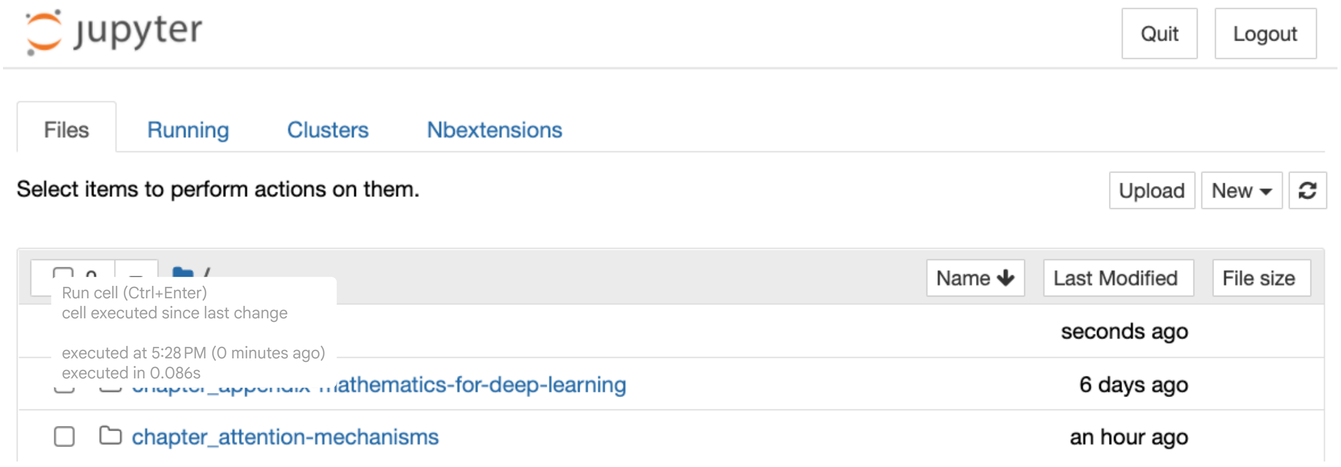
Using Jupyter Notebooks

:label: `sec_jupyter`

This section describes how to edit and run the code in each section of this book using the Jupyter Notebook. Make sure you have installed Jupyter and downloaded the code as described in :ref: `chap_installation` . If you want to know more about Jupyter see the excellent tutorial in their documentation.

