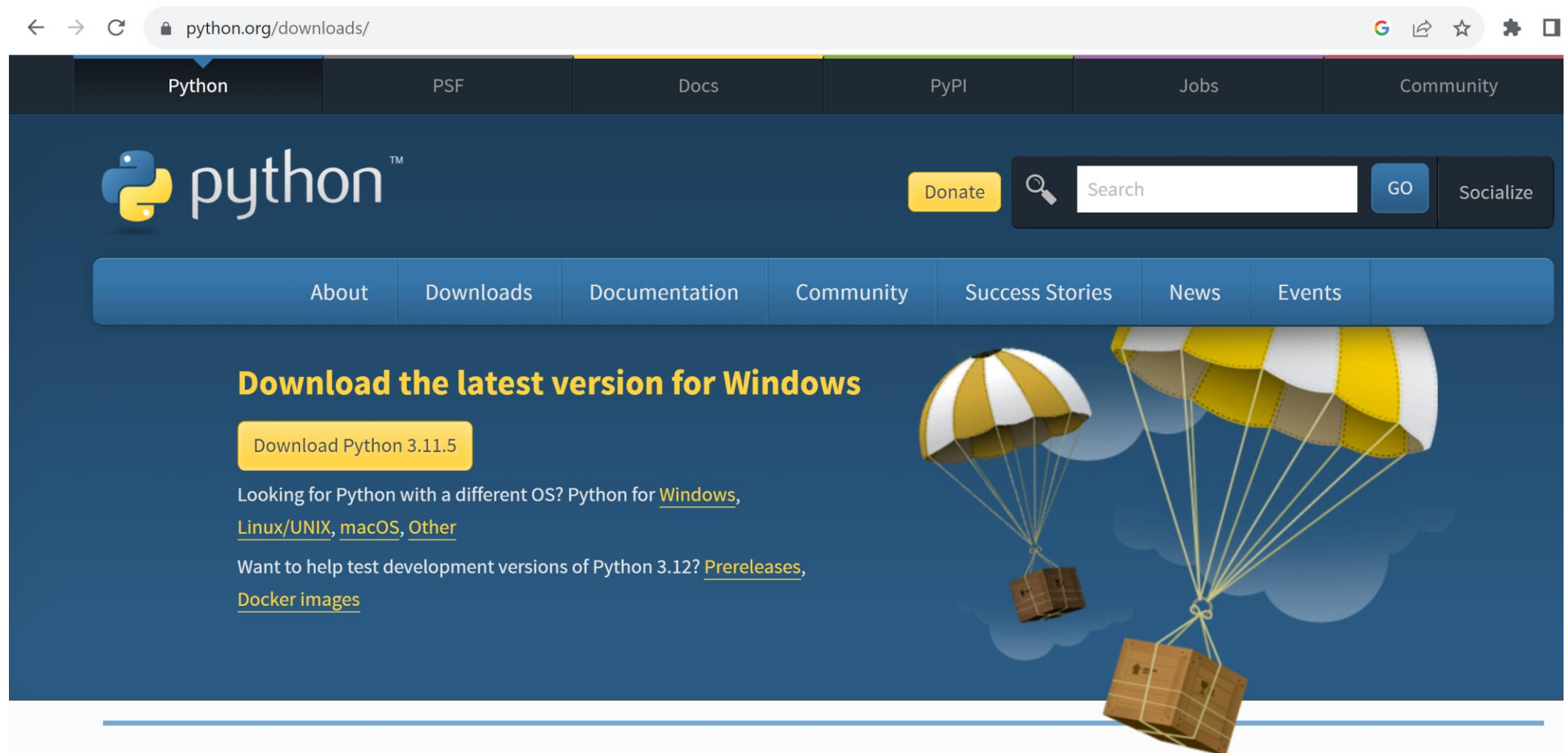


Installing Jupyter Notebook for Spark

Arnaud.nauwynck@gmail.com

Step 1 : install python (simplest method, not using AnaConda)



Step 2 : install jupyter

Choose class Jupyter Notebook... OK



JupyterLab

Install JupyterLab with `pip`:

```
pip install jupyterlab
```

Note: If you install JupyterLab with conda or mamba, we recommend using [the conda-forge channel](#).

