

Week 3: import/export | plotting

**NRSC 7657 Workshop in Advanced Programming for
Neuroscientists**

course business

- Application topic
 - ?

Built - ins

The screenshot shows a web browser displaying the Python 3.9.5 documentation for built-in functions. The title "Built-in Functions" is at the top, followed by a brief description: "The Python interpreter has a number of functions and types built into it that are always available. They are listed here in alphabetical order." To the left is a sidebar with links to "Previous topic" (Introduction), "Next topic" (Built-in Constants), and "This Page" (Report a Bug, Show Source). The main content area contains a table titled "Built-in Functions" with two columns of function names.

| Built-in Functions | |
|----------------------------|-----------------------------|
| <code>abs()</code> | <code>delattr()</code> |
| <code>all()</code> | <code>dict()</code> |
| <code>any()</code> | <code>dir()</code> |
| <code>ascii()</code> | <code>divmod()</code> |
| <code>bin()</code> | <code>enumerate()</code> |
| <code>bool()</code> | <code>eval()</code> |
| <code>breakpoint()</code> | <code>exec()</code> |
| <code>bytearray()</code> | <code>filter()</code> |
| <code>bytes()</code> | <code>float()</code> |
| <code>callable()</code> | <code>format()</code> |
| <code>chr()</code> | <code>frozenset()</code> |
| <code>classmethod()</code> | <code>getattr()</code> |
| <code>compile()</code> | <code>globals()</code> |
| <code>complex()</code> | <code>hasattr()</code> |
| | |
| <code>hash()</code> | <code>help()</code> |
| <code>id()</code> | <code>hex()</code> |
| <code>input()</code> | <code>next()</code> |
| <code>int()</code> | <code>object()</code> |
| <code>iter()</code> | <code>oct()</code> |
| <code>len()</code> | <code>property()</code> |
| <code>list()</code> | <code>range()</code> |
| <code>locals()</code> | <code>repr()</code> |
| <code>map()</code> | <code>reversed()</code> |
| <code>max()</code> | <code>round()</code> |
| | |
| <code>memoryview()</code> | <code>set()</code> |
| <code>min()</code> | <code>setattr()</code> |
| <code>next()</code> | <code>slice()</code> |
| <code>object()</code> | <code>sorted()</code> |
| <code>oct()</code> | <code>staticmethod()</code> |
| <code>open()</code> | <code>str()</code> |
| <code>ord()</code> | <code>sum()</code> |
| <code>pow()</code> | <code>super()</code> |
| <code>print()</code> | <code>tuple()</code> |
| <code>property()</code> | <code>type()</code> |
| <code>range()</code> | <code>vars()</code> |
| <code>repr()</code> | <code>zip()</code> |
| <code>reversed()</code> | |
| <code>round()</code> | |

Imports

- You want to use something besides built-ins

```
[ ]: import pynwb
```

```
[ ]:
```

- You've got functions you wrote in a notebook, like last week. Now you have a new notebook. How do you get your (cool, useful) functions to your new notebook? Or a script? Or your labmate's notebook?

```
[ ]: from dlab import sorting_quality as sq
```

```
[ ]:
```

Imports

Modules

- **Find** a module
- **Give** it a name in the namespace

The screenshot shows a web browser displaying the Python 3.9.5 documentation for the 'import' module. The title '5. The import system' is at the top. To the left is a 'Table of Contents' sidebar with a hierarchical list of topics under '5. The import system'. The main content area contains several paragraphs of text explaining the import process, including the search and loading phases, and how it differs from direct calls to `__import__()`.

