

## Pure Python

## Type

$a = 3$	$\Rightarrow$ not equal
$b = 3 \cdot 4$	$\Rightarrow 12$ not
$c = 3 \cdot 369$	$\Rightarrow$ expressed 100
$d = 3 \cdot 3 \rightarrow 9$	$\Rightarrow$ smaller
$e = 3 \cdot 3 \cdot 3$	$\Rightarrow$ smaller
$f = 3 \cdot 3 \cdot 3 \cdot 3$	$\Rightarrow$ not less

## List 8

[illegible]

Dictionary:

```

d = {'foo': 'FOO', 'bar': 'BAR'}      # dictionary
d = dict('foo')                       # from iterable
d = {value for key, value in a.items()} # loop through container
d = dict.fromkeys('foobar', 'no explanation found') # create defaults

```

### Savings

```
g = 'VSE'           # equipment
dset = 422          # access identifier (dataset)
vsel = 'line'        # string constant
l1, l2, dres, nlines = split_string_line(
    l1, l2, dres, nlines, 'VSE', 'dres')
```

### Operation

[illegible]

### Control Flow

[illegible]

## Functions, Classes, Generators, Decorators

[illegible]

## IPython

## carvone

[illegible]

```

# examine history
history
history -r 1000 # shows a lot of last session

```

```
# run shell commands
break # profile command with '*'

# show response
break
```

de-ba-pce

```

4 # create list obj
5 # initialize it to the main file of lib
6 # set location to "utils.py" at line 2
7 # create instance
8
9 # set current position to the end
10 # print the "main" variable
11 # pretty print the "main" variable
12 # stop this subroutine
13
14 # print arguments that a function received
15 # store all variables as local scope
16 # store all variables as global scope

```

command line

```

[option -qk -- quiet] [option -qk]    # quiet after exception
[option -j -- quiet] [option -j]      # quiet after fail

```

## NutriPy (Nutri Supply, Inc.)

## array initialization

[illegible][Downloaded from](#)

<code>u = np.arange(100)</code>	→ initialize array with 0 to 99
<code>u[0] = 0</code>	→ set the first element to zero
<code>u[99] = 0</code>	→ set the last element to zero
<code>u.shape[0]</code>	→ gives the form (number of values)
	→ number to value array
<code>u[0], u[-1], u[50]</code>	→ return array index values of the array
<code>u = u.reshape(100, 10)</code>	→ transform to 10 x 10 matrix
<code>u.T</code>	→ return transposed array
<code>u = np.transpose(u, [1, 0])</code>	→ transform array to the same order
<code>u[u &gt; 0]</code>	→ return only nonnegative samples

## array priorities and operations

```

a.shape           # a tuple with the lengths of multi-dim a
a.ndim            # length of a
a.size            # number of elements (total)
a.dtype.name      # str: array name
a.flags.c_contiguous # contiguous array in C dimension
a.flags.f_contiguous # Fortran contiguous
a.flags.owndata    # True if owned by this array
a.data            # pointer to the start of the array data
a.itemsize        # number of bytes of one data element
a.tolist()         # convert to a Python list of the values
a[0,0].item()     # True if all elements are True
a.any()           # True if any elements are True
a.all()           # True if all elements are True
a.nonzero()       # returns an ndarray with the indices of all the non-zero

```

### boardroom's journey

<code>is &amp; b</code>	<code>&amp;</code> returns array with boolean values
<code>(a &amp; b) &amp; c</code>	<code>&amp;</code> elementwise logical and
<code>(a &amp; b)   c</code>	<code> </code> elementwise logical or
<code>~a</code>	<code>~</code> logical NOT (inverts array)

elementwise operations and math functions

```

u ~ N
# distributed across units
u ~ N
# distributed across studies
u ~ N
# distributed across sites
u ~ N
# distributed across 100 days (for discussion by site)
np.mean(u)
# expected (population) mean
u ~ N
# u is the mean
np.mean(u)
# size
np.mean(u)
# variance
np.mean(u)
# precision
np.mean(u)
# degrees of freedom
np.mean(u)
# variance in degrees
np.mean(u)
# one piece of it (e.g.
np.mean(u)