Once installed, launch JupyterLab with:

```
jupyter lab
```

Jupyter Notebook

Install the classic Jupyter Notebook with:

```
pip install notebook
```

To run the notebook:

```
jupyter notebook
```

Step 3 : install jupyter spylon_kernel
(for using Spark with Scala language)

python -m spylon_kernel install --user

(or `pip install spylon_kernel --user`)

```
C:\Users\arnaud>python -m spylon_kernel install --user
0.00s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.
[InstallKernelSpec] Installed kernelspec spylon-kernel in C:\Users\arnaud\AppData\Roaming\jupyter\kernels\spylon-k
ernel
```

Star Jupyter Notebook

jupyter notebook

```
C:\Users\arnaud>jupyter notebook
[I 2023-09-06 23:09:00.357 ServerApp] Package notebook took 0.0000s to import
[I 2023-09-06 23:09:00.482 ServerApp] Package jupyter_lsp took 0.1187s to import
[W 2023-09-06 23:09:00.482 ServerApp] A `_jupyter_server_extension_points` function was not found in jupyter_lsp.
Instead, a `_jupyter_server_extension_paths` function was found and will be used for now. This function name will
be deprecated in future releases of Jupyter Server.
[I 2023-09-06 23:09:00.544 ServerApp] Package jupyter_server_terminals took 0.0598s to import
[I 2023-09-06 23:09:00.544 ServerApp] Package jupyterlab took 0.0000s to import
```

[[truncated logs Also contains errors??]]

Jupyter Notebook ..

```
[I 2023-09-06 23:09:01.879 ServerApp] jupyterlab | extension was successfully loaded.  
[I 2023-09-06 23:09:01.879 ServerApp] notebook | extension was successfully loaded.  
[I 2023-09-06 23:09:01.895 ServerApp] Serving notebooks from local directory: C:\Users\arnaud  
[I 2023-09-06 23:09:01.895 ServerApp] Jupyter Server 2.7.3 is running at:  
[I 2023-09-06 23:09:01.895 ServerApp] http://localhost:8888/tree?token=a9195d7e950f42e4e26a7b93bbebf3664d1794b94959c5e3  
[I 2023-09-06 23:09:01.911 ServerApp] http://127.0.0.1:8888/tree?token=a9195d7e950f42e4e26a7b93bbebf3664d1794b94959c5e3  
[I 2023-09-06 23:09:01.911 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).  
[C 2023-09-06 23:09:01.974 ServerApp]
```

To access the server, open this file in a browser:

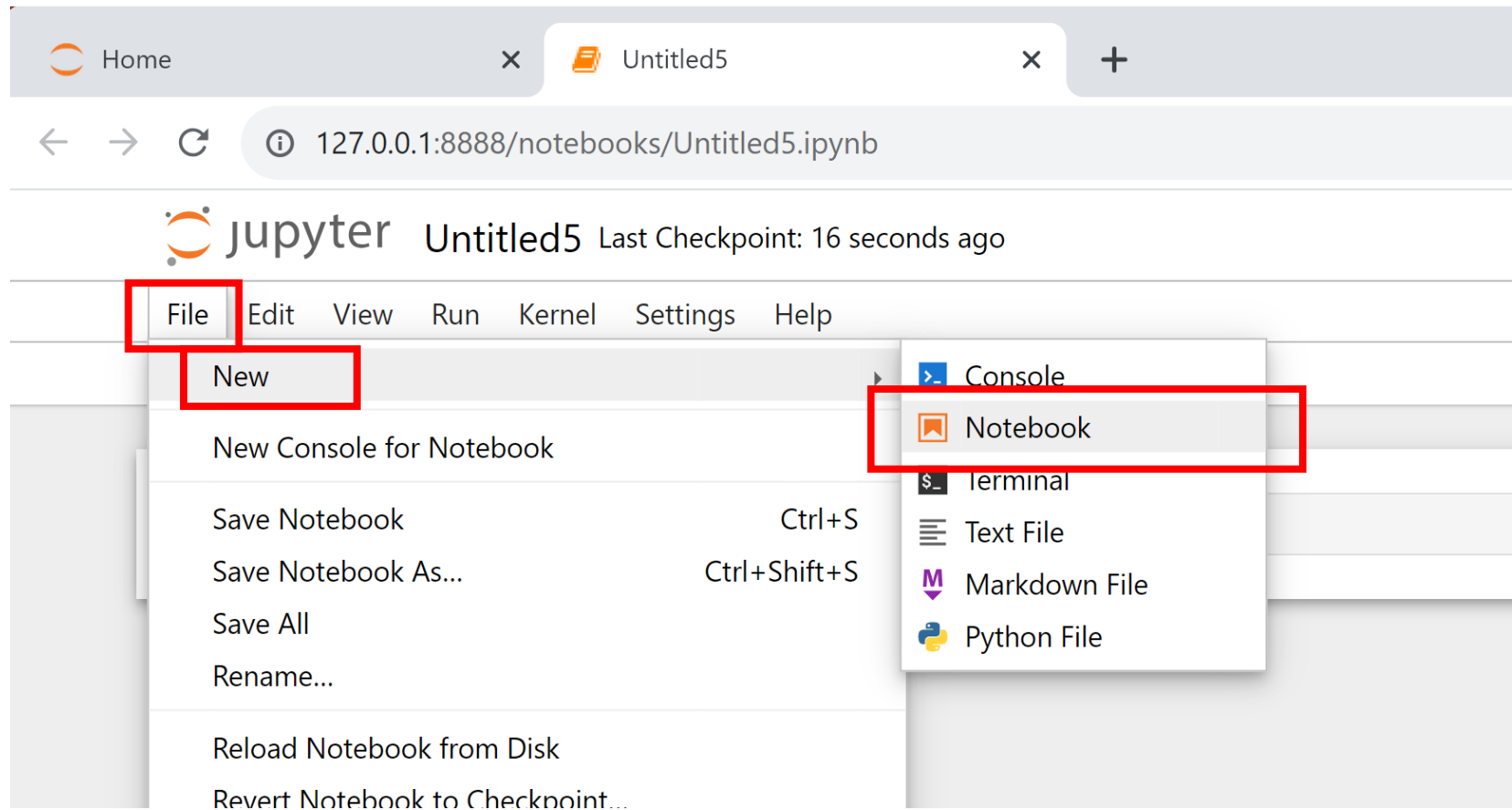
`file:///C:/Users/arnaud/AppData/Roaming/jupyter/runtime/jpserver-636-open.html`