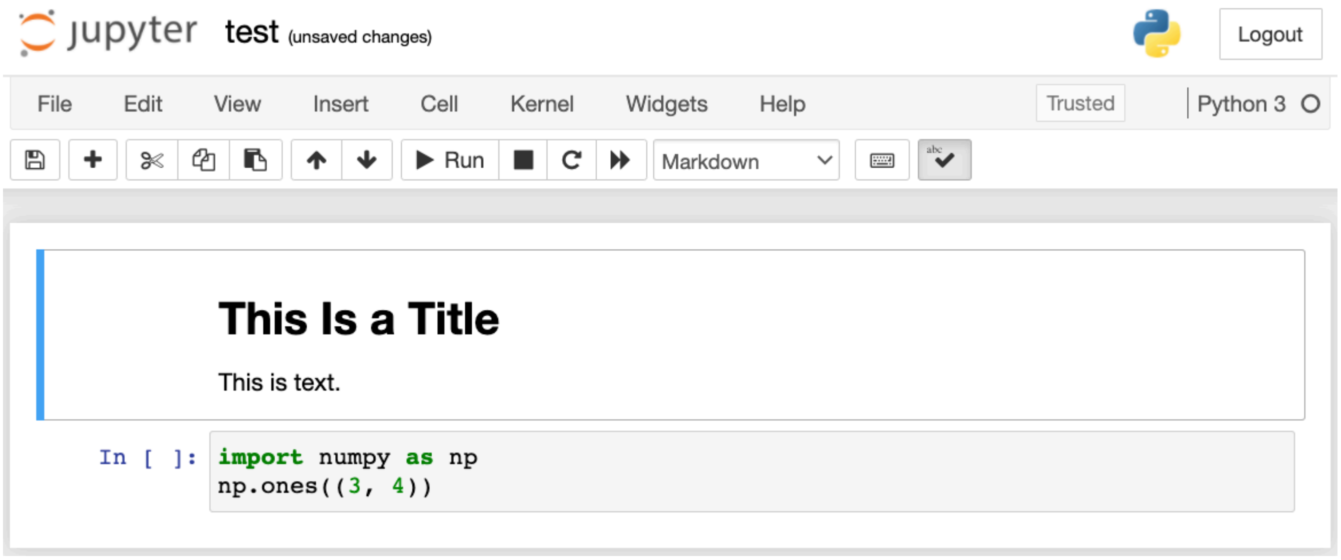
Editing and Running the Code Locally

Suppose that the local path of the book's code is `xx/yy/d2l-en/` . Use the shell to change the directory to this path (`cd xx/yy/d2l-en`) and run the command `jupyter notebook` . If your browser does not do this automatically, open http://localhost:8888 and you will see the interface of Jupyter and all the folders containing the code of the book, as shown in :numref: `fig_jupyter00` .


jupyter
Quit
Logout
Files
Running
Clusters
Nbextensions
Select items to perform actions on them.
Upload
New
Name
Last Modified
File size
Run cell (Ctrl+Enter)
cell executed since last change
executed at 5:28 PM (0 minutes ago)
executed in 0.086s
seconds ago
mathematics-for-deep-learning
6 days ago
chapter_attention-mechanisms
an hour ago

:width: `600px` :label: `fig_jupyter00`

You can access the notebook files by clicking on the folder displayed on the webpage. They usually have the suffix ".ipynb". For the sake of brevity, we create a temporary "test.ipynb" file. The content displayed after you click it is shown in :numref: `fig_jupyter01` . This notebook includes a markdown cell and a code cell. The content in the markdown cell includes "This Is a Title" and "This is text.". The code cell contains two lines of Python code.