```

## inner / outer products

```

%0 = add i32 @4, @0;               @0 = add i32 @4, @0
%1 = extractvalue %0, 0;            @0 = extractvalue %0, 0
%2 = extractvalue %0, 1;            @1 = extractvalue %0, 1
%3 = add i32 @4, @0;                @0 = add i32 @4, @0
%4 = add i32 @4, @1;                @1 = add i32 @4, @1
%5 = add i32 @4, @2;                @2 = add i32 @4, @2
%6 = add i32 @4, @3;                @3 = add i32 @4, @3
%7 = add i32 @4, @4;                @4 = add i32 @4, @4
%8 = add i32 @4, @5;                @5 = add i32 @4, @5
%9 = add i32 @4, @6;                @6 = add i32 @4, @6
%10 = add i32 @4, @7;               @7 = add i32 @4, @7
%11 = add i32 @4, @8;               @8 = add i32 @4, @8
%12 = add i32 @4, @9;               @9 = add i32 @4, @9
%13 = add i32 @4, @10;              @10 = add i32 @4, @10
%14 = add i32 @4, @11;              @11 = add i32 @4, @11
%15 = add i32 @4, @12;              @12 = add i32 @4, @12
%16 = add i32 @4, @13;              @13 = add i32 @4, @13
%17 = add i32 @4, @14;              @14 = add i32 @4, @14
%18 = add i32 @4, @15;              @15 = add i32 @4, @15
%19 = add i32 @4, @16;              @16 = add i32 @4, @16
%20 = add i32 @4, @17;              @17 = add i32 @4, @17
%21 = add i32 @4, @18;              @18 = add i32 @4, @18
%22 = add i32 @4, @19;              @19 = add i32 @4, @19
%23 = add i32 @4, @20;              @20 = add i32 @4, @20
%24 = add i32 @4, @21;              @21 = add i32 @4, @21
%25 = add i32 @4, @22;              @22 = add i32 @4, @22
%26 = add i32 @4, @23;              @23 = add i32 @4, @23
%27 = add i32 @4, @24;              @24 = add i32 @4, @24
%28 = add i32 @4, @25;              @25 = add i32 @4, @25
%29 = add i32 @4, @26;              @26 = add i32 @4, @26
%30 = add i32 @4, @27;              @27 = add i32 @4, @27
%31 = add i32 @4, @28;              @28 = add i32 @4, @28
%32 = add i32 @4, @29;              @29 = add i32 @4, @29
%33 = add i32 @4, @30;              @30 = add i32 @4, @30
%34 = add i32 @4, @31;              @31 = add i32 @4, @31
%35 = add i32 @4, @32;              @32 = add i32 @4, @32
%36 = add i32 @4, @33;              @33 = add i32 @4, @33
%37 = add i32 @4, @34;              @34 = add i32 @4, @34
%38 = add i32 @4, @35;              @35 = add i32 @4, @35
%39 = add i32 @4, @36;              @36 = add i32 @4, @36
%40 = add i32 @4, @37;              @37 = add i32 @4, @37
%41 = add i32 @4, @38;              @38 = add i32 @4, @38
%42 = add i32 @4, @39;              @39 = add i32 @4, @39
%43 = add i32 @4, @40;              @40 = add i32 @4, @40
%44 = add i32 @4, @41;              @41 = add i32 @4, @41
%45 = add i32 @4, @42;              @42 = add i32 @4, @42
%46 = add i32 @4, @43;              @43 = add i32 @4, @43
%47 = add i32 @4, @44;              @44 = add i32 @4, @44
%48 = add i32 @4, @45;              @45 = add i32 @4, @45
%49 = add i32 @4, @46;              @46 = add i32 @4, @46
%50 = add i32 @4, @47;              @47 = add i32 @4, @47
%51 = add i32 @4, @48;              @48 = add i32 @4, @48
%52 = add i32 @4, @49;              @49 = add i32 @4, @49
%53 = add i32 @4, @50;              @50 = add i32 @4, @50
%54 = add i32 @4, @51;              @51 = add i32 @4, @51
%55 = add i32 @4, @52;              @52 = add i32 @4, @52
%56 = add i32 @4, @53;              @53 = add i32 @4, @53
%57 = add i32 @4, @54;              @54 = add i32 @4, @54
%58 = add i32 @4, @55;              @55 = add i32 @4, @55
%59 = add i32 @4, @56;              @56 = add i32 @4, @56
%60 = add i32 @4, @57;              @57 = add i32 @4, @57
%61 = add i32 @4, @58;              @58 = add i32 @4, @58
%62 = add i32 @4, @59;              @59 = add i32 @4, @59
%63 = add i32 @4, @60;              @60 = add i32 @4, @60
%64 = add i32 @4, @61;              @61 = add i32 @4, @61
