

Jupyter Kernels & Supported Languages

Jupyter Slides & RISE

RISE stands for `Reveal.js Jupyter/iPython Slideshow Extension`

RISE is a Jupyter Extension which can turn a Jupyter Notebook into a slide with ease

Installation

Using Conda

```
In [ ]: conda install -c conda-forge rise
```

Using pip

```
In [ ]: pip install RISE
```

Usage

The screenshot displays the JupyterLab interface in presentation mode. At the top, the Jupyter logo is followed by the text "jupyter" and "Untitled (unsaved changes)". To the right is a Python logo and a "Logout" button. Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar are "Trusted" and "Python 3" indicators. A toolbar below the menu bar contains icons for saving, adding, undo, redo, and other editing functions, along with a "Run" button and a "Code" dropdown menu. The main area shows five slides, each with a prompt "In []:" and a "Slide Type" dropdown menu. The slide types are "Slide", "Sub-Slide", "Fragment", "Skip", and "Notes". The "Notes" slide is currently selected, indicated by a blue vertical bar on the left.

jupyter Untitled (unsaved changes)

Python Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Slide Type Slide

Slide Type Sub-Slide

Slide Type Fragment

Slide Type Skip

Slide Type Notes

Exporting your notebook

Switch to the location of the file

Run the command and export

```
In [ ]: jupyter nbconvert your_slide.ipnb
```

nbconvert exports to HTML by default, but you can export your notebook to different formats as TeX, PDF, among others

Jupyter Kernels

Jupyter Kernels can be thought as reference implementations for supported languages

There are around 100 kernels that can be used into Jupyter

Examples

Running Julia

Install Julia

```
In [ ]: sudo apt install julia
```

Import Julia into Jupyter (with Julia console)

```
In [ ]: > using Pkg  
> Pkg.add("IJulia")
```


Sample code in Julia

```
In [1]: function mandelbrot(a)
        z = 0
        for i=1:50
            z = z^2 + a
        end
        return z
    end

    for y=1.0:-0.05:-1.0
        for x=-2.0:0.0315:0.5
            abs(mandelbrot(complex(x, y))) < 2 ? print("*") : print(" ")
        end
        println()
    end
end
```

```
function mandelbrot(a)
```

$$z = 0$$

```
for i=1:50
```

$$z = z^2 + a$$

end

```
return z
```

end

```
for y=1.0:-0.05:-1.0
```

```
for x=-2.0:0.0315:0.5
```

```
abs(mandelbrot(complex(x, y))) < 2 ? print("*") : print(" ")
```

end

```
println()
```

end

**

***** ** *

*** *****

*

*** **

Running R

Installing R language IRKernel

```
In [ ]: sudo apt-get install libzmq3-dev libcurl4-openssl-dev libssl-dev jupyter-core jupyter-client
```

Installing the libraries with R command line

```
In [ ]: > install.packages(c('repr', 'IRdisplay', 'IRkernel'), type = 'source')
```

Importing IRKernel into Jupyter

```
In [ ]: > IRkernel::installspec(user = FALSE)
```

Sample Code in R

```
In [2]: vector1 <- c(1, 6, 9, 63, 4, 5)  
sort(vector1)
```

1 4 5 6 9 63

SoS Kernel

SoS consists of a playground notebook that allows the use of multiple kernels in one Jupyter notebook

Running SoS

Install and Setup with Conda

```
In [ ]: conda install sos-notebook jupyterlab-sos sos-papermill -c conda-forge
```

Install and Setup with pip

```
In [ ]: pip install sos  
pip install sos-pbs  
python -m sos.install
```

Examples

Hello World in JavaScript

```
In [ ]: let hello = ['Hello', 'World'];
```

```
In [ ]: console.log(hello.join(', '))
```


Objects in TypeScript

```
In [ ]: interface Pilot{  
    name: string;  
    age: number;  
    team: string;  
    car_number: number;  
}
```

```
In [ ]: let pilot : Pilot = {  
    name: 'Ayrton Senna',  
    age: 34,  
    team: 'Williams',  
    car_number: 2,  
};
```

```
In [ ]: console.log(pilot)
```

References and More Info

- <https://rise.readthedocs.io/en/stable/index.html>
(<https://rise.readthedocs.io/en/stable/index.html>).
- <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>
(<https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>).
- <https://github.com/vatlab/SOS> (<https://github.com/vatlab/SOS>).
- <https://github.com/yunabe/tslab> (<https://github.com/yunabe/tslab>).
- <https://julialang.org/learning/code-examples/> (<https://julialang.org/learning/code-examples/>).
- <https://julialang.org/downloads/> (<https://julialang.org/downloads/>).
- <https://irkernel.github.io/installation/#linux-panel>
(<https://irkernel.github.io/installation/#linux-panel>).