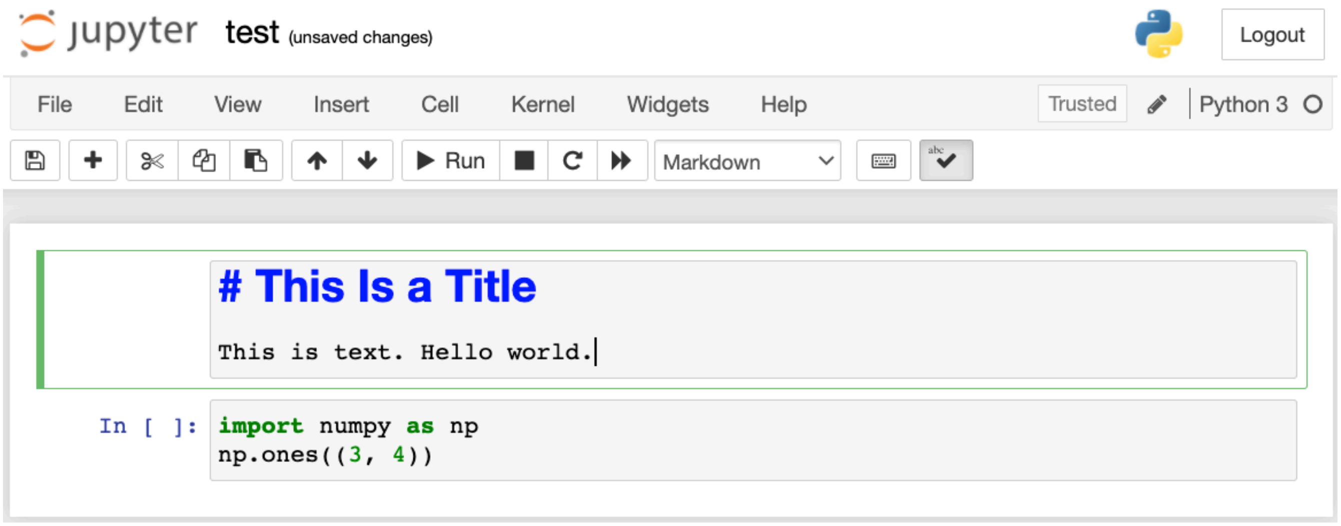
This Is a Title

This is text.

```
In [ ]: import numpy as np
        np.ones((3, 4))
```

:width: `600px` :label: `fig_jupyter01`

Double click on the markdown cell to enter edit mode. Add a new text string "Hello world." at the end of the cell, as shown in :numref: `fig_jupyter02` .



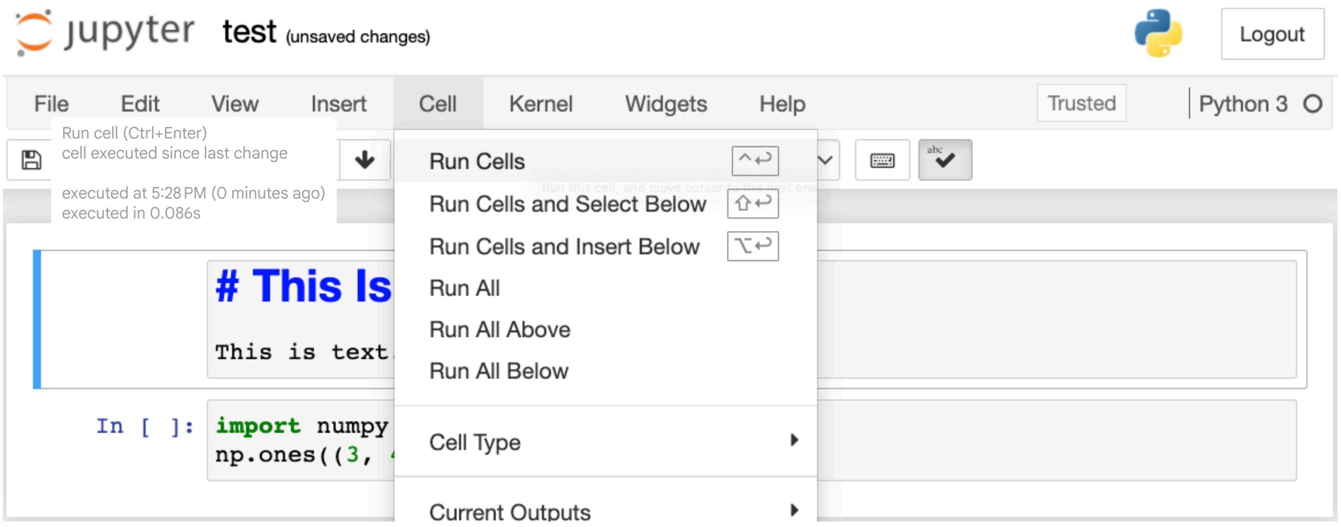
```
# This Is a Title

This is text. Hello world.
```

```
In [ ]: import numpy as np
        np.ones((3, 4))
```

:width: `600px` :label: `fig_jupyter02`

As demonstrated in :numref: `fig_jupyter03` , click "Cell" $\rightarrow$ "Run Cells" in the menu bar to run the edited cell.



