++
```

```
root [1] a
```

```
(int) 1
```

```
root [2] std::vector<int> b; ← NOT Valid C++  
(Only works in ROOT)
```

```
input_line_7:2:7: error: no member named 'vector' in namespace  
'std'
```

```
std::vector<int> b;
```

```
~~~~~^
```

```
...
```

```
root [3] #include <vector> ← Valid if you #include vector
```

```
root [4] std::vector<int> b;
```

```
root [5] b = {1,2,3}
```

```
(std::vector &) { 1, 2, 3 }
```

## Strict option (1/3)

List of `--strict` does NOT support = C++ superset support in ROOT

- Auto auto

```
root [0] i = 12 // Interpreted as "auto i = 12"
```

- PCH

```
root [0] TString s; // Without #include
```

- using namespace std

```
root [0] string s; // Instead of std::string
```

- Eval print

```
root [0] 40+2 // without semicolon at the end  
(int) 42
```

## Strict option (2/3)

List of `--strict` does NOT support = C++ superset support in ROOT

- Auto loading TFile objects

```
root [0] TFile::Open("tutorials/hsimple.root");  
root [1] hpx->Draw();
```

Info in <TCanvas::MakeDefCanvas>: created default TCanvas with name c1

- Auto loading

```
root [0] TTree t;
```

Info in <TMacOSXSystem::Load>: loaded library  
/Users/axel/build/root/cmake/lib/libTree.so, status 0

## Strict option (3/3)

List of `--strict` does NOT support = C++ superset support in ROOT

- Auto-parsing

```
root [0] gSystem->Load("libGeom")
```

```
Info in <TMacOSXSystem::Load>: loaded library
```

```
/Users/axel/build/root/cmake/lib/libGeom.so, status 0...
```

```
root [1] TGeoManager g; // triggers auto-parsing!
```

```
Info in <TInterpreter::AutoParse>: Parsing full payload for TGeoManager
```

- Ptr check

```
root [0] int *p = (int*)(0x120 + 0x3);
```

```
root [1] *p
```

```
ROOT_prompt_1:1:2: warning: invalid memory pointer passed to a callee:
```

```
*p  
^
```

# Conclusion

- Summary of old and new options
  - -memstat, ACLiC, -config
  - - - notebook, -t, - - web
- **Strict option**
  - Eliminate interpreter support of C++ superset
  - Enhance C++ code quality
  - Debugging purpose
  - “Pure” Cling



**Stay tuned for 6.16!**



**Thank you for your  
attention!**



# Backup Slides



# Strict

```
[yuka@yuka-arch module-release]$ root --pedantic -q -l
***** CLING *****
* Type C++ code and press enter to run it *
*           Type .q to exit           *
*****
[cling]$ vector<int> a = {1, 2, 3}
input_line_4:2:2: error: use of undeclared identifier 'vector'
vector<int> a = {1, 2, 3}
^
input_line_4:2:12: error: expected '(' for function-style cast or type construction
vector<int> a = {1, 2, 3}
      ~~~^
input_line_4:2:14: error: use of undeclared identifier 'a'
vector<int> a = {1, 2, 3}
      ^
[cling]$ #include <vector>
[cling]$ vector<int> b = {1, 2, 3}
input_line_6:2:2: error: use of undeclared identifier 'vector'
vector<int> b = {1, 2, 3}
^
input_line_6:2:12: error: expected '(' for function-style cast or type construction
vector<int> b = {1, 2, 3}
      ~~~^
input_line_6:2:14: error: use of undeclared identifier 'b'
vector<int> b = {1, 2, 3}
      ^
[cling]$ std::vector<int> b = {1, 2, 3}
(std::vector<int> &) { 1, 2, 3 }
[cling]$
```

