DataScientest • com

First steps on the platform

Welcome to the DataScientest platform!

> Our *Data Science* training platform is **secure** and **accessible 24 hours a day, 7 days a week** from any browser with internet access. You therefore have the possibility of working both from your workplace and from your personal computer.
>
> Our training courses adopt the pedagogy of *learning by doing*: for each concept addressed on our platform, you will be asked to put it into practice by coding on exercises. This pedagogy will allow you to learn at your own pace and independently with the aim of obtaining certifications in Data Science skills corresponding to your educational path.
>
> In practice, the training is divided into *modules*, themselves made up of *notebooks*, which are powered by the ***Jupyter Notebook*** technology.
>
> *Jupyter* notebooks are the ideal format for learning Data Science:
>
> - They are made of a succession of *Markdown* cells (simple text formatting language) and *Python* code cells. This format will allow you to learn theoretical or technical notions and put them into practice on the same document.
>
> Jupyter notebooks are very popular among **Data Scientists** and have many features. Below, we introduce you to the most important features to use the training platform in the best conditions.

Introduction to notebook features

> When you write code in a cell, the cell must be run to transmit the code to the Python interpreter so that the code inside it is executed.
>
> To run a cell: use the **Run Cell** button at the top left of your screen or use the keyboard shortcut: `Ctrl + Enter`

- Run the following cell.

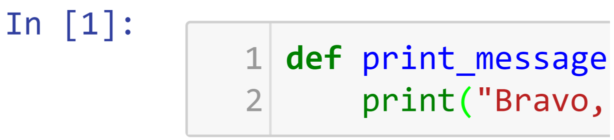
NB: The cell [illegible]

In [4]:

```
def print_message(a=1) :
    if a == 1:
        print("Congratulations, you just ran your 1st cell !")
    elif a == 2:
        print("Congratulations, you just ran your 2nd cell !")
    elif a == 3:
        print("Congratulations, you just ran your 3rd cell !")
    else:
        print("Congratulations, you just ran your %ith cell !" %a)
```

Running the cell above did not produce any output results. However, the instruction did have an effect: the **`print_message`** function was created.

The number in square brackets on the left next to **`In`** is a counter. It is incremented each time a cell is executed:



In [1]:

```
def print_message
    print("Bravo,
```

The imported packages or the objects created in a cell are then available in all the other code cells of the notebook. For example, the function defined in the cell above can be called in any code cell.

ℹ Throughout your training, you will have to answer a number of questions in code form. For each question, a fillable cell is made available to you to answer the questions, and a **"Show Solution"** button below allows you to display a proposed solution.

- Call the **`print_message`** function with the argument `a = 2`. This is done by running the instruction:

```
print_message(a = 2)
```

Do not forget to **run the cell** once your code is written (Ctrl + Enter).

In [6]:

```
# Insert your code here
print_message(a=2)
```

```
Congratulations, you just ran your 2nd cell !
```

Hide solution

In [ ]:

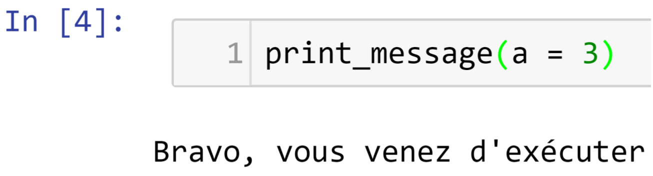
```
print_message(a = 2)
```

> In a Notebook, you can modify a cell and re-run it as much as you want.

- In the cell above, call the `print_message` function again but with the argument `a = 3`.

When you launch a cell, the execution time can vary from a millisecond to several minutes (for a code involving many computations).

Once the code present in the cell has finished being interpreted and executed, a new number appears in brackets next to **In**:



```
In [4]:  print_message(a = 3)

         Bravo, vous venez d'exécuter
```

This corresponds to the number of cells executed since the **Kernel** was started.

What is a Kernel?
An **kernel** is an independent program capable of interpreting the code in the cell and running it. When you open an exercise for the first time, a kernel is automatically started.

