



# Landing on Jupyter

A guided tour of interactive  
notebooks

*Gregory Farage and Saunak Sen*

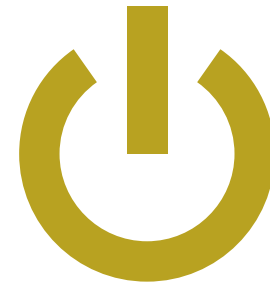
# Outline

- What are Jupyter notebooks?
- Explain what Jupyter kernels are.
- Learn how to create and export a notebook.
- Possible use case demonstrations.
- how to install Jupyter?

# Time to show your programming work!



Raw source code



A compiled executable.

```

57     t.appeared = false;
58     return;
59 }
60 //is the element inside the visible window?
61 var a = w.scrollLeft();
62 var b = w.scrollTop();
63 var o = t.offset();
64 var x = o.left;
65 var y = o.top;
66
67 var ax = settings.accX;
68 var ay = settings.accY;
69 var th = t.height();
70 var wh = w.height();
71 var tw = t.width();
72 var ww = w.width();
73
74 if (y + th + ay >= b &&
75     y <= b + wh + ay &&
76     x + tw + ax >= a &&
77     x <= a + ww + ax) {
78     //trigger the custom event
79     if (!t.appeared) t.trigger('appear', settings.data);
80 } else {
81     //it scrolled out of view
82     t.appeared = false;
83 }
84 };
85
86 //create a modified fn with some additional logic
87 var modifiedFn = function() {
88     //mark the element as visible
89     t.appeared = true;
90
91     //is this supposed to happen only once?
92     if (settings.one) {
93         //remove the check
94         w.unbind('scroll', check);
95         var i = $.inArray(check, $.fn.appear.checks);
96         if (i >= 0) $.fn.appear.checks.splice(i, 1);
97     }
98 }

```

Photo by [Markus Spiske](#) on [Unsplash](#)

What your raw code may  
look like for programmers  
who know the language.



