Three key syntaxes to deal with Python objects:

. To access attributes of an object
() To call a function or a method
= To assign

→ You may need to re-learn other syntaxes (i.e., operators) depending on your choice of packages

How to study each package?

0. What kind of objects (i.e., classes, if any) are there?
1. Check out how to *initialize* the object. There may be multiple ways.
2. Read examples to understand *methods* to manipulate the data
3. Check if there are any syntactic Kool-Aid, such as useful *operators*.

DNA
RNA
Protein
etc…

Built-in types
Int, float, complex, etc…
Lists, dictionaries, etc…

Not number-friendly

NumPy

Introduction to NumPy

[1]:
```
a = list(range(10))
a
```

[1]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

[2]:
```
b = a*2
b
```

[2]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

[3]:
```
b = [ x*2 for x in a ]
b
```

[3]:
```
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

```
[1]: a = list(range(10))
     a
```

```
[1]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
[1]: import numpy as np
     a = np.arange(10)
     a
```

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

```
[2]: type(a)
```

```
[2]: numpy.ndarray
```

```
[3]: b = a*2
     b
```

```
[3]: array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])
```

Biopython: Seq, SeqRecord classes

NumPy: ndarray class

How to study each package?

numpy.ndarray

0. What kind of objects (i.e., classes, if any) are there?
1. Check out how to *initialize* the object. There may be multiple ways.
2. Read examples to understand *methods* to manipulate the data
3. Check if there are any syntactic Kool-Aid, such as useful *operators*.

```
[1]: import numpy as np
     a = np.arange(10)
     a
```

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

```
[2]: type(a)
```

```
[2]: numpy.ndarray
```

```
[3]: dir(a)
```

```
[3]: ['T',
      '__abs__',
      '__add__',
      '__and__',
      '__array__',
      '__array_finalize__',
      '__array_function__',
      '__array_interface__',
      '__array_prepare__',
      '__array_priority__',
      '__array_struct__',
      '__array_ufunc__',
      '__array_wrap__',
      '__bool__',
      'all',
      'any',
      'argmax',
      'argmin',
      'argpartition',
      'argsort',
      'astype',
      'base',
      'byteswap',
      'choose',
      'clip',
      'compress',
      'conj',
      'conjugate',
      'copy',
      'cumsum',
      'data',
      'diagonal',
      'dot',
      'dtype',
      'dump',
      'dumps',
      'fill',
      'flags',
      'flat',
      'flatten',
      'getfield',
      'imag',
      'item',
      'itemset',
      'itemsize',
      'max',
      'mean',
      'min',
      'nbytes',
      'ndim',
      'newbyteorder',
      'nonzero',
      'partition',
      'prod',
      'ptp',
      'put',
      'ravel',
      'real',
      'repeat',
      'reshape',
      'resize',
      'round',
      'searchsorted',
      'setfield',
      'setflags',
      'shape',
      'size',
      'sort',
      'squeeze',
      'std',
      'strides',
      'sum',
      'swapaxes',
      'take',
      'tobytes',
      'tofile',
      'tolist',
      'tostring',
      'trace',
      'transpose',
      'var',
      'view']
```

About 1,370,000 results (0.60 seconds)

numpy.org › doc › stable › reference › generated › nu...

numpy.moveaxis — NumPy v1.20 Manual

moveaxis. Move **axes** of an **array** to new positions. Other **axes** remain in their original **order**.

numpy.org › doc › stable › reference › generated › nu...

numpy.swapaxes — NumPy v1.20 Manual

Parameters. aarray_like. Input **array**. axis1int. First **axis**. axis2int. Second **axis**. ... only if the **order** of the **axes** is changed, otherwise the input **array** is returned.

numpy.org › doc › stable › reference

numpy.transpose — Num

transpose. Reverse or permute the
with two **axes**, transpose(a) gives t

stackoverflow.com › questions › rea

Rearranging axes in num

Aug 10, 2019 — **np**. moveaxis(a, sou
rearrange specific dimensions of ar
be used to rearrange all dimensions
2 answers

Swapping the dimensions of a **num**
How to reshape a **Numpy array** fron
How does **numpy**.swapaxes work?
How does **NumPy's** transpose() me
More results from stackoverflow.co

```
[4]: help(a.swapaxes)
```

```
Help on built-in function swapaxes:

swapaxes(...) method of numpy.ndarray instance
    a.swapaxes(axis1, axis2)

    Return a view of the array with `axis1` and `axis2` interchanged.

    Refer to `numpy.swapaxes` for full documentation.

    See Also
    --------
    numpy.swapaxes : equivalent function
```

How to study each package?

numpy.ndarray

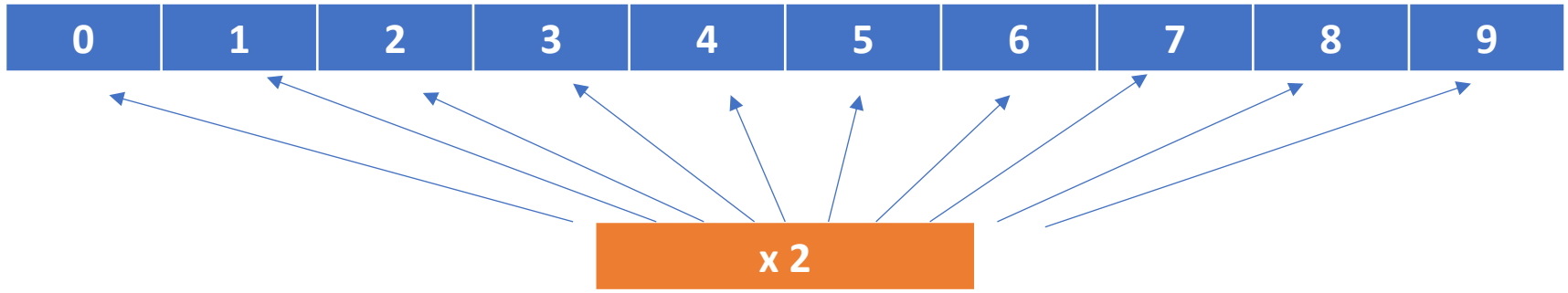
0. What kind of objects (i.e., classes, if any) are there?
1. Check out how to *initialize* the object. There may be multiple ways.
2. Read examples to understand *methods* to manipulate the data
3. Check if there are any syntactic Kool-Aid, such as useful *operators*.

```
[3]: b = a*2
     b
```

```
b = [ x*2 for x in a ]
```

```
[3]: array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])
```

Broadcasting



How to study each package?

`numpy.ndarray`

0. What kind of objects (i.e., classes, if any) are there?
1. Check out how to *initialize* the object. There may be multiple ways.
2. Read examples to understand *methods* to manipulate the data
3. Check if there are any syntactic Kool-Aid, such as useful *operators*.

NumPy