Sometimes several minutes pass and the cell is still being executed:

In [7]:

```
i = 10
while i <= 1e9 :
    if i % 100000000 == 0:
        print(i // 1000000, "million iterations have been executed.")
    i += 1
```

```
100 million iterations have been executed.
200 million iterations have been executed.
300 million iterations have been executed.
400 million iterations have been executed.
500 million iterations have been executed.
600 million iterations have been executed.
700 million iterations have been executed.
800 million iterations have been executed.
900 million iterations have been executed.
1000 million iterations have been executed.
```

Many keyboard shortcuts exist on Jupyter and can facilitate your user experience on the platform.

For example, some questions will contain multiple items, which will display multiple outputs. For a visually pleasing rendering, you can add **new** code cells.

To add a new cell below another, you can use the shortcut: **Esc + B**.

In [ ]:

```
## Add a few code cells below this one
```

In [ ]:

```
```

In [ ]:

```
```

Hello this is Markdown

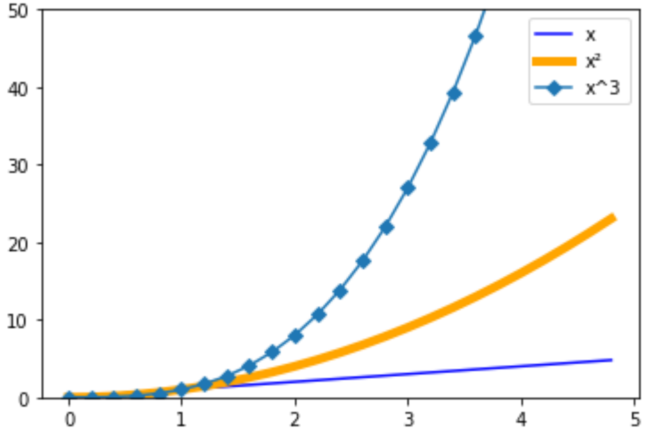
In [ ]:

```
# Hello this is Text
```

In [12]:

```
%matplotlib inline

import numpy as np
import matplotlib.pyplot as plt

t = np.arange(0, 5, 0.2)
#print(t)

plt.plot(t, t, 'b', label = "x")
plt.plot(t, t**2, 'orange', linewidth=5, label= "x²")
plt.plot(t, t**3, marker='D', label = "x^3")
plt.ylim([0,50])
plt.xlabel
plt.legend()

plt.show()
```

```
[0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8 2.  2.2 2.4 2.6 2.8 3.  3.2 3.4
 3.6 3.8 4.  4.2 4.4 4.6 4.8]
```



In [ ]:

```
%matplotlib inline

import numpy as np import matplotlib.pyplot as plt

t = np.arange(0, 5, 0.2)

plt.plot(t, t, 'b', label = "x") plt.plot(t, t2, 'orange', linewidth=5, label= "x²") plt.plot(t, t3, marker='D', label = "x^

plt.show()
```

> When you add a cell, by default you get a cell of **code**. Code cells are identifiable thanks to the `In[ ]` on the left.
>
> - To transform a **code** cell into a **text** (*Markdown*) cell you can use the shortcut: **Esc + M**.
> - To transform a **text** cell into a **code** cell, you can use the shortcut: **Esc + Y**.

%matplotlib inline

import numpy as np import matplotlib.pyplot as plt

t = np.arange(0, 5, 0.2)

plt.plot(t, t, 'b', label = "x") plt.plot(t, t**2, 'orange', linewidth=5, label= "x²") plt.plot(t, t**3, marker='D', label = "x^3") plt.ylim([0,50]) plt.xlabel plt.legend()

plt.show()

Hide solution

In [ ]:

```
"""
The cell above is of Markdown (text) type.
Press Esc + Y to transform it into a code cell.
"""

%matplotlib inline

import numpy as np
import matplotlib.pyplot as plt

t = np.arange(0, 5, 0.2)

plt.plot(t, t, 'b', label = "x")
plt.plot(t, t**2, 'orange', linewidth=5, label= "x²")
plt.plot(t, t**3, marker='D', label = "x^3")
plt.ylim([0,50])
plt.legend()

plt.show()
```

> To obtain the list of available shortcuts, press **Esc + H**.
>
> It is also possible to open the command palette using the shortcut **Esc + P**.

