

# HOW TO RESHAPE ARRAYS WITH NUMPY RESHAPE

SHARP SIGHT

# WHAT YOU'LL LEARN

- How to change the shape of a Numpy array
- Two tools:
  - the `np.reshape()` function
  - the `reshape()` method
- Examples of reshaping data

# A QUICK INTRODUCTION TO NUMPY RESHAPE

# REMEMBER: ALL NUMPY ARRAYS HAVE A "SHAPE"

The shape is essentially the number of rows and columns

shape (3,)

7	7	7
---	---	---

shape (2, 3)

7	7	7
7	7	7

shape (3, 2)

7	7
7	7
7	7

# NUMPY RESHAPE CHANGES THE SHAPE OF A NUMPY ARRAY

shape (2, 3)

7	7	7
7	7	7

`numpy.reshape()`



shape (3, 2)

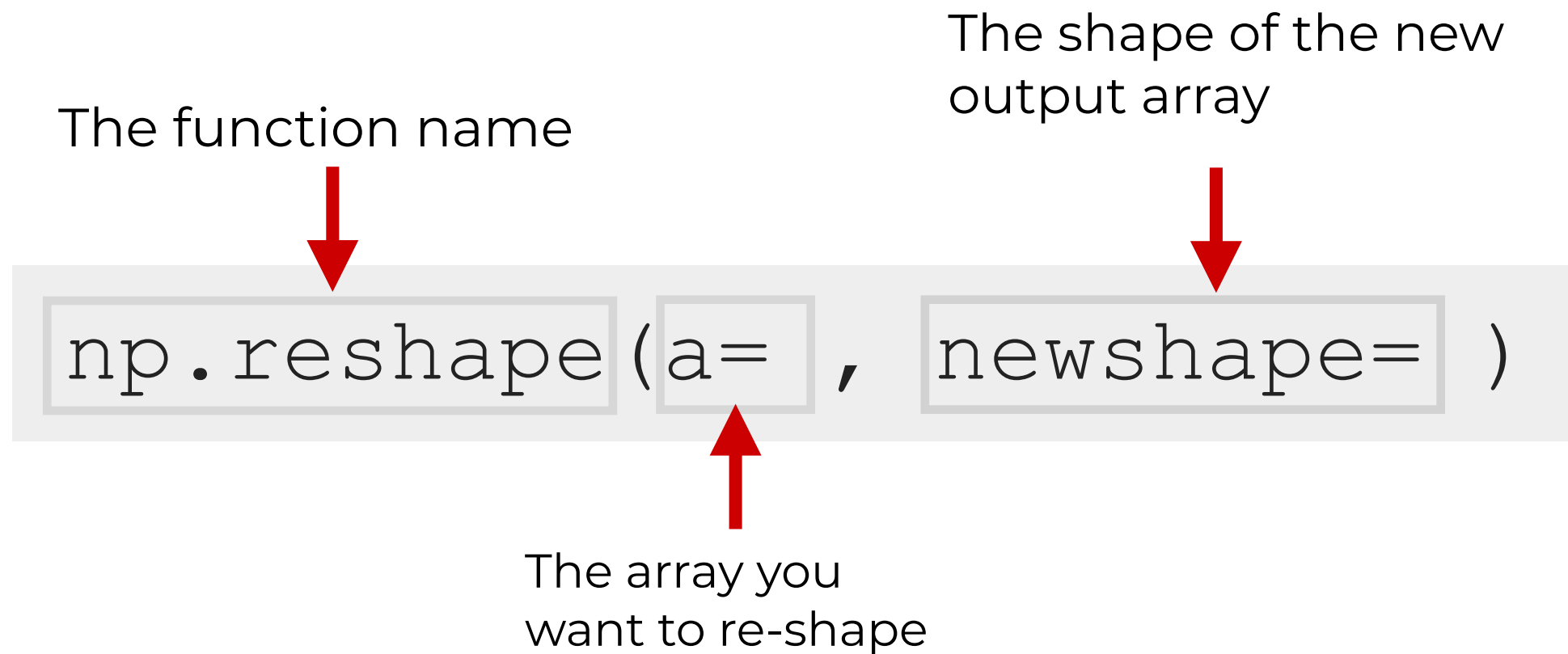
7	7
7	7
7	7

# THE SYNTAX OF THE NUMPY RESHAPE FUNCTION

# THERE ARE TWO DIFFERENT VERSIONS OF NUMPY RESHAPE THAT WE WILL COVER

- The Numpy reshape *function*
- The Numpy reshape *method*
- These are similar, but slightly different
- We'll start with the reshape function

# THE SYNTAX OF THE NUMPY RESHAPE FUNCTION





# THE PARAMETERS OF THE NUMPY LINESPACE FUNCTION

Parameter	What it controls	Argument	Required?
a	The input array that you want to re-shape	a Numpy array	Yes
newshape	The new shape of the output array	An integer or tuple of integers	Yes

Note: there is also another parameter called `order`, which we won't cover here.

