



Интерактивная платформа для удаленной разработки и обмена исследовательскими данными



Докладчик: Самохин Н.Ю., инженер отдела ОВС ОФВЭ

При решении современных научных задач, практически во всех случаях, используются специализированные программные пакеты. Это могут быть и широко распространённые пакеты математической обработки данных с широкой специализацией (MatLab, Mathematica, Ansys и др.), так и узкоспециализированные программные продукты (например, пакеты AliRoot, FairRoot, разработанные для конкретных экспериментальных установок в области релятивистской ядерной физики). Установка этих пакетов на локальные ресурсы может быть связана с такими трудностями как лицензионные ограничения, технические ограничения по «мощности» или спецификации локальных ресурсов, технические сложности в установке

Что такое Jupyter?

Jupyter — это проект с открытым исходным кодом, платформа, которая помогает максимально просто получить собственную web-среду разработки, не задумываясь о локальных пакетах и развертывании. С помощью Jupyter можно не только осуществлять обработку данных, но и делиться результатами с другими

- Различные языки программирования (kernels)
- Интерактивный код
- Визуализация результатов

- редактировать код в браузере, с подсветкой синтаксиса, автоотступами и автодополнением
- запускать код в браузере
- отображать результаты вычислений с медиа представлением (схемы, графики)
- работать с языком разметки Markdown и LaTeX
- работа с Git

- Все в одном в файле *.ipynb [iPython Notebook]
- Метаданные в формате JSON
- Возможность экспорта в HTML, PDF
- Готовая к работе инфраструктура (опыт не требуется)

Поддерживаемые ядра



50+ languages supported.

More complete list at

<https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>



Logout

Control Panel

File Edit View Insert Cell Kernel Widgets Help

Trusted

python3_with_ROOT

Code nbdiff

In [1]: `import ROOT`

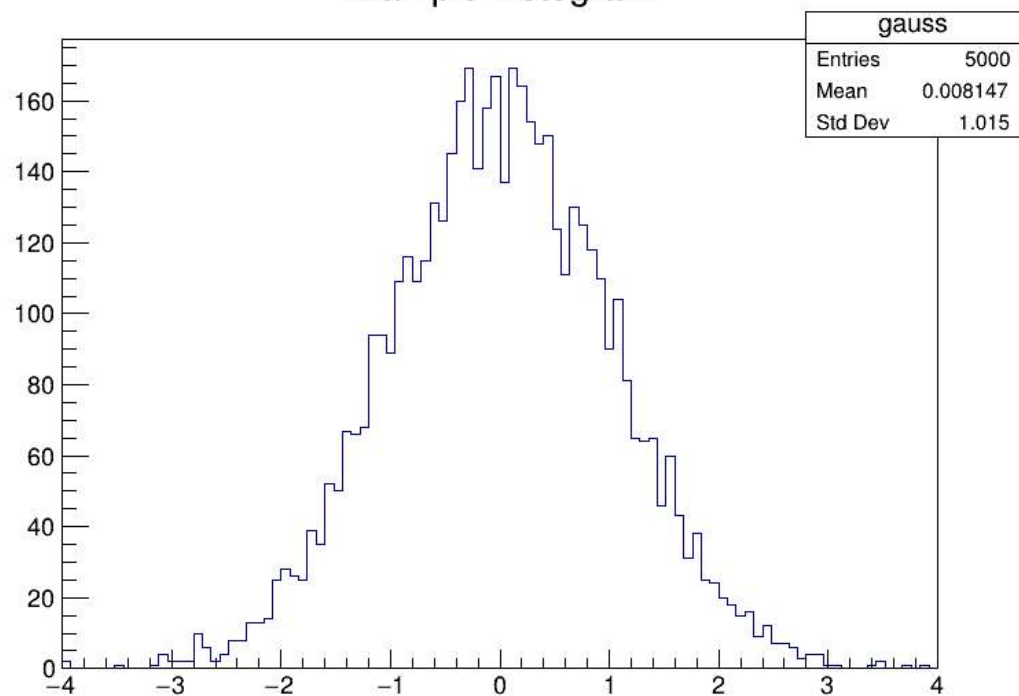
Welcome to JupyROOT 6.22/02

In [2]: `h = ROOT.TH1F("gauss", "Example histogram", 100, -4, 4)`
`h.FillRandom("gaus")`