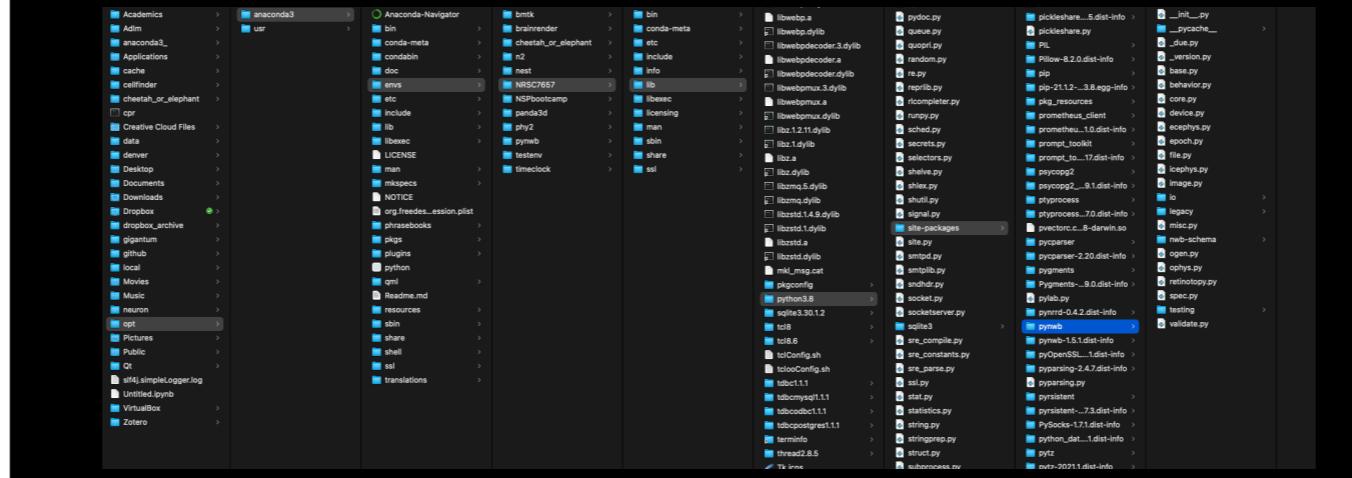
Table of Contents

- 5. The import system
 - 5.1. `importlib`
 - 5.2. Packages
 - 5.2.1. Regular packages
 - 5.2.2. Namespace packages
 - 5.3. Searching
 - 5.3.1. The module cache
 - 5.3.2. Finders and loaders
 - 5.3.3. Import hooks
 - 5.3.4. The meta path
 - 5.4. Loading
 - 5.4.1. Loaders
 - 5.4.2. Submodules
 - 5.4.3. Module spec
 - 5.4.4. Import-related module attributes
 - 5.4.5. `module.__path__`
 - 5.4.6. Module reprs
 - 5.4.7. Cached bytecode invalidation

Imports

Finding: Absolute and relative

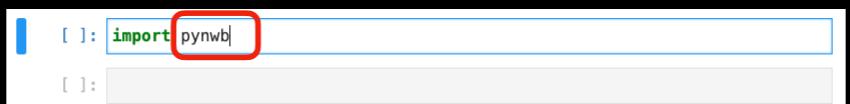
- Import has to know where to look



Imports

Naming

- Import has to know what to map the module it found to in the namespace



A screenshot of a code editor interface. The top bar shows a blue icon followed by '[]:'. Below this is a text input field containing the text 'import pynwb'. A red rectangular box highlights the word 'import'. Below the input field is another row with '[]:' followed by a smaller, empty input field.

Imports

Module structure

- Folder, init

| | |
|---------------------------|----------------------|
| mistune.py | __config__.py |
| mkl | __init__.py |
| mkl_fft | __pycache__ |
| mkl_fft-1.3....8.egg-info | _distributor_init.py |
| mkl_random | _globals.py |
| mkl_random-2.1.dist-info | _pytesttester.py |
| mkl_servic...3.8.egg-info | compat |
| mpl_toolkits | confest.py |
| multidict | core |
| multidict-5.1.0.dist-info | ctypeslib.py |
| nbclassic | distutils |
| nbclassic-0.2.6.dist-info | doc |
| nbclient | dual.py |
| nbclient-0.5.3.dist-info | f2py |
| nbconvert | fft |
| nbconvert-6.0.7.dist-info | lib |
| nbformat | LICENSE.txt |
| nbformat-5.1.3.dist-info | linalg |
| ndx_events | ma |
| ndx_events....0.dist-info | matlib.py |
| nest_asyncio.py | matrixlib |
| networkx | polynomial |
| networkx-2.5.1.dist-info | random |
| notebook | setup.py |
| notebook-6.4.0.dist-info | testing |
| nrrd | tests |
| numexpr | version.py |
| numexpr-2.7.3.dist-info | |
| numpy | |
| numpy-1.18.5.dist-info | |
| olefile | |
| olefile-0.46.dist-info | |

Imports

Create a package

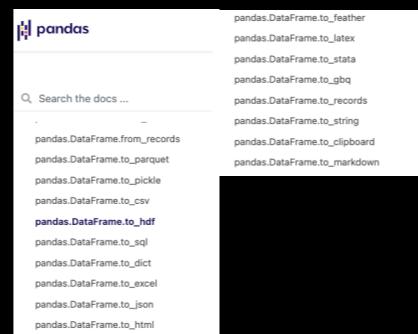
- djd

Exports

- Variables are defined in the scope of your program - whether that is MATLAB, a jupyter notebook, a python IDE, a python script...
- If you want them after the scope of your program doesn't exist anymore, you better save them to disk.

Exports

- `_pickle` is good for saving dictionaries. but you can also pickle anything
 - Some 2 to 3 compatibility issues; also not secure
- `numpy.save`
- Pandas: `df.to_xyz`



Imports v. Exports

- Import and export as used here are not really opposites or paired - import is about bringing in **code**, where export is referring to sending out **data**.
- They are paired here pedagogically by the concept of the scope and namespace of your analysis/coding, but that is it.

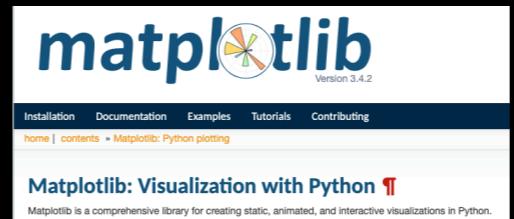


Start up a jupyter notebook

Plotting

the package

- In python, originally trying to catch up with matlab: matplotlib



Plotting

packages

- Seaborn + Pandas
- ipywidgets, Bokeh, plotly/Dash (Jupyter-dash)
- specialized: Brainrender, geoplot,...

