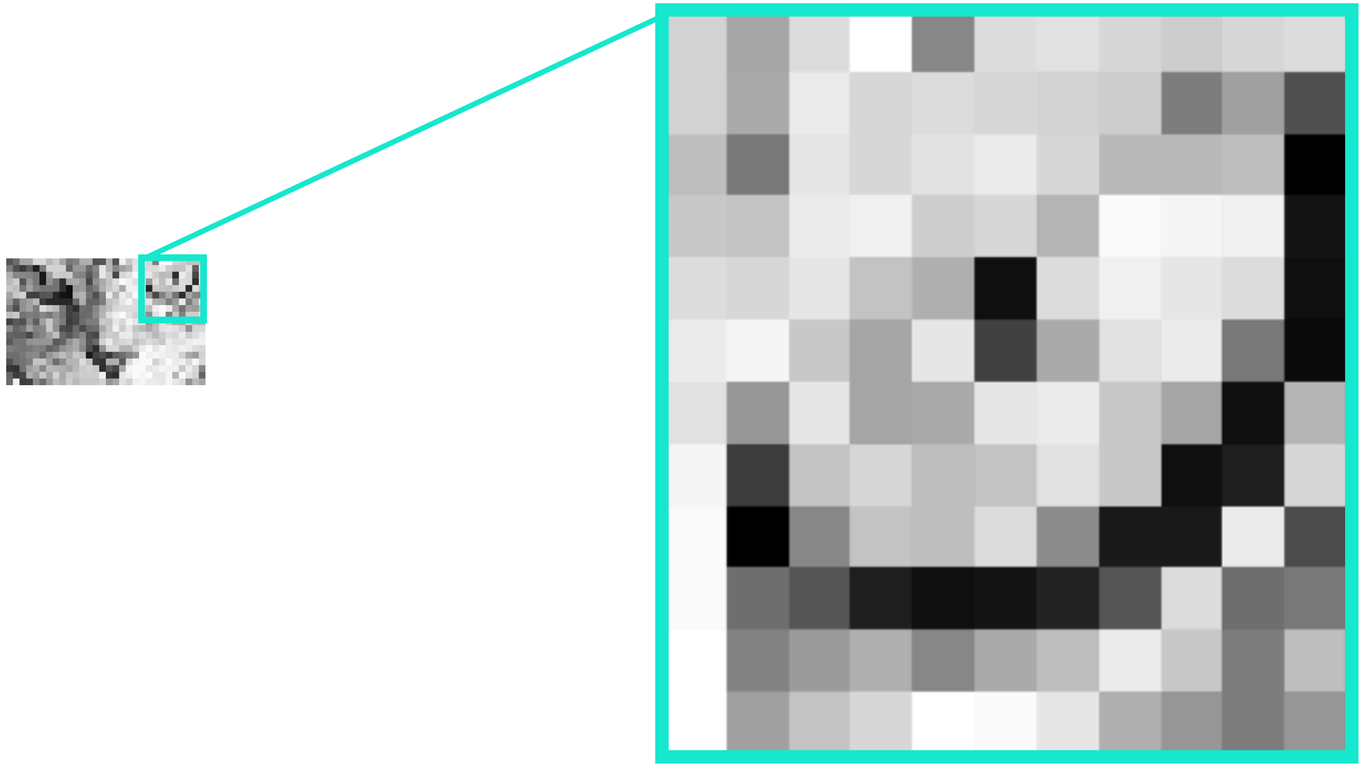