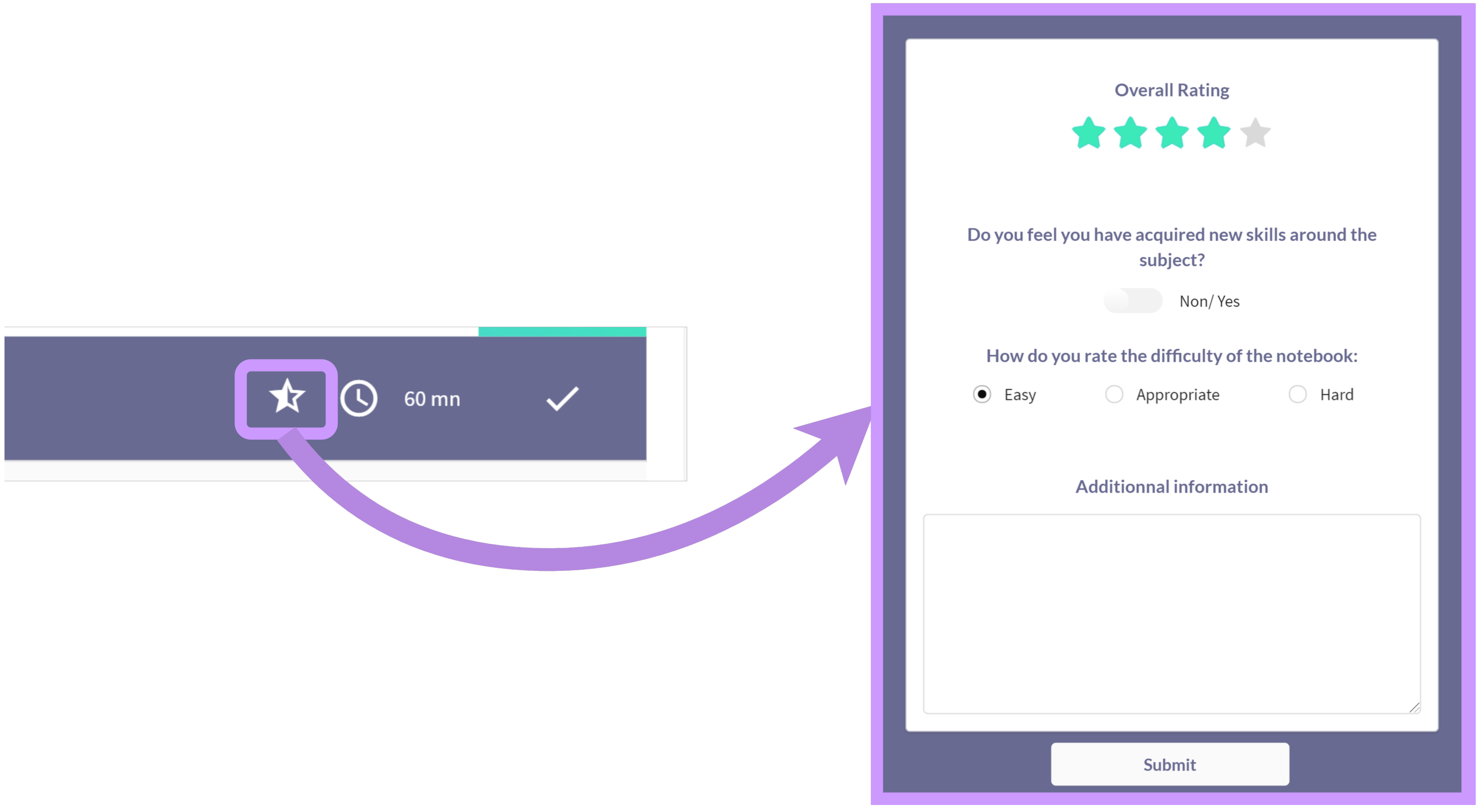
Support

> The content offered on the platform is designed for you to be as independent as possible in your learning. Nevertheless, it is possible that questions remain, and in this case our team of Data Scientists is waiting for you on **Slack**, the collaborative messaging platform.
>
> > Slack ?
> > Don't forget to join the **Slack** workspace dedicated to your training group in order to have access to our support.
> > Slack is essential for communicating with the support team and other users in your training group. This is the **only** channel on which the **informations** concerning your training will be published regularly.
>
> **If you have not received any invitations to your Slack workspace**, check that the invitation e-mail has not been lost in your spam folder. If this is not the case, send an e-mail to help@datascientest.com (mailto:help@datascientest.com).
>
> Email support is available through our support account **Daniel - DataScientest**. Daniel is available every **on working days** from **9 a.m.** to **7 p.m.** to answer your questions regarding the **content** of the platform, or to resolve concerns related to a **technical issue**.
>
> If you experience a problem outside of support hours, you can also ask your urgent questions 24/7 at help@datascientest.com (mailto:help@datascientest.com). Remember to add as much detail to your message as possible, such as:
>
> - A screenshot of your problem.
> - The context.
> - How to reproduce your problem etc ...

Feedback

No one is perfect, not even the team of teachers! **We are always listening to your feedback.** It is possible:

- To assign a rating and / or a comment to **each notebook** validated directly on our platform by clicking on the **star** which appears next to the name of the notebook:



Overall Rating

★★★★☆

Do you feel you have acquired new skills around the subject?