Run cell (Ctrl+Enter)
cell executed since last change

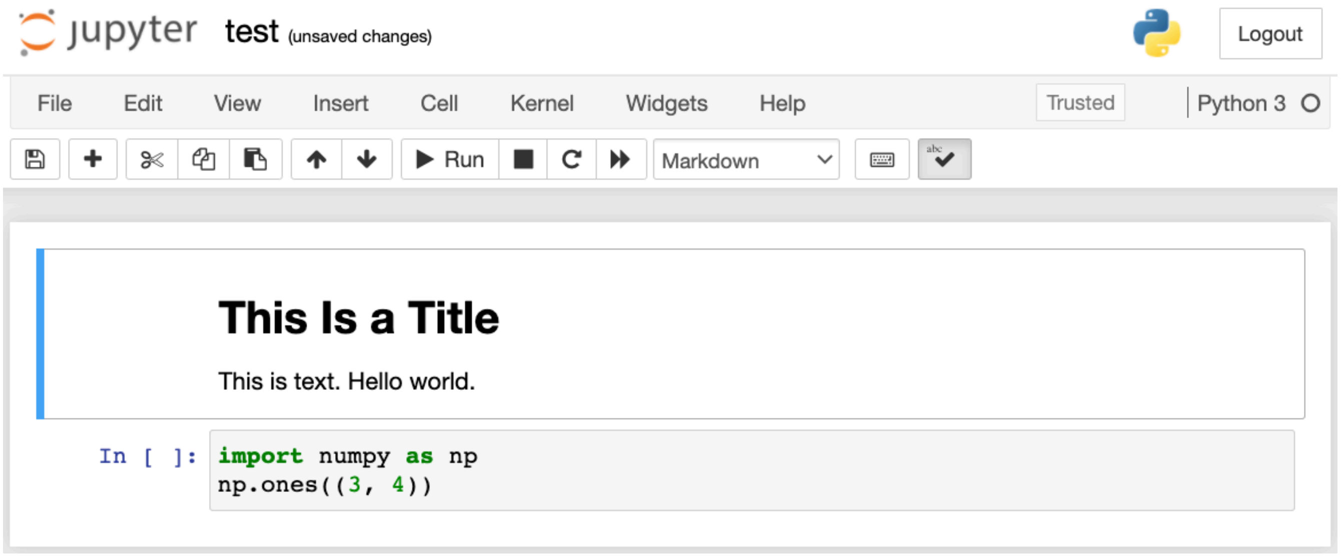
executed at 5:28 PM (0 minutes ago)
executed in 0.086s

```
# This Is
This is text

In [ ]: import numpy
        np.ones((3,
```

:width: `600px` :label: `fig_jupyter03`

After running, the markdown cell is shown in :numref: `fig_jupyter04` .