Raw source code



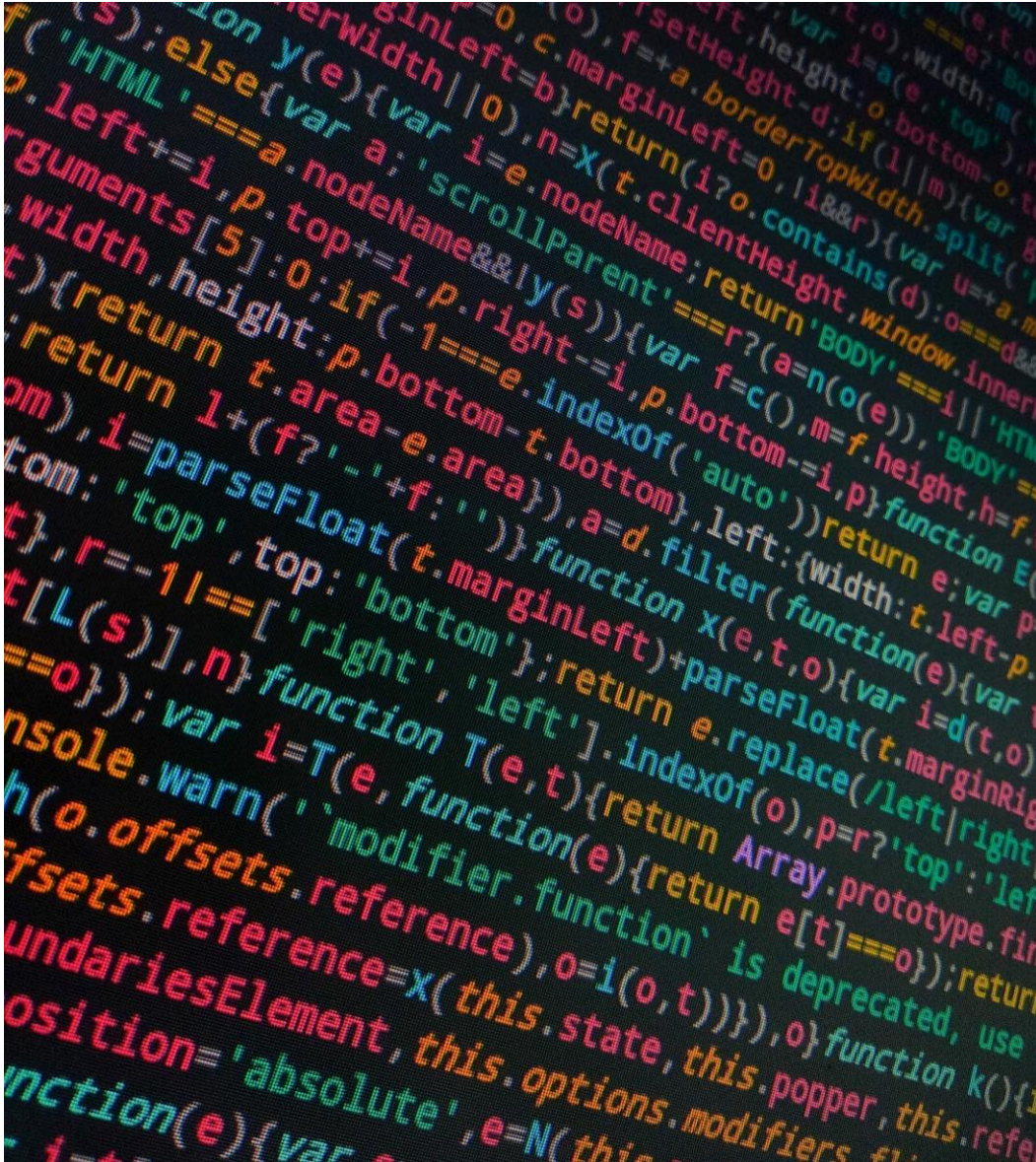


Photo by [Walkator](#) on [Unsplash](#)

What your raw code may  
look like for programmers  
who do not know the  
language.