In [3]: `c = ROOT.TCanvas("myCanvasName", "The Canvas Title", 800, 600)`
`h.Draw()`

In [4]: `c.Draw()`

Example histogram



Лаборатории

- Synchrotron Soleil
- CERN
- European Southern Observatory
- J-PARC MLF
- MAX IV Synchrotron
- European Spallation Source
- Brookhaven National Laboratory
- EuXFEL

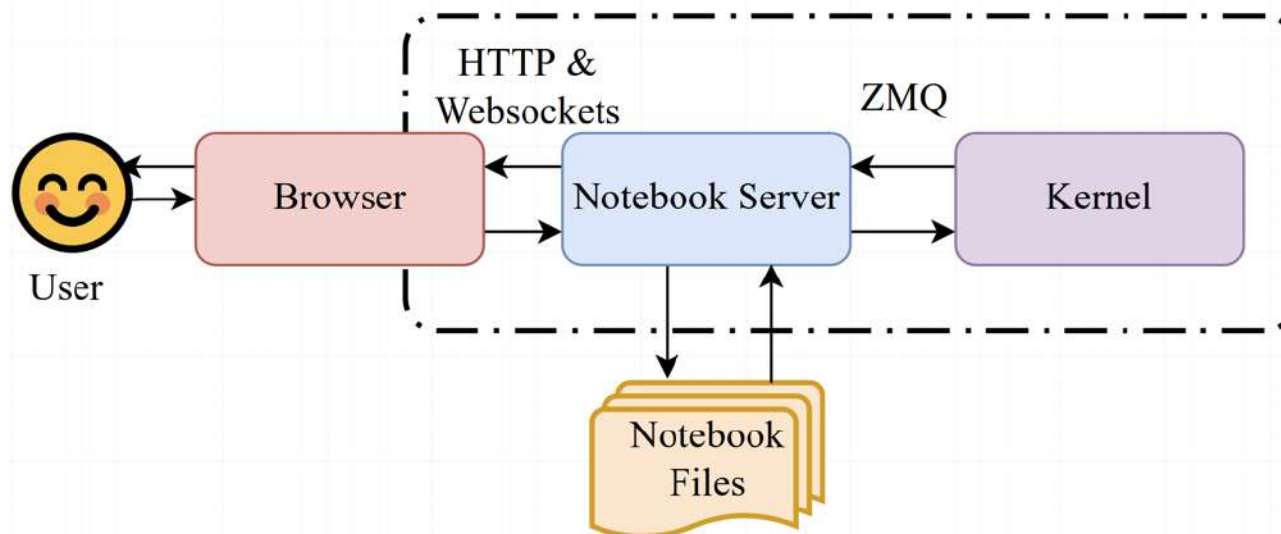
- CERN: SWAN (Jupyter + CernBox + Apache Spark + EOS), с 2016
- Google: Colaboratory (Jupyter + Tensorflow + Google Disk)
- Netflix: ноутбуки для высоконагруженных систем (поточковый видеоконтент по запросу)
- Yandex: аналитика и совместная Python-разработка
- СПбГУ: аналог CERN SWAN, с 2018
- UC Berkeley: самый большой курс по data science в мире, 1200+ студентов