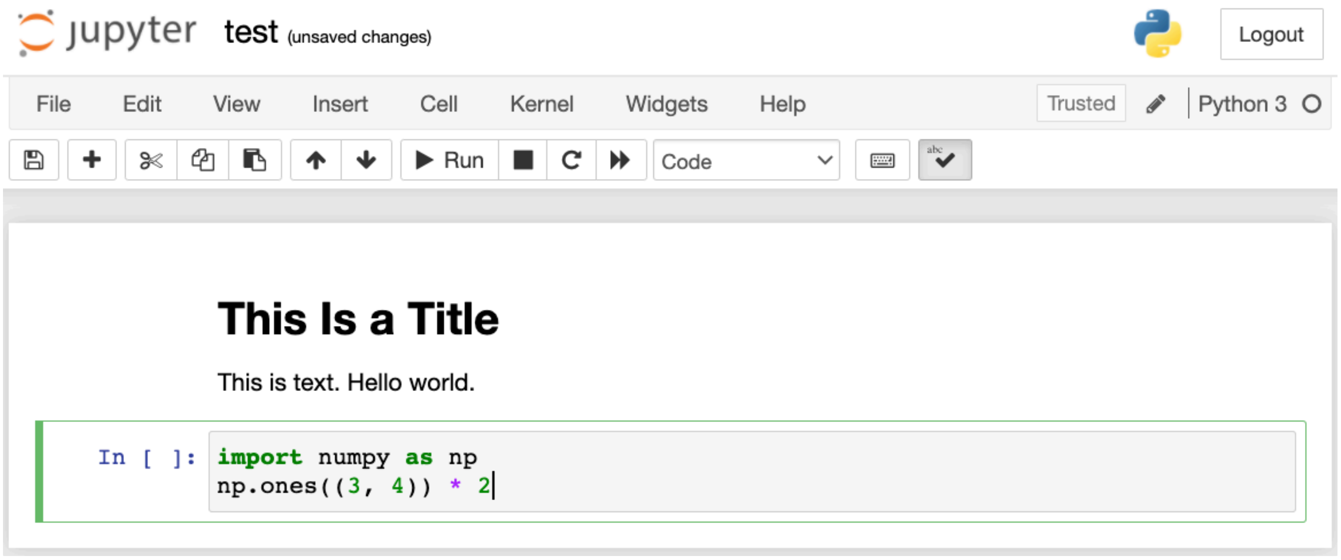
This Is a Title

This is text. Hello world.

```
In [ ]: import numpy as np
        np.ones((3, 4))
```

:width: `600px` :label: `fig_jupyter04`

Next, click on the code cell. Multiply the elements by 2 after the last line of code, as shown in :numref: `fig_jupyter05` .


jupyter test (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

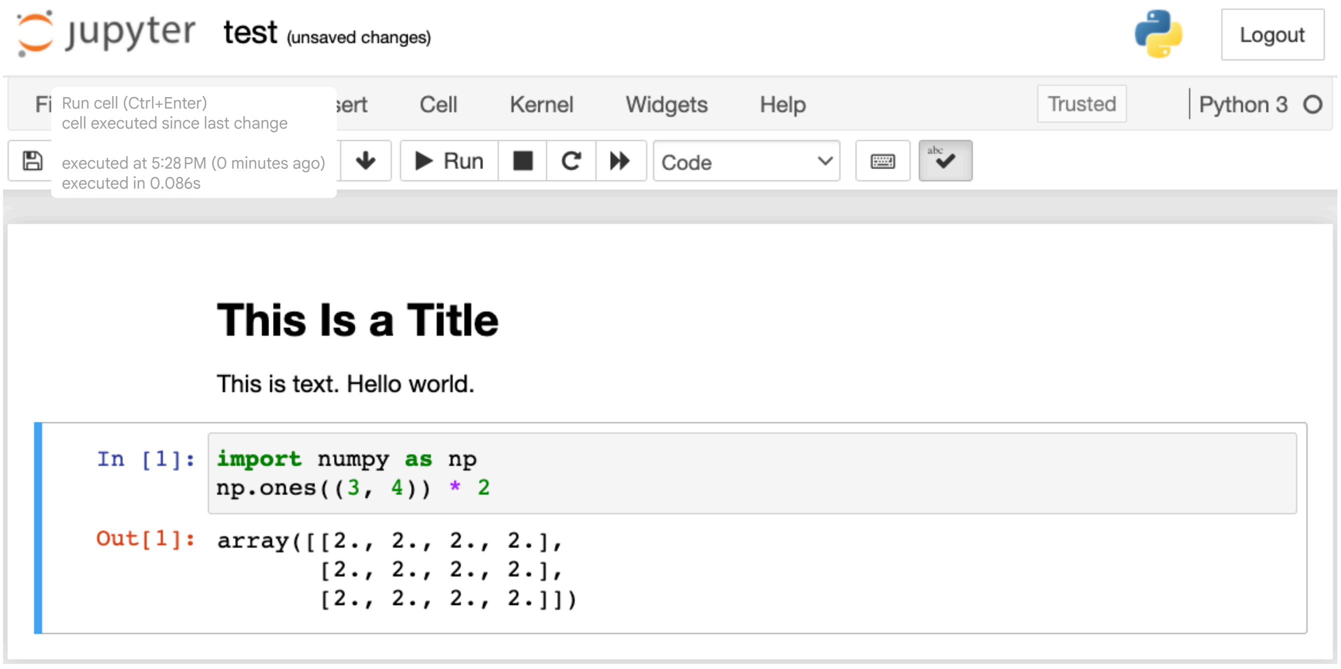
This Is a Title

This is text. Hello world.

```
In [ ]: import numpy as np
        np.ones((3, 4)) * 2
```