---



Raw source code

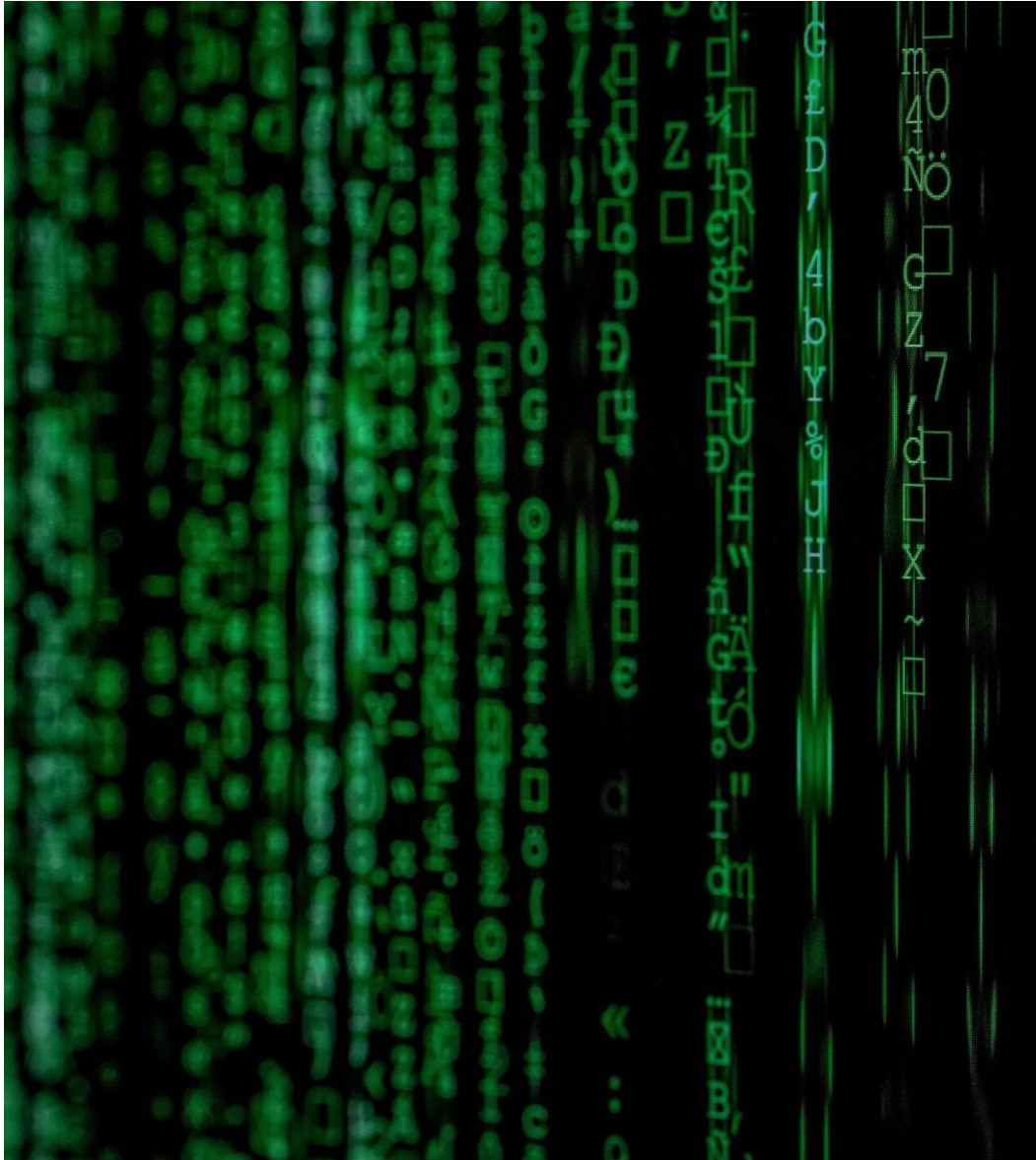
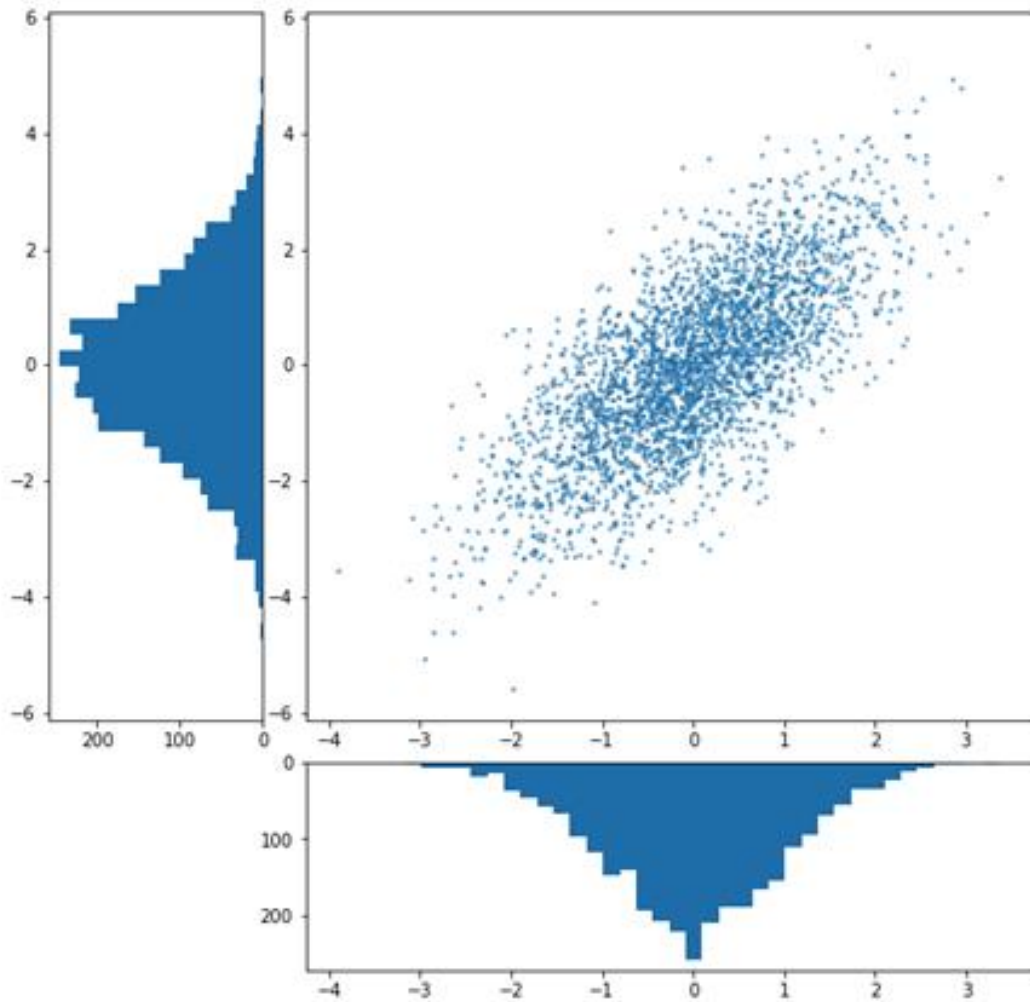


Photo by [Markus Spiske](#) on [Unsplash](#)

What your raw code may  
look like for non-  
programmers.