%65 = add i32 @4, @62;              @62 = add i32 @4, @62
%66 = add i32 @4, @63;              @63 = add i32 @4, @63
%67 = add i32 @4, @64;              @64 = add i32 @4, @64
%68 = add i32 @4, @65;              @65 = add i32 @4, @65
%69 = add i32 @4, @66;              @66 = add i32 @4, @66
%70 = add i32 @4, @67;              @67 = add i32 @4, @67
%71 = add i32 @4, @68;              @68 = add i32 @4, @68
%72 = add i32 @4, @69;              @69 = add i32 @4, @69
%73 = add i32 @4, @70;              @70 = add i32 @4, @70
%74 = add i32 @4, @71;              @71 = add i32 @4, @71
%75 = add i32 @4, @72;              @72 = add i32 @4, @72
%76 = add i32 @4, @73;              @73 = add i32 @4, @73
%77 = add i32 @4, @74;              @74 = add i32 @4, @74
%78 = add i32 @4, @75;              @75 = add i32 @4, @75
%79 = add i32 @4, @76;              @76 = add i32 @4, @76
%80 = add i32 @4, @77;              @77 = add i32 @4, @77
%81 = add i32 @4, @78;              @78 = add i32 @4, @78
%82 = add i32 @4, @79;              @79 = add i32 @4, @79
%83 = add i32 @4, @80;              @80 = add i32 @4, @80
%84 = add i32 @4, @81;              @81 = add i32 @4, @81
%85 = add i32 @4, @82;              @82 = add i32 @4, @82
%86 = add i32 @4, @83;              @83 = add i32 @4, @83
%87 = add i32 @4, @84;              @84 = add i32 @4, @84
%88 = add i32 @4, @85;              @85 = add i32 @4, @85
%89 = add i32 @4, @86;              @86 = add i32 @4, @86
%90 = add i32 @4, @87;              @87 = add i32 @4, @87
%91 = add i32 @4, @88;              @88 = add i32 @4, @88
%92 = add i32 @4, @89;              @89 = add i32 @4, @89
%93 = add i32 @4, @90;              @90 = add i32 @4, @90
%94 = add i32 @4, @91;              @91 = add i32 @4, @91
%95 = add i32 @4, @92;              @92 = add i32 @4, @92
%96 = add i32 @4, @93;              @93 = add i32 @4, @93
%97 = add i32 @4, @94;              @94 = add i32 @4, @94
%98 = add i32 @4, @95;              @95 = add i32 @4, @95
%99 = add i32 @4, @96;              @96 = add i32 @4, @96
%100 = add i32 @4, @97;             @97 = add i32 @4, @97
%101 = add i32 @4, @98;             @98 = add i32 @4, @98
%102 = add i32 @4, @99;             @99 = add i32 @4, @99
%103 = add i32 @4, @100;            @100 = add i32 @4, @100
%104 = add i32 @4, @101;            @101 = add i32 @4, @101
%105 = add i32 @4, @102;            @102 = add i32 @4, @102
%106 = add i32 @4, @103;            @103 = add i32 @4, @103
%107 = add i32 @4, @104;            @104 = add i32 @4, @104
%108 = add i32 @4, @105;            @105 = add i32 @4, @105
%109 = add i32 @4, @106;            @106 = add i32 @4, @106
%110 = add i32 @4, @107;            @107 = add i32 @4, @107
%111 = add i32 @4, @108;            @108 = add i32 @4, @108
%112 = add i32 @4, @109;            @109 = add i32 @4, @109
%113 = add i32 @4, @110;            @110 = add i32 @4, @110
%114 = add i32 @4, @111;            @111 = add i32 @4, @111
%115 = add i32 @4, @112;            @112 = add i32 @4, @112
%116 = add i32 @4, @113;            @113 = add i32 @4, @113
%117 = add i32 @4, @114;            @114 = add i32 @4, @114
%118 = add i32 @4, @115;            @115 = add i32 @4, @115
%119 = add i32 @4, @116;            @116 = add i32 @4, @116
%120 = add i32 @4, @117;            @117 = add i32 @4, @117
%121 = add i32 @4, @118;            @118 = add i32 @4, @118
%122 = add i32 @4, @119;            @119 = add i32 @4, @119
%123 = add i32 @4, @120;            @120 = add i32 @4, @120
%124 = add i32 @4, @121;            @121 = add i32 @4, @121
%125 = add i32 @4, @122;            @122 = add i32 @4, @122
%126 = add i32 @4, @123;            @123 = add i32 @4, @123
%127 = add i32 @4, @124;            @124 = add i32 @4, @124
%128 = add i32 @4, @125;            @125 = add i32 @4, @125
%129 = add i32 @4, @126;            @126 = add i32 @4, @126
%130 = add i32
```