# EXAMPLES OF NUMPY RESHAPE

# RESHAPE FROM 1-DIMENSIONAL TO 2-DIMENSIONAL

```
np_array_1d = np.arange(start = 0, stop = 6)  
  
np.reshape(a = np_array_1d, newshape = (2,3))
```



# RESHAPE A 2-DIMENSIONAL ARRAY TO A NEW SHAPE

```
np_array_2x3 = np.full(shape = (2,3),fill_value = 7)  
  
np.reshape(a = np_array_2x3, newshape = (3,2))
```

7	7	7
7	7	7

reshape()



7	7
7	7
7	7

# THE SYNTAX OF THE NUMPY RESHAPE *METHOD*

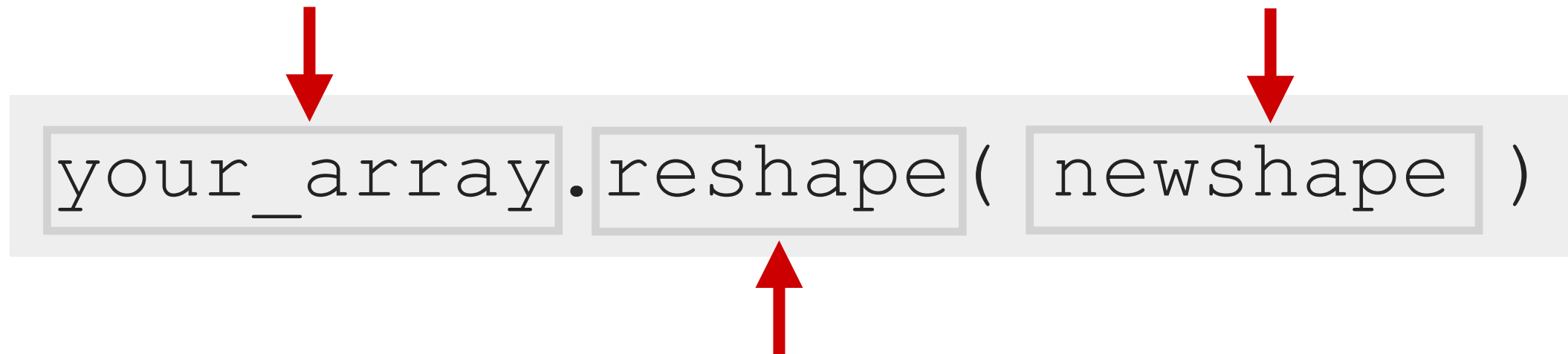
# THE NUMPY RESHAPE *METHOD* IS AN ALTERNATIVE WAY TO RESHAPE ARRAYS

- Similar to the Numpy reshape function but slightly different

# THE SYNTAX OF THE NUMPY RESHAPE METHOD

The name of the  
array you want to  
re-shape

The shape of the new  
output array



The diagram shows the syntax `your_array.reshape(newshape)` within a light gray rectangular box. Three red arrows point to specific parts of the code: one points down to `your_array`, one points down to `newshape`, and one points up to `reshape`. Each arrow is accompanied by a text label explaining its target.

```
your_array.reshape(newshape)
```

The method name

Note that you can't explicitly reference newshape ...  
you can only provide an argument to the method

# EXAMPLE OF THE NUMPY RESHAPE *METHOD*



# RESHAPE FROM 1-DIMENSIONAL TO 2-DIMENSIONAL

```
np_array_1d = np.arange(start = 0, stop = 6)  
  
np_array_1d.reshape( (2,3) )
```



Note: this does not change the original array, it creates a new array

RECAP

# RECAP OF WHAT WE LEARNED

- Numpy has two tools to reshape existing numpy arrays
  - the reshape *function*
  - the reshape *method*
- The syntax of NumPy reshape
- How to use `np.reshape` to reshape NumPy arrays