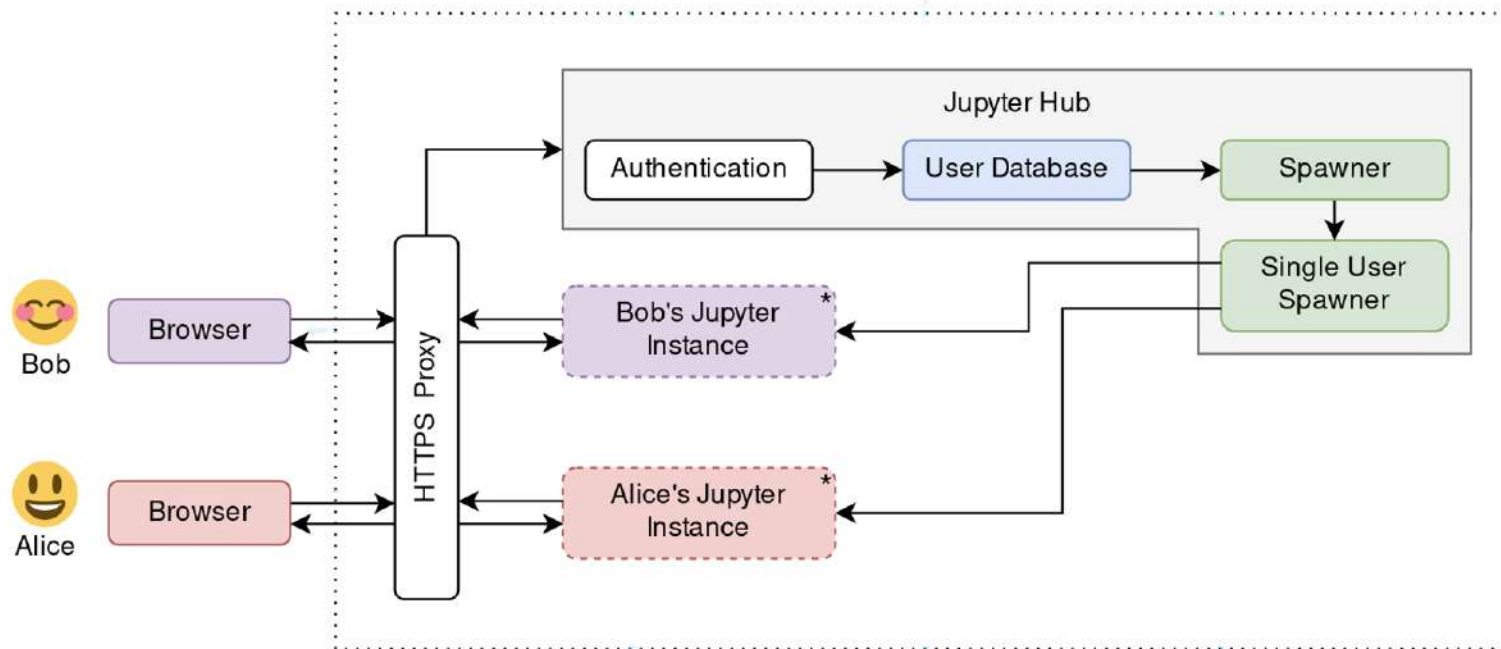
JupyterHub



- Персональный ноутбук-сервер для пользователя
- Инициализация ядра (kernel) при подключении к ноутбуку
- Общение между ядром и сервером посредством ZMQ
- Сервер использует браузер в качестве интерфейса (JavaScript)



JupyterHub - персональный инстанс ноутбука для нескольких пользователей



Пример 1

```
In [35]: import numpy as np
import matplotlib.pyplot as plt
plt.ion()
```

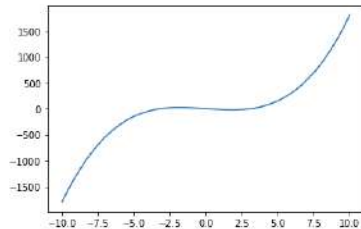
```
"Hello world"
```

```
Out[35]: 'Hello world'
```

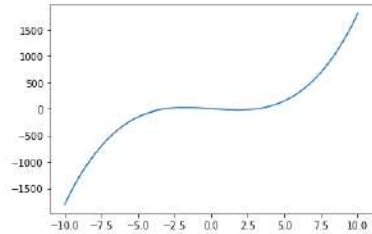
```
In [36]: def polynom(x):
    return 2 * x**3 - 20 * x + 2
```

```
X = np.linspace(-10, 10)
Y = polynom(X)
plt.plot(X, Y)
```

```
Out[36]: [<matplotlib.lines.Line2D at 0x7fe2bd1a2c10>]
```