Or copy and paste one of these URLs:

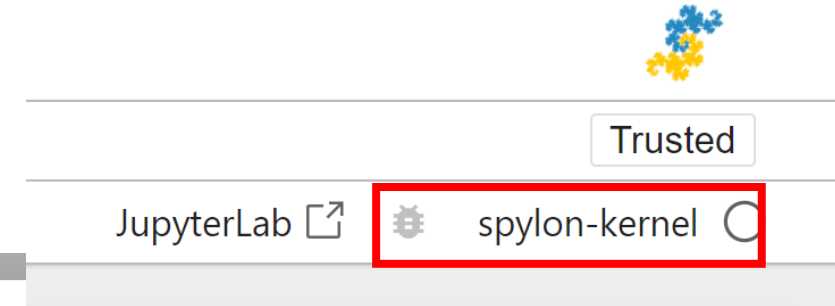
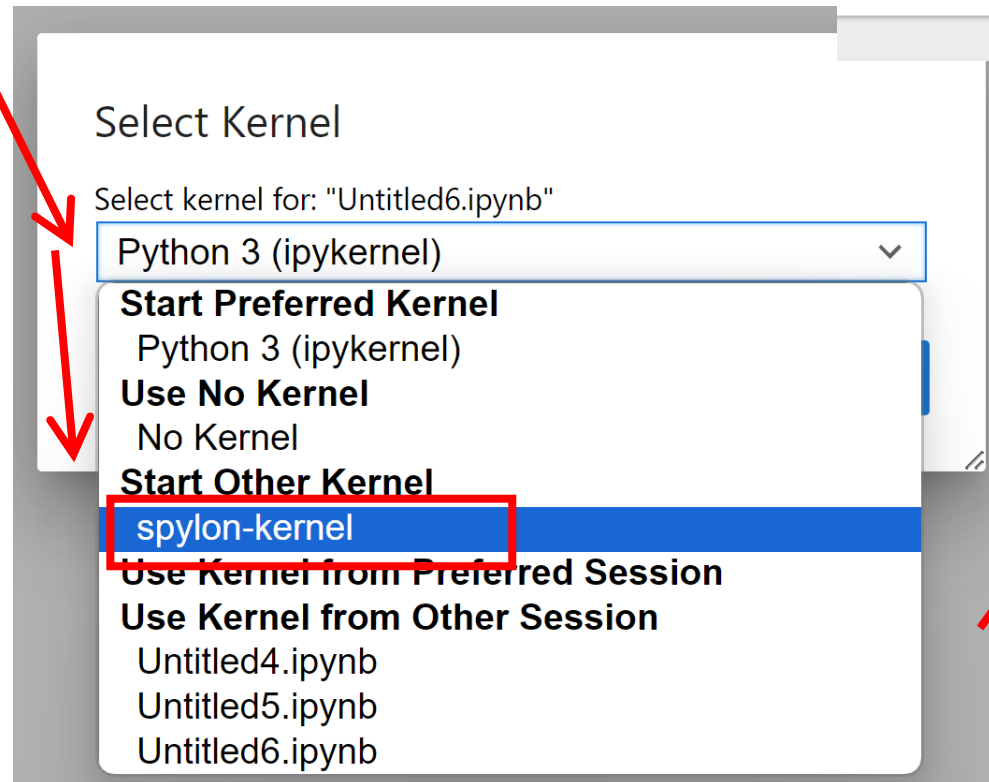
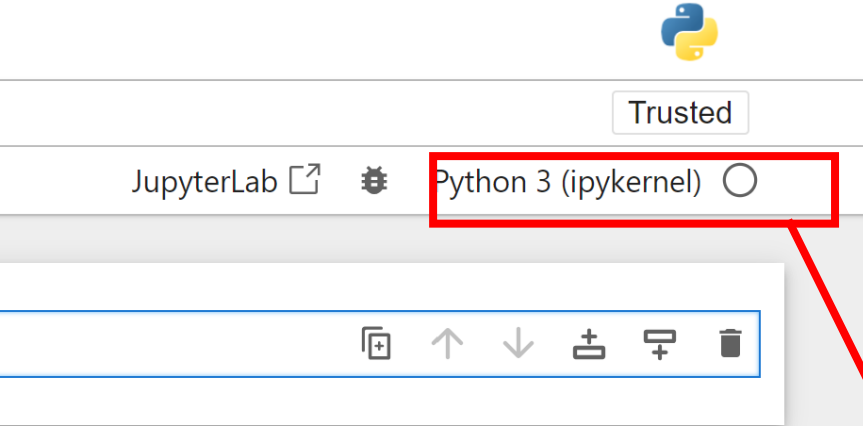
`http://localhost:8888/tree?token=a9195d7e950f42e4e26a7b93bbebf3664d1794b94959c5e3`