Raw source code



The executable can show  
what the program does.



A compiled executable.





Photo by [Laura Ockel](#) on [Unsplash](#)

But the executable does not show  
how it works (no transparency).

Even with the source code it can be  
difficult to grasp exactly how it  
works.



A compiled executable.



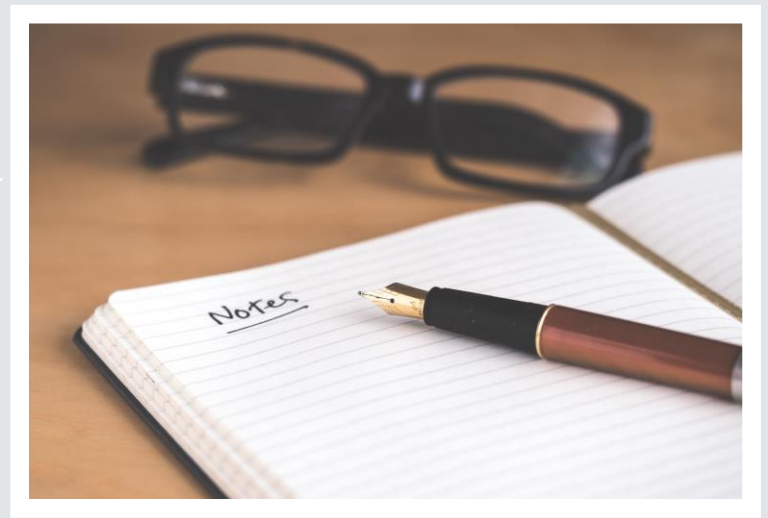
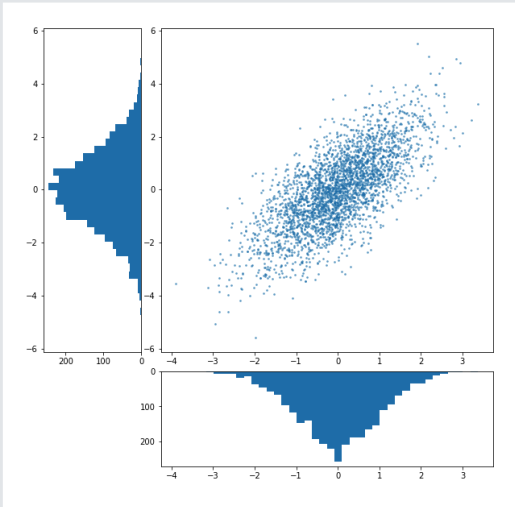


Photo by [Aaron Burden](#) on [Unsplash](#)

jupyter

```
66 var ax = settings.acxX;
67 var ay = settings.acyY;
68 var th = t.height();
69 var wh = w.height();
70 var tw = t.width();
71 var ww = w.width();
72
73 if (y + th + ay >= b 66
74     y <= b + wh + ay 66
75     x + tw + ax >= a 66
76     x <= a + ww + ax) {
77
78     //trigger the custom event
79     if (!t.appeared) t.trigger('appear', settings.detail);
80
81     } else {
82
83     //it scrolled out of view
84     t.appeared = false;
85
86     }
87
88 };
89
90 //create a modified fn with some additional logic
91 var modifiedFn = function() {
92
93     //mark the element as visible
94     t.appeared = true;
95
96     //this supposed to happen only once!
```

Photo by [Markus Spiske](#) on [Unsplash](#)



Example

# What is a Jupyter notebook?

Open-source web application combining in a single document:



