

## 0.1 Special Arrays

Functions are available to construct a number of useful, frequently encountered arrays.

### 0.1.1 ones

`ones` generates an array of 1s and is generally called with one argument, a tuple, containing the size of each dimension. `ones` takes an optional second argument (`dtype`) to specify the data type. If omitted, the data type is float.

```
>>> M, N = 5, 5
>>> x = ones((M,N)) # M by N array of 1s
>>> x = ones((M,M,N)) # 3D array
>>> x = ones((M,N), dtype='int32') # 32bit integers
```

`ones_like` creates an array with the same shape and data type as the input. Calling `ones_like(x)` is equivalent to calling `ones(x.shape,x.dtype)`.

### 0.1.2 zeros

`zeros` produces an array of 0s in the same way `ones` produces an array of 1s, and commonly used to initialize an array to hold values generated by another procedure. `zeros` takes an optional second argument (`dtype`) to specify the data type. If omitted, the data type is float.

```
>>> x = zeros((M,N)) # M by N array of 0s
>>> x = zeros((M,M,N)) # 3D array of 0s
>>> x = zeros((M,N),dtype='int64') # 64 bit integers
```

`zeros_like` creates an array with the same size and shape as the input. Calling `zeros_like(x)` is equivalent to calling `zeros(x.shape,x.dtype)`.

### 0.1.3 empty

`empty` produces an empty (uninitialized) array to hold values generated by another procedure. `empty` takes an optional second argument (`dtype`) which specifies the data type. If omitted, the data type is float.

```
>>> x = empty((M,N)) # M by N empty array
>>> x = empty((N,N,N,N)) # 4D empty array
>>> x = empty((M,N),dtype='float32') # 32bit
```

## 0.2 floats (single precision)

Using `empty` is slightly faster than calling `zeros` since it does not assign 0 to all elements of the array – the “empty” array created will be populated with (essential random) non-zero values. `empty_like` creates an array with the same size and shape as the input. Calling `empty_like(x)` is equivalent to calling `empty(x.shape,x.dtype)`.

### 0.2.1 `eye`, `identity`

`eye` generates an identity array – an array with ones on the diagonal, zeros everywhere else. Normally, an identity array is square and so usually only 1 input is required. More complex zero-padded arrays containing an identity matrix can be produced using optional inputs.

```
>>> In = eye(N)
```

`identity` is a virtually identical function with similar use, `In = identity(N)`.