Code editor

```
print("Hello, world!")
```
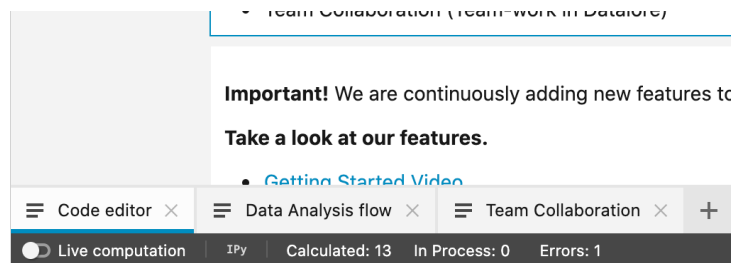
Hello, world!

Getting started tutorial

Datalore is an online data analysis notebook that enables you to edit, execute, and share your data more productively.

In this tutorial, we will learn to work with several **worksheets**:

- Code editor (general Datalore overview)
- Data analysis flow (example of dataset analysis)
- Team collaboration (teamwork in Datalore)

Important! We are always adding new features to Datalore. You can find the most up-to-date tutorial with the newest features here.

Take a look at our features:

- Watch our Getting Started video
- Code editor
  - Task 1:
    - Create a new worksheet
    - Create cells
  - Task 2:
    - Run some code
- Coding assistance
  - Task 3:
    - Code completion
    - Open documentation
    - Inspections and quick-fixes
    - Auto imports
  - Task 4:
    - Adjust notebook look
  - Task 5:
    - Install libraries and packages
  - Task 6:
    - Choose a kernel
    - Create a notebook
- Data analysis flow
  - Collecting data
    - Upload a dataset
  - Exploring data

- Visualizing results
- Sharing insights

- Team collaboration
  - Share notebooks
  - Share workspaces
  - History checkpoints

- Tutorials
- Follow us

Watch our Getting Started video

```
from IPython.display import HTML

HTML('<div align="center"><iframe align = "middle" width="560" height="315" src="https://www.youtube.com/embed/MjvFQxqNSe0" frameborder
```

Code editor

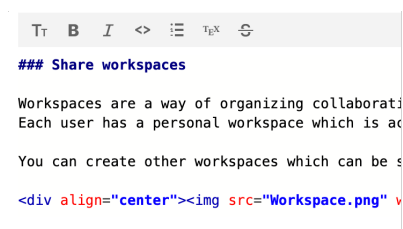Learn how to use the editor by completing the following small tasks.

Task 1:

Create a new worksheet

Click `+` button at the bottom left corner of the editor to create a new worksheet. All sheets in a notebook share the same files, and sheets on the right inherit the environment from the sheets on the left.

Create cells

1. Add a new **code cell** by pressing `Ctrl+Alt+Enter` / `Cmd+Option+Enter` or by clicking on the "+" button in the top right of the cell. You can toggle the cell type between code and Markdown with `Ctrl+M` / `Cmd+M` .

2. Add a **Markdown cell** by hovering over the middle of the bottom of the cell and clicking "Add markdown cell".

   - Markdown cells support Markdown text editing and LaTeX transformations;
   - Double newline clears formatting and starts a new paragraph;
   - All input is HTML-escaped.

3. **Move cells up and down** by pressing "Move cell up" or "Move cell down" in the cell toolbar.

Task 2:

Run some code

1. Execute the code in the following cell by pressing the **Run/Stop buttons** at the top-right corner of the cell **or** just press 'Shift+Enter'.

2. If you encounter any problems, try the **Kernel** → **Restart** or **Kernel** → **Interrupt** commands in the main menu.

```
# Try it now
import datetime

now_moment = datetime.datetime.now()
print('- What is the right time to start?')
print('- Now!\n')
```

Coding assistance

We hope you enjoy Datalore's smart coding assistance and that it helps you work more productively.

Task 3:

Code completion

Add a new code cell and print the `now_moment` variable from the previous code cell. Whenever you type a function, method, class, or variable name, Datalore suggests possible completions.

Open documentation

Try opening the documentation for the pyplot.hist() function in the following cell.

Use:

- `Ctrl/Command + hover` for brief documentation.
- `set the caret + F1` for full documentation.

```
[2]  # Try it now!
     # 1.Run this code cell
     # 2. Add a Markdown block, move it up and write there
     # "Visualisation Cheat S  class datetime @classmethod def now(cls: Type[_S], tz: Optional[tzinfo] = ...) ->
     # Make it a title using   _S Possible types: • (cls: Type[_S], tz: None) -> _S •
     import datetime          (cls: Type[datetime], tz: tzinfo) -> datetime Construct a datetime from time.time()
                              and optional time zone info.
     print('- What is the rig
     print('- Now!\n')         Show more
     print(datetime.datetime.now())


     - What is the right time to start?
     - Now!

     2020-08-21 07:31:46.444630
```

```
# Try opening Documentation for pyplot.hist() function

from matplotlib import pyplot as plt
import numpy as np

mu, sigma = 100, 15
x = mu + sigma * np.random.randn(10000)

# the histogram of the data
n, bins, patches = plt.hist(x, 50, density=1, facecolor='g', alpha=0.75)
plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title('Histogram of IQ')
plt.text(60, .025, r'$\mu=100,\ \sigma=15$')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)
plt.show()
```

Inspections and quick-fixes

Click the string "Probability" in the previous cell and apply the suggested quick-fix. Just press `Alt/Option+Enter` on the error and choose a fix from the popup menu.