Py4Science 2 @ UND 2009-06-11

```
In [1]: 5/2
Out[1]: 2

In [2]: 5/2.0
Out[2]: 2.5
```

```
In [3]: z = 1 + 2j

In [4]: z.imag
Out[4]: 2.0
```

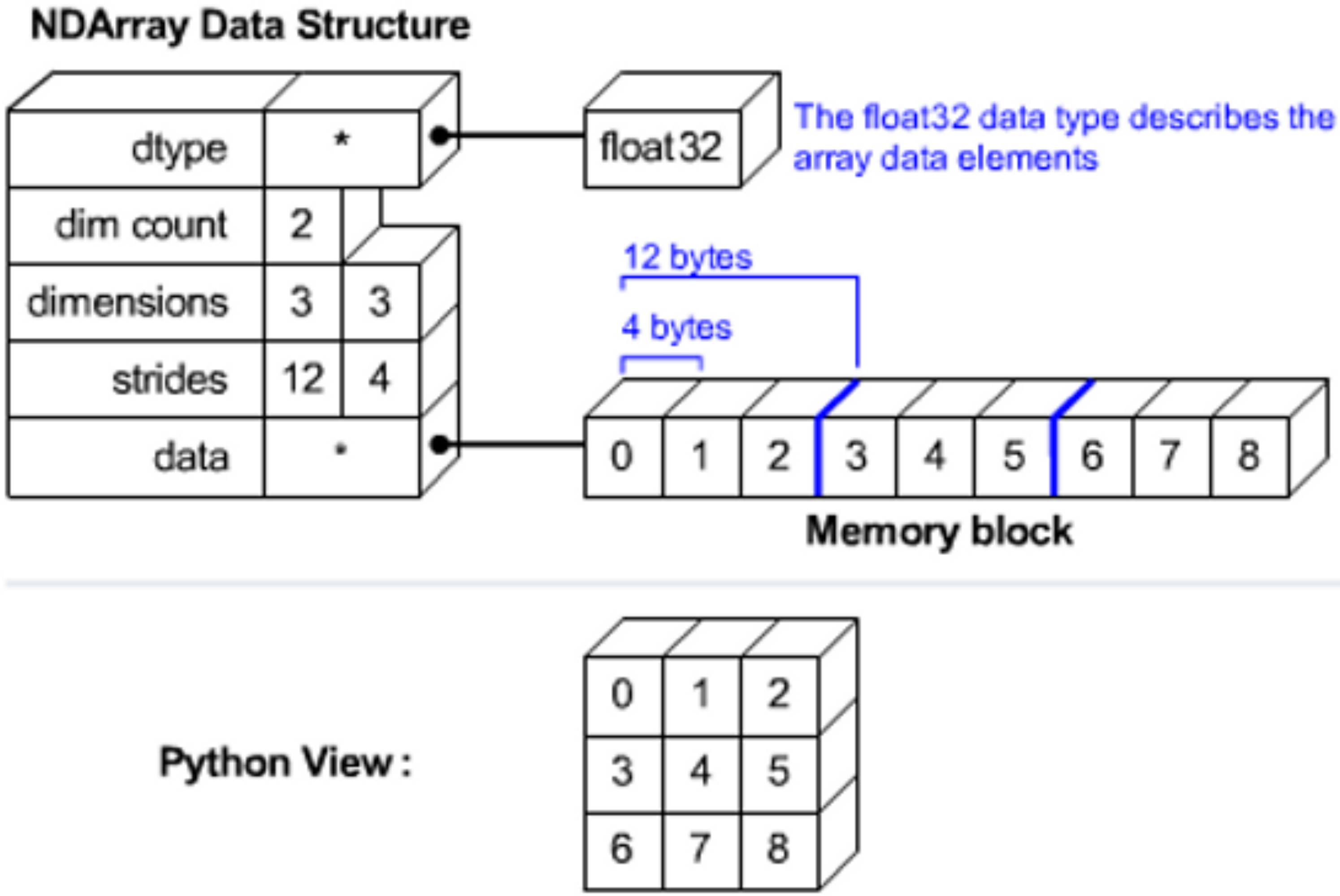
```
In [5]: alist = [1, 'a', [9,8,7], ('abc', {1:'und'})]

In [7]: alist.pop(2)
Out[7]: [9, 8, 7]

In [9]: atuple = (1, 'a', [9,8,7], ('abc', {1:'und'}))
In [10]: range(10)
Out[10]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

In [11]: arange(10)
Out[11]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
In [12]: linspace(0,10, num=100)

In [13]: %time r1=range(10**7)

In [15]: %time r2=xrange(10**7)
```

loops and generators...

- Indentation with 4 spaces
- No tab and space mixing
- Spaces before and / or after operators
- First character of variables is alphabetic
- 80 characters per line
- Code Like a Pythonista: Idiomatic Python

Arrays

Array Data Structure

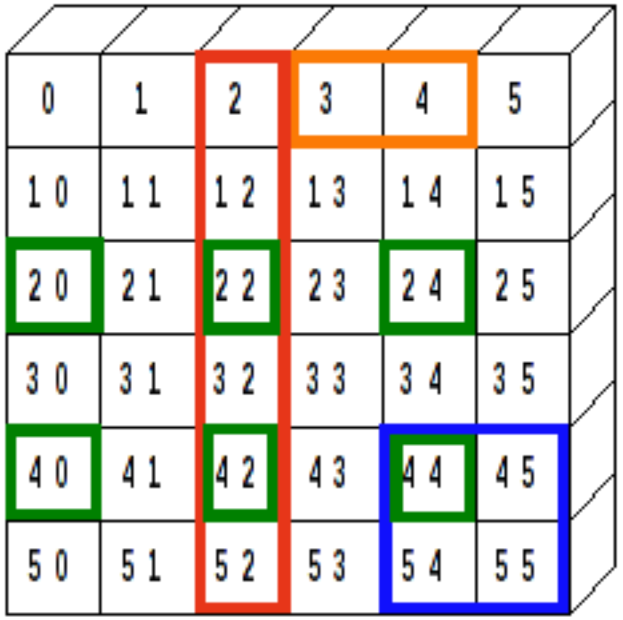


Slicing

```
arr = array([(i, i+1, i+2, i+3, i+4, i+5) for i in range(0,51,10)])
```

```
arr[0, 3:5]
array([3, 4])
```

```
arr[4:,4:]
array([[44, 45],
       [54, 55]])
```



```
arr[:,2]
array([ 2, 12, 22, 32, 42, 52])
```

striding -> arr[2::2]

arr[2::2, ::2]

NumPy

The Case of memmap()

Loading a text-file into an array:

```
In [5]: data = loadtxt('data.txt').T
```

Mapping an array from/to a disk file:

```
In[6]: memarr = memmap('data.
txt',                              dtype='float',
shape=(5,3))
```

The Case of memmap() 2

From Numpy Wiki:

http://docs.scipy.org/numpy/docs/numpy.core.memmap.memmap/

*Create a memory-map to an array stored in a "**binary**" file on disk.*

Loading the ASCII files

```
# Open file
f1 = open(sys.argv[1], 'r')

# Need skiprows to read right section of the file
skiprows = int(f1.next()[:2])

f1.seek(0)
f1header = [f1.readline() for lines in range(skiprows)]

# Close the file
f1.close()

# Read whole data and extract needed variables
tarray = np.loadtxt(sys.argv[1], skiprows=skiprows).T
plot(tarray[0], tarray[5])
```

Programs must be written for people to read, and only incidentally for machines to execute.

—Abelson & Sussman, *Structure and Interpretation of Computer Programs*