readings/writing files

```

sp. fromfile("data/foodpart", "group", "foodcat", "count")  # counts data from file
sp. loadfile("data/foodpart", "group", "foodcat", "count")  # load data from file
sp. loadfile("data/foodpart", "group", "foodcat", "count")  # load data from file
sp. loadfile("data/foodpart", "group", "foodcat", "count")  # load data from file

```

interpolation, integration, optimization

[illegible]

## 11

<code>sp.FR_1/10</code>	$\rightarrow$ complex Fourier transform of $x$
<code>F = sp.FR_1/FR_Freq(2000)</code>	$\rightarrow$ FR frequency
<code>sp.FR_1/64N/2</code>	$\rightarrow$ shift to zero frequency for the real/imag
<code>sp.FR_1/FR_I</code>	$\rightarrow$ real Fourier transform of $x$
<code>sp.FR_1/FR_Freq(2000)</code>	$\rightarrow$ real FR frequency

rounding

```

ms_avail()  # number for nearest upper end
ms_lower()  # number for nearest lower end
ms_mid()    # number for nearest int

```

Keywords: *variegation*

```
from sklearn import normal, rand, uniform, random
normal = normal.Normal()
uniform = uniform.Uniform()
random = random.RandomState()

normal(100)
uniform(10, 100)
random(10, 100)
```

Matplotlib (<https://matplotlib.org/>)

### Figures and axes

```
Fig = glib.figure(Fig.knownKeys, Fig)  # initialize figure
ax = Fig.add_subplot(1, 1, 1)         # add second subplot to a 1x1 grid
Fig.axes = glib.subplots(1, 1, Fig.knownKeys, Fig) # Fig.axes is a 1-tuple of axes
ax = Fig.add_subplot(1, 1, 1, known, aspect, figsize) # add second axis
```

## figures and axes properties

[illegible]

## plotting routines

[illegible]Science | [REPORT • POLYMER NANOPARTICLES](#) 1133

industry guidelines

```
from scipy.interpolate import map_coordinates    # interpolate data
pts_new = map_coordinates(pts, flow_indices,     # do index pts to new
                           order=1)            # order
```

[Send us your comments](#)

**From copy arguments:** `import quat`  $\rightarrow$  defines integral of system  
value  $\rightarrow$  number, low limit, up limit  $\rightarrow$  function-argument