`http://127.0.0.1:8888/tree?token=a9195d7e950f42e4e26a7b93bbebf3664d1794b94959c5e3`

Creating a new notebook



Step 3 ... testing spylon



Step 3 ... Testing Spylon .. Write Scala

The diagram illustrates the process of testing Spylon by writing Scala code. It shows a code editor with the following Scala code:

```
[1]: for(i <- 0 to 5) println(s"Scala code.. ${i}")
```

Below the code, a red box highlights the text "Intitilizing Scala interpreter ...".

To the right of the code editor, a toolbar contains a "Run" button (a play icon) and a "Code" dropdown menu. Below the toolbar, a text box says "Run this cell and advance (Shift+Enter)".

Below the code editor, a console output shows the results of running the code:

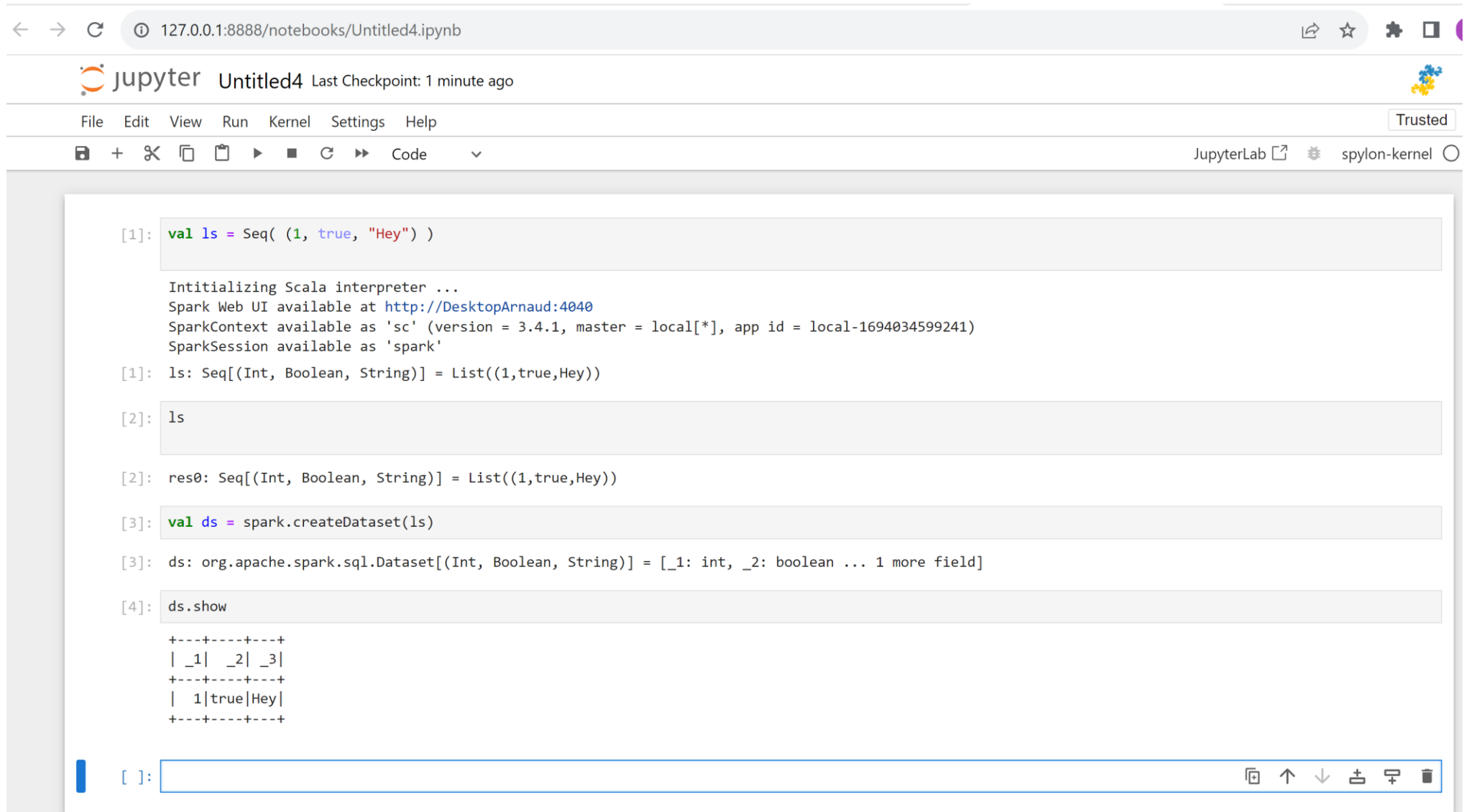
```
[1]: for(i <- 0 to 5) println(s"Scala code.. ${i}")  
Intitilizing Scala interpreter ...  
Spark Web UI available at http://DesktopArnaud:4041  
SparkContext available as 'sc' (version = 3.4.1, master = local[*], app id = local-1694035899154)  
SparkSession available as 'spark'  
Scala code.. 0  
Scala code.. 1  
Scala code.. 2  
Scala code.. 3  
Scala code.. 4  
Scala code.. 5
```

Red arrows indicate the flow of the process: from the code to the "Run" button, and from the "Run" button to the console output.

Get Spark process