EXPLANATORY  
(NARRATIVE) TEXT



LIVE PROGRAMMING  
CODE



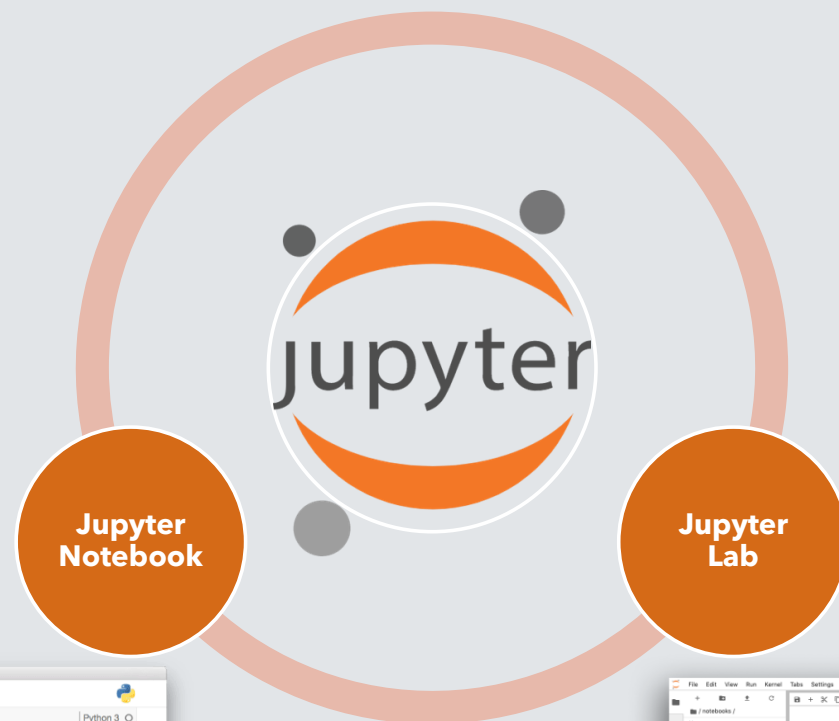
VISUALIZATIONS



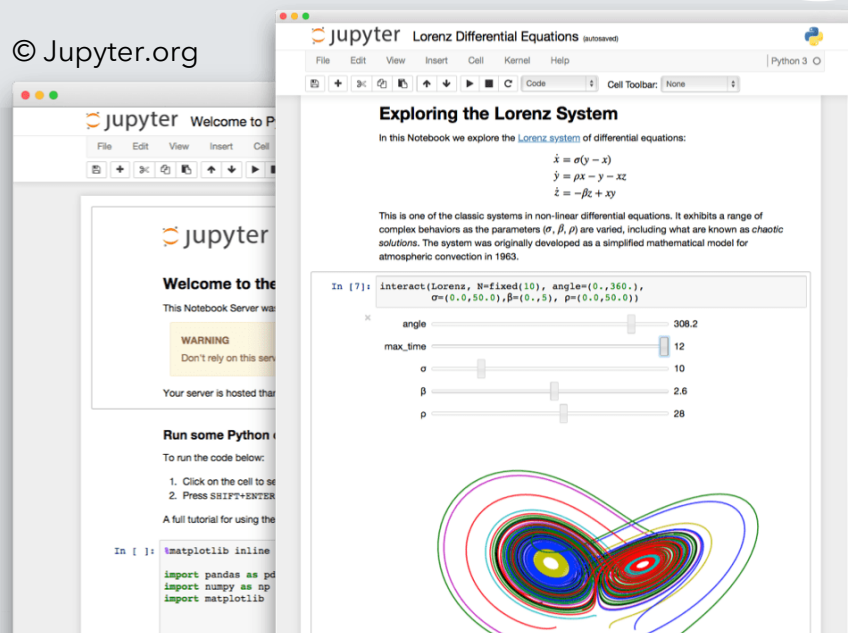
MULTIMEDIA  
RESOURCES



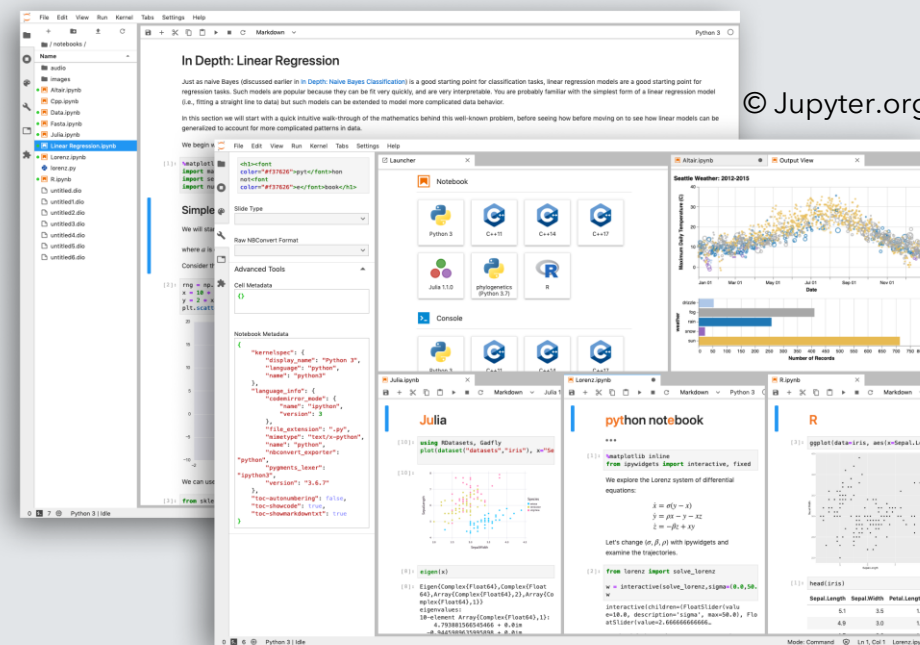
**Example**

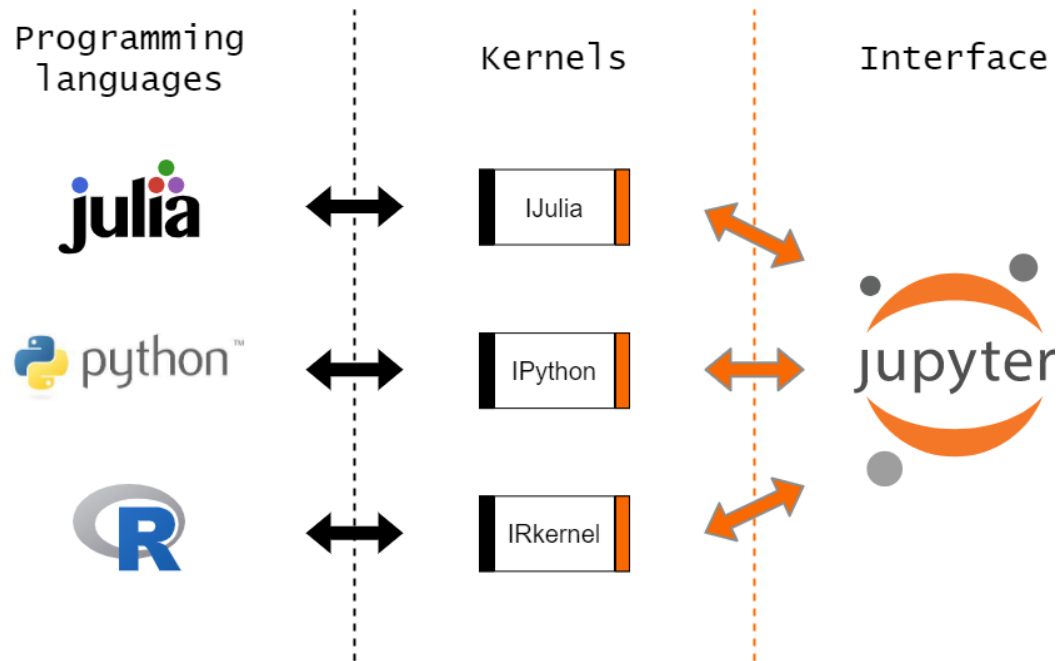


© Jupyter.org



© Jupyter.org





## Kernels

“Kernels are programming language specific processes that run independently and interact with the Jupyter Applications and their user interfaces.”

([Jupyter.readthedocs.io](https://jupyter.readthedocs.io))



Try it!