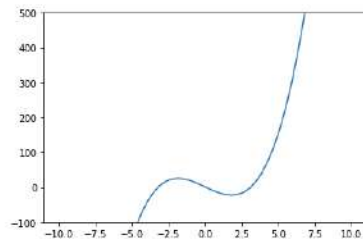
Out[36]: [



In [37]: `from ipywidgets import interact, FloatSlider`

```
def parabolic(offset):  
    X = np.linspace(-10, 10)  
    Y = polynomial(X-offset)  
    # use same y scale for all offsets  
    plt.gca().set_ylim([-100, 500])  
    plt.plot(X, Y);  
  
interact(parabolic, offset=FloatSlider(min=-10., max=10., step=0.25))
```

offset



Out[37]: `<function __main__.parabolic(offset)>`

Пример 2

```
In [ ]: ▶ import ROOT
from sys import exit

try:
    import numpy as np
except:
    print("Failed to import numpy.")
    exit()

# Helper function to create an example tree
def make_example():
    root_file = ROOT.TFile("pyroot002_example.root", "RECREATE")
    tree = ROOT.TTree("tree", "tutorial")
    x = np.empty((1), dtype="float32")
    y = np.empty((1), dtype="float32")
    tree.Branch("x", x, "x/F")
    tree.Branch("y", y, "y/F")

    for i in range(4):
        x[0] = i
        y[0] = -i
        tree.Fill()
    root_file.Write()

    return (root_file, x, y), tree
```



```
print("Tree converted to a pandas.DataFrame:\n{}".format(df))
```

Tree content:

```
[[ 0.  0.]  
 [ 1. -1.]  
 [ 2. -2.]  
 [ 3. -3.]]
```

Tree converted to a numpy array:

```
[[ 0.  0.]  
 [ 1. -1.]  
 [ 2. -2.]  
 [ 3. -3.]]
```

Return numpy array and labels:

```
['x', 'y']  
[[ 0.  0.]  
 [ 1. -1.]  
 [ 2. -2.]  
 [ 3. -3.]]
```

Mean of the columns retrieved with a numpy method: [1.5 -1.5]

Only the content of the branch 'x':

```
[0. 1. 2. 3.]
```

Read all branches except 'x':

```
[ 0. -1. -2. -3.]
```

Return numpy array with data-type 'int':

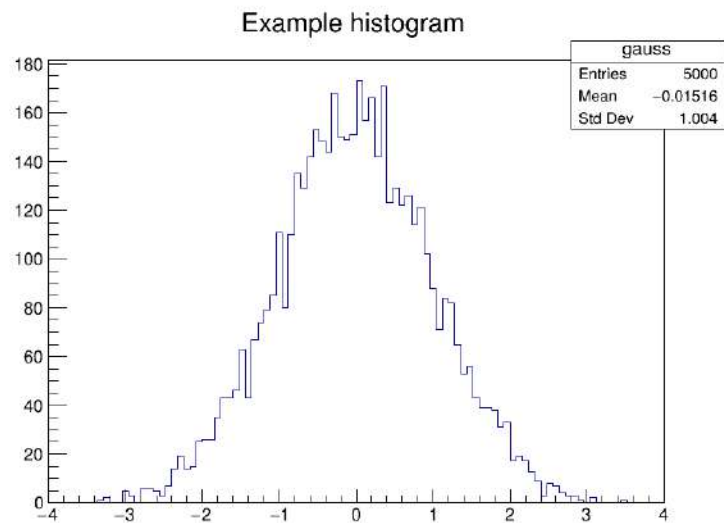
```
[[ 0  0]  
 [ 1 -1]  
 [ 2 -2]  
 [ 3 -3]]
```