:width: `600px` :label: `fig_jupyter05`

You can also run the cell with a shortcut ("Ctrl + Enter" by default) and obtain the output result from :numref: `fig_jupyter06` .


jupyter test (unsaved changes)
Logout
Fi
sert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Run cell (Ctrl+Enter)
cell executed since last change
executed at 5:28 PM (0 minutes ago)
executed in 0.086s

This Is a Title

This is text. Hello world.

```
In [1]: import numpy as np
        np.ones((3, 4)) * 2

Out[1]: array([[2., 2., 2., 2.],
               [2., 2., 2., 2.],
               [2., 2., 2., 2.]])
```

:width: `600px` :label: `fig_jupyter06`

When a notebook contains more cells, we can click "Kernel" $\rightarrow$ "Restart & Run All" in the menu bar to run all the cells in the entire notebook. By clicking "Help" $\rightarrow$ "Edit Keyboard Shortcuts" in the menu bar, you can edit the shortcuts according to your preferences.

Advanced Options

Beyond local editing two things are quite important: editing the notebooks in the markdown format and running Jupyter remotely. The latter matters when we want to run the code on a faster server. The former matters since Jupyter's native ipynb format stores a lot of auxiliary data that is irrelevant to the content, mostly related to how and where the code is run. This is confusing for Git, making reviewing contributions very difficult. Fortunately there is an alternative---native editing in the markdown format.

Markdown Files in Jupyter

If you wish to contribute to the content of this book, you need to modify the source file (md file, not ipynb file) on GitHub. Using the notedown plugin we can modify notebooks in the md format directly in Jupyter.

First, install the notedown plugin, run the Jupyter Notebook, and load the plugin:

```
pip install d2l-notedown  # You may need to uninstall the original notedown.
jupyter notebook --NotebookApp.contents_manager_class='notedown.NotedownContentsManager'
```

You may also turn on the notedown plugin by default whenever you run the Jupyter Notebook. First, generate a Jupyter Notebook configuration file (if it has already been generated, you can skip this step).

```
jupyter notebook --generate-config
```

Then, add the following line to the end of the Jupyter Notebook configuration file (for Linux or macOS, usually in the path `~/.jupyter/jupyter_notebook_config.py`):

```
c.NotebookApp.contents_manager_class = 'notedown.NotedownContentsManager'
```

After that, you only need to run the `jupyter notebook` command to turn on the notedown plugin by default.

Running Jupyter Notebooks on a Remote Server

Sometimes, you may want to run Jupyter notebooks on a remote server and access it through a browser on your local computer. If Linux or macOS is installed on your local machine (Windows can also support this function through third-party software such as PuTTY), you can use port forwarding:

```
ssh myserver -L 8888:localhost:8888
```

The above string `myserver` is the address of the remote server. Then we can use http://localhost:8888 to access the remote server `myserver` that runs Jupyter notebooks. We will detail on how to run Jupyter notebooks on AWS instances later in this appendix.

Timing

We can use the `ExecuteTime` plugin to time the execution of each code cell in Jupyter notebooks. Use the following commands to install the plugi

Run cell (Ctrl+Enter)
cell executed since last change

executed at 5:28 PM (0 minutes ago)
executed in 0.086s

```
pip                                      tensions
jupy                                     :all --user
jupyter nbextension enable execute_time/ExecuteTime
```

Summary

- Using the Jupyter Notebook tool, we can edit, run, and contribute to each section of the book.
- We can run Jupyter notebooks on remote servers using port forwarding.