(Scala interpreter)

Type SCALA code
... Shift+Enter

Step 3 ... Testing Spylon .. Write Scala



The screenshot shows a JupyterLab notebook titled "Untitled4" with a last checkpoint of 1 minute ago. The interface includes a top bar with navigation icons, a menu bar (File, Edit, View, Run, Kernel, Settings, Help), and a toolbar with various icons. The notebook content consists of several code cells:

```
[1]: val ls = Seq( (1, true, "Hey") )
```

Initializing Scala interpreter ...
Spark Web UI available at <http://DesktopArnaud:4040>
SparkContext available as 'sc' (version = 3.4.1, master = local[*], app id = local-1694034599241)
SparkSession available as 'spark'

```
[1]: ls: Seq[(Int, Boolean, String)] = List((1,true,Hey))
```

```
[2]: ls
```

```
[2]: res0: Seq[(Int, Boolean, String)] = List((1,true,Hey))
```

```
[3]: val ds = spark.createDataset(ls)
```

```
[3]: ds: org.apache.spark.sql.Dataset[(Int, Boolean, String)] = [_1: int, _2: boolean ... 1 more field]
```

```
[4]: ds.show
```

```
+---+-----+
|_1|_2|_3|
+---+-----+
| 1|true|Hey|
+---+-----+
```

The bottom of the notebook shows an empty code cell with the prompt `[]:` and a toolbar with icons for copy, paste, undo, redo, and delete.

Step 4 : install kernel for pyspark...
(for using PySpark with Python language)