---

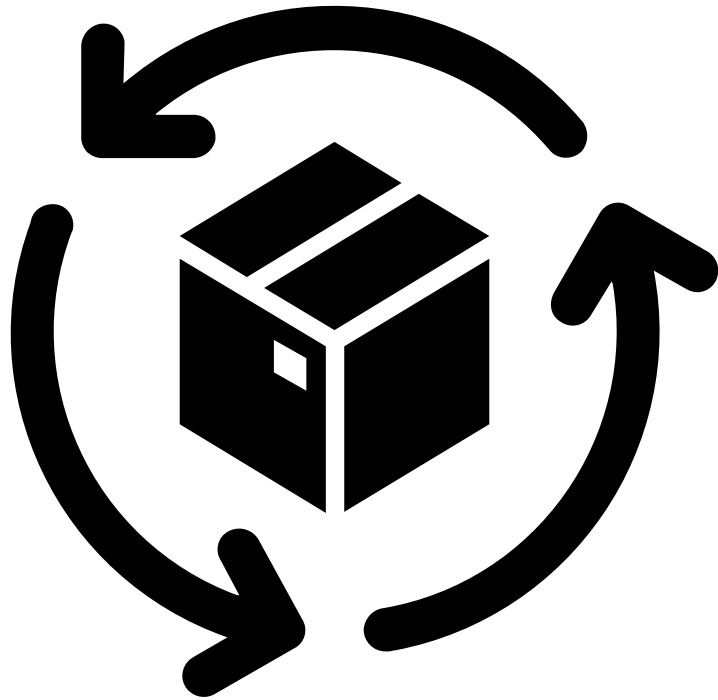
<https://jupyter.org/try>

---

<https://colab.research.google.com>

# Demo





# Installation

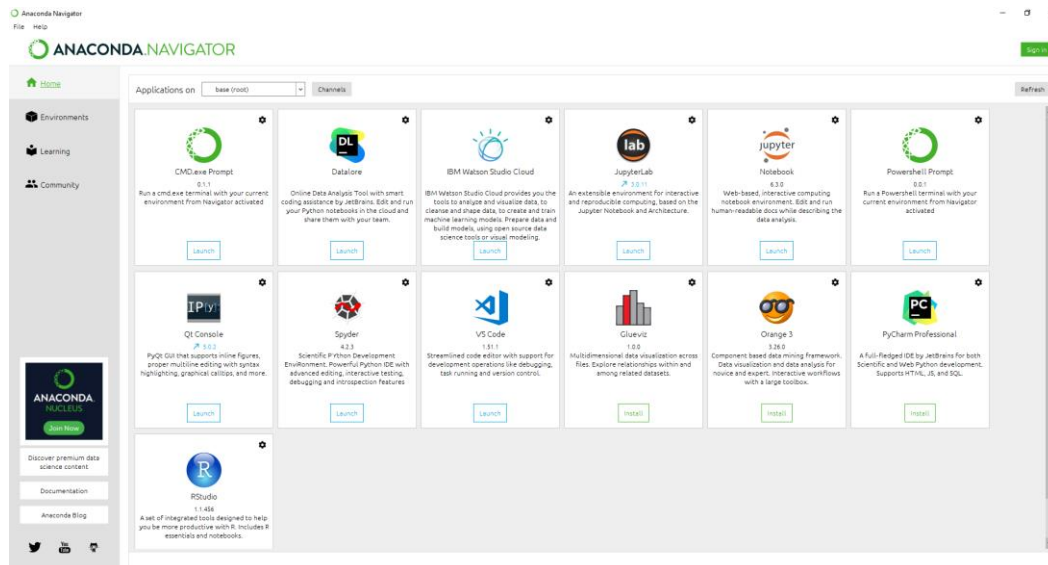
---

1. Install Jupyter
  - <https://jupyter.org/install>
2. Install Kernel
  - [Kernel list](#)
  - Example
  - ⌘ <https://irkernel.github.io/installation>



# Installation

- [Anaconda](#)







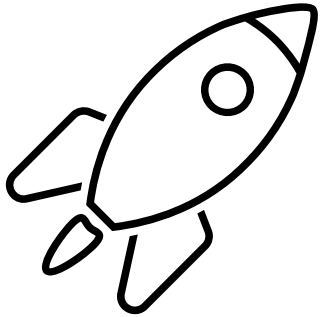
## Conclusion

### Main benefits:

- Choice of language
- Live interaction with code
- Notebooks sharing
- Interactive output
- Great for prototyping

### Main Limitations:

- Sequence of execution matters
- IDE-like but no debugger
- Careful from where it is launched
- Learning curve for maintaining extensions



Thank you and go  
explore Jupyter!

---