Пример 3

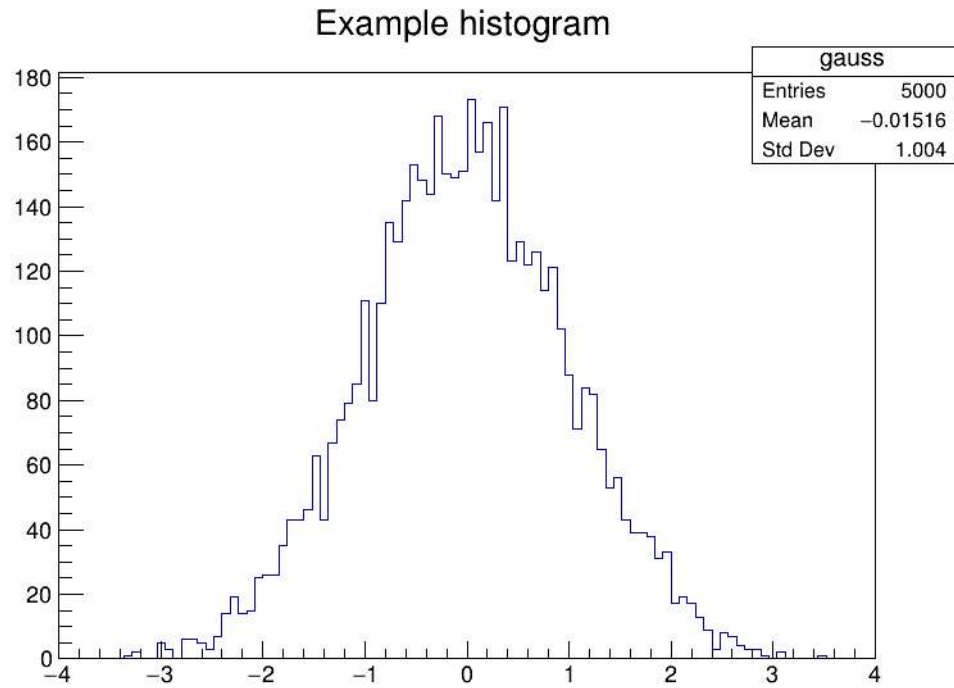
```
In [39]: import ROOT
h = ROOT.TH1F("gauss", "Example histogram", 100, -4, 4)
h.FillRandom("gaus")
c = ROOT.TCanvas("myCanvasName", "The Canvas Title", 800, 600)
```

Warning in <TCanvas::Constructor>: Deleting canvas with same name: myCanvasName

```
In [40]: h.Draw()
c.Draw()
```



```
In [40]: ▶ h.Draw()  
c.Draw()
```



```
In [42]: ▶ %%cpp -a  
class CompileMe {  
public:  
    CompileMe() {}  
    void run() {}  
};
```

```
In [43]: import sys
!{sys.executable} -m pip install pandas
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in /home/llidd/.local/lib/python3.8/site-packages (1.2.3)
Requirement already satisfied: numpy>=1.16.5 in /opt/anaconda3/envs/py_root/lib/python3.8/site-packages (from pandas) (1.19.1)
Requirement already satisfied: pytz>=2017.3 in /home/llidd/.local/lib/python3.8/site-packages (from pandas) (2020.5)
Requirement already satisfied: python-dateutil>=2.7.3 in /opt/anaconda3/envs/py_root/lib/python3.8/site-packages (from pandas) (2.8.1)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/envs/py_root/lib/python3.8/site-packages (from python-dateutil>=2.7.3->pandas) (1.15.0)
```

IDE: JupyterLab

- Интерфейс для ноутбуков нового поколения
- Среда разработки внутри браузера
- Упрощенное отображение и кастомизация интерфейса
- Дополнительные инструменты - расширения, плагины, просмотр CSV, терминал
- Drag/Drop для всех панелей (включая код)

notebook --> jupyterlab

File Edit View Run Kernel Tabs Settings Help

+

+

+

Name	Last Modified
/	
gith	8 days ago
my_notebooks	14 days ago
pres	16 minutes ago
cpp example.ipynb	14 days ago
jupyterhub_cookie_secret	4 months ago
jupyterhub-proxy.pid	11 days ago
jupyterhub.sqlite	14 minutes ago

Presentation.ipynb x cpp example.ipynb x

Code v

```
[1]: class Rectangle {
      private:
          double w;
          double h;

      public:

          Rectangle(double w_, double h_) {
              w = w_;
              h = h_;
          }
          double area(void) {
              return w * h;
          }
          double perimeter(void) {
              return 2 * (w + h);
          }
      };
```

```
[2]: Rectangle r = Rectangle(5, 4);
      r.area()
```

```
[2]: 20.000000
```

```
[ ]:
```

SEARCH

CONSOLE

- Change Kernel...
- Clear Console Cells
- Close and Shut Down...
- Insert Line Break
- Interrupt Kernel
- New Console
- Restart Kernel...
- Run Cell (forced)
- Run Cell (unforced)
- Show All Kernel Activity

EXTENSION MANAGER

- Enable Extension Manager (experimental)

FILE OPERATIONS

- Autosave Documents
- Open from Path...
- Reload from Disk
- Revert to Checkpoint
- Save Ctrl+S
- Save As... Ctrl+Shift+S
- Show Active File in File Browser
- Trust HTML File