Non/ Yes

How do you rate the difficulty of the notebook:

● Easy ○ Appropriate ○ Hard

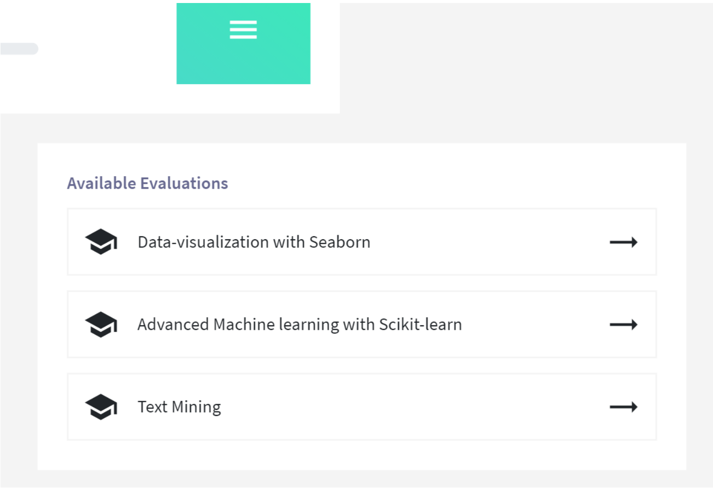
Additionnal information

- More complete feedback about a **module** can be given with the following Google form: [https://goo.gl/forms/qbuEVcJtJjQGMLhi2 (https://goo.gl/forms/qbuEVcJtJjQGMLhi2)](https://goo.gl/forms/qbuEVcJtJjQGMLhi2)

Please note: your corporate network may prevent you from opening Google Forms.

Evaluations

When all the notebooks of a module are validated, the associated evaluation is unlocked and appears **at the bottom of your Dashboard**.



Available Evaluations

Data-visualization with Seaborn ⟶

Advanced Machine learning with Scikit-learn ⟶

Text Mining ⟶

The evaluation aims to validate the knowledge you acquired during the module.

An unlocked evaluation remains accessible in your dashboard in the **Available evaluations** section for **7 days**. If you exceed this deadline, you may request a new invitation to **Daniel**.

The results of your evaluation will be communicated to you within a maximum of **48 working hours**.

- If successful, you will receive an email containing your certification and its license number.
- In case of failure, you will be able to retake the evaluation **up to 2 additional times**. We strongly recommend that you work on the concepts that seemed difficult to you between each trial.

As specified in the educational path, **some modules do not have evaluations**. They are in this case a preparation for a future evaluation.

If you have not received your educational path, do not hesitate to ask **Daniel** to find out about the modules that are associated with a certification.

ℹ The validation rate for an exam on the first try may be less than 50% for certain modules. Anticipate several trials in case you have to finish your exam before a given deadline

✖ Unvalidate