Exercises

1. Edit and run the code in this book with the Jupyter Notebook on your local machine.
2. Edit and run the code in this book with the Jupyter Notebook *remotely* via port forwarding.
3. Compare the running time of the operations $\mathbf{A}^\top \mathbf{B}$ and $\mathbf{A}\mathbf{B}$ for two square matrices in $\mathbb{R}^{1024 \times 1024}$. Which one is faster?

Discussions

```
import numpy as np

 .__version__
 import langchain
 langchain.__version__
```

✓ ✕

```
  File "<ipython-input-2-b65c07388179>", line 1
    .__version__
    ^
SyntaxError: invalid syntax
```

```
np.__version__
```

```
'2.0.2'
```

creating array

```
my_list=[0,1,2,3,4,5]
my_list
```

```
[0, 1, 2, 3, 4, 5]
```

```
type(my_list)
```

```
list
```

```
np.arrange(10)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-7-d5d6780a30fc> in <cell line: 0>()
----> 1 np.arrange(10)

/usr/local/lib/python3.11/dist-packages/numpy/__init__.py in __getattr__(attr)
    408             return char.chararray
    409
--> 410         raise AttributeError("module {!r} has no attribute "
    411                              "{!r}".format(__name__, attr))
    412

AttributeError: module 'numpy' has no attribute 'arrange'
```

Next steps: Explain error

```
np.arange(10)
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
np.arange(5.0)
```

Run cell (Ctrl+Enter)
cell executed since last change

executed at 5:28 PM (0 minutes ago)
executed in 0.086s

```
np.ar
```

```
array([0., 1., 2., 3., 4.])
```

```
np.arange(10,20)
```

```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
np.arange(16,10)
```

```
array([], dtype=int64)
```

```
np.arange(20,10)
```

```
array([], dtype=int64)
```

```
np.arange(-10,-20)
```

```
array([], dtype=int64)
```

```
np.arange(10,30,5)
```

```
array([10, 15, 20, 25])
```

np zeros

```
np.zeros(10)
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
np.zeros((2,2),dtype=int)
```

```
array([[0, 0],
       [0, 0]])
```

```
np.zeros((2,10))
```

```
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

```
np.zeros((10,30))
```

```
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

```
np.zeros((2,10),dtype=int)
```

```
array([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]])
```

```
np.ones(3)
```

```
array([1., 1., 1.])
```

```
np.ones((3,3))
```

Run cell (Ctrl+Enter)
cell executed since last change

executed at 5:28 PM (0 minutes ago)
executed in 0.086s

```
np.ones((3,3),dtype=int)
```

```
array([[1, 1, 1],
       [1, 1, 1],
       [1, 1, 1]])
```

```
np.twos((2,3))
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-26-ea940da5f119> in <cell line: 0>()
----> 1 np.twos((2,3))

/usr/local/lib/python3.11/dist-packages/numpy/__init__.py in __getattr__(attr)
    408             return char.chararray
    409 
--> 410         raise AttributeError("module {!r} has no attribute "
    411                              "{!r}".format(__name__, attr))
    412 

AttributeError: module 'numpy' has no attribute 'twos'
```

Next steps: Explain error

```
rand(2,3)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-27-31cbb7407a5d> in <cell line: 0>()
----> 1 rand(2,3)

NameError: name 'rand' is not defined
```

Next steps: Explain error

```
random.rand(2,3)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-28-d1c2c57d213f> in <cell line: 0>()
----> 1 random.rand(2,3)

NameError: name 'random' is not defined
```

Next steps: Explain error

```
np.random.rand(5)
```

```
array([0.90933426, 0.40407496, 0.39772155, 0.146588  , 0.70033356])
```

Start coding or generate with AI.

Start coding or generate with AI.

Run cell (Ctrl+Enter)
cell executed since last change

executed at 5:28 PM (0 minutes ago)
executed in 0.086s