HELP

- Jupyter Reference
- JupyterLab FAQ
- JupyterLab Reference
- Launch Classic Notebook
- Markdown Reference
- Reset Application State

HUB

- Hub Control Panel
- Log Out

Presentation.ipynb × cpp example.ipynb × Launcher ×

Notebook

- Python 3
- C++11
- C++14
- C++17
- Python 2
- python3_with_R OOT

Console

- Python 3
- C++11
- C++14
- C++17
- Python 2
- python3_with_R OOT

Other

- Terminal
- Text File
- Markdown File
- Show Contextual Help

File Edit View Run Kernel Tabs Settings Help

0_JUPYTER_AND_COLAB.IPYNB

0_Jupyter_and_Colab.ipynb x 1_Python_code_EXERCISES.ipynb x

Multiple tabs

1. Jupyter Notebooks and Colab

Jupyter Notebooks are a widely used tool for data science. They contain both code and rich-text elements like paragraphs, equations and charts. Notebooks are both (a) human-readable documents used to present analysis and results; and (b) executable documents that your computer can run.

1.0.1. Important shortcuts

1.0.1.1. Types of cell

Notebooks are made up of a series of cells. There are two types:

- **Code cells.** These contain executable commands in Python.
- **Text ('markdown') cells.** These include plain text, or you can use markdown conventions to add formatting.

```
[6]: print('Hello, I am a code cell')
print('Running me executes the code and renders the output below.')
```

Hello, I am a code cell
Running me executes the code and renders the output below.

Table of Contents extension

Dark theme!

Action	Colab Shortcut
Executes current cell	<CTRL-ENTER>
Executes current cell and moves to next cell	<SHIFT-ENTER>
Insert cell above	<CTRL-M> <A>
Append cell below	<CTRL-M>
Convert cell to code	<CTRL-M> <Y>
Convert cell to Markdown	<CTRL-M> <M>
Autocomplete	<TAB>
Goes from edit to "command" mode	<ESC>
Goes from "command" to edit mode	<ENTER>
Note: On OS X use <COMMAND> instead of <CTRL>	

rosca@DESKTOP-P9E1B71:~/environments\$

Multiple panels

BLI2015.csv x Explore files (e.g. CSV)

Delimiter: ,

	"LOCATION"	Country	INDICATOR	Indicator
1	AUS	Australia	HO_BASE	Dwellings without basic facilities
2	AUT	Austria	HO_BASE	Dwellings without basic facilities
3	BEL	Belgium	HO_BASE	Dwellings without basic facilities
4	CAN	Canada	HO_BASE	Dwellings without basic facilities
5	CZE	Czech Republic	HO_BASE	Dwellings without basic facilities
6	DNK	Denmark	HO_BASE	Dwellings without basic facilities
7	FIN	Finland	HO_BASE	Dwellings without basic facilities
8	FRA	France	HO_BASE	Dwellings without basic facilities
9	DEU	Germany	HO_BASE	Dwellings without basic facilities
10	GRC	Greece	HO_BASE	Dwellings without basic facilities
11	HUN	Hungary	HO_BASE	Dwellings without basic facilities
12	ISL	Iceland	HO_BASE	Dwellings without basic facilities
13	IRL	Ireland	HO_BASE	Dwellings without basic facilities
14	ITA	Italy	HO_BASE	Dwellings without basic facilities
15	JPN	Japan	HO_BASE	Dwellings without basic facilities
16	KOR	Korea	HO_BASE	Dwellings without basic facilities
17	LUX	Luxembourg	HO_BASE	Dwellings without basic facilities
18	MEX	Mexico	HO_BASE	Dwellings without basic facilities
19	NLD	Netherlands	HO_BASE	Dwellings without basic facilities
20	NZL	Zealand	HO_BASE	Dwellings without basic facilities
21	NOR	Norway	HO_BASE	Dwellings without basic facilities
22	POL	Poland	HO_BASE	Dwellings without basic facilities

1 2

BLI2015.csv

Попробовать

<https://hepd-lmsys001.pnpi.spb.ru:8019>



Sign in

Username:

Password:

Sign In

Do not have an account? Sign up [here](#)

QUESTIONS

E-mail: samon@itmo.ru

Tel: +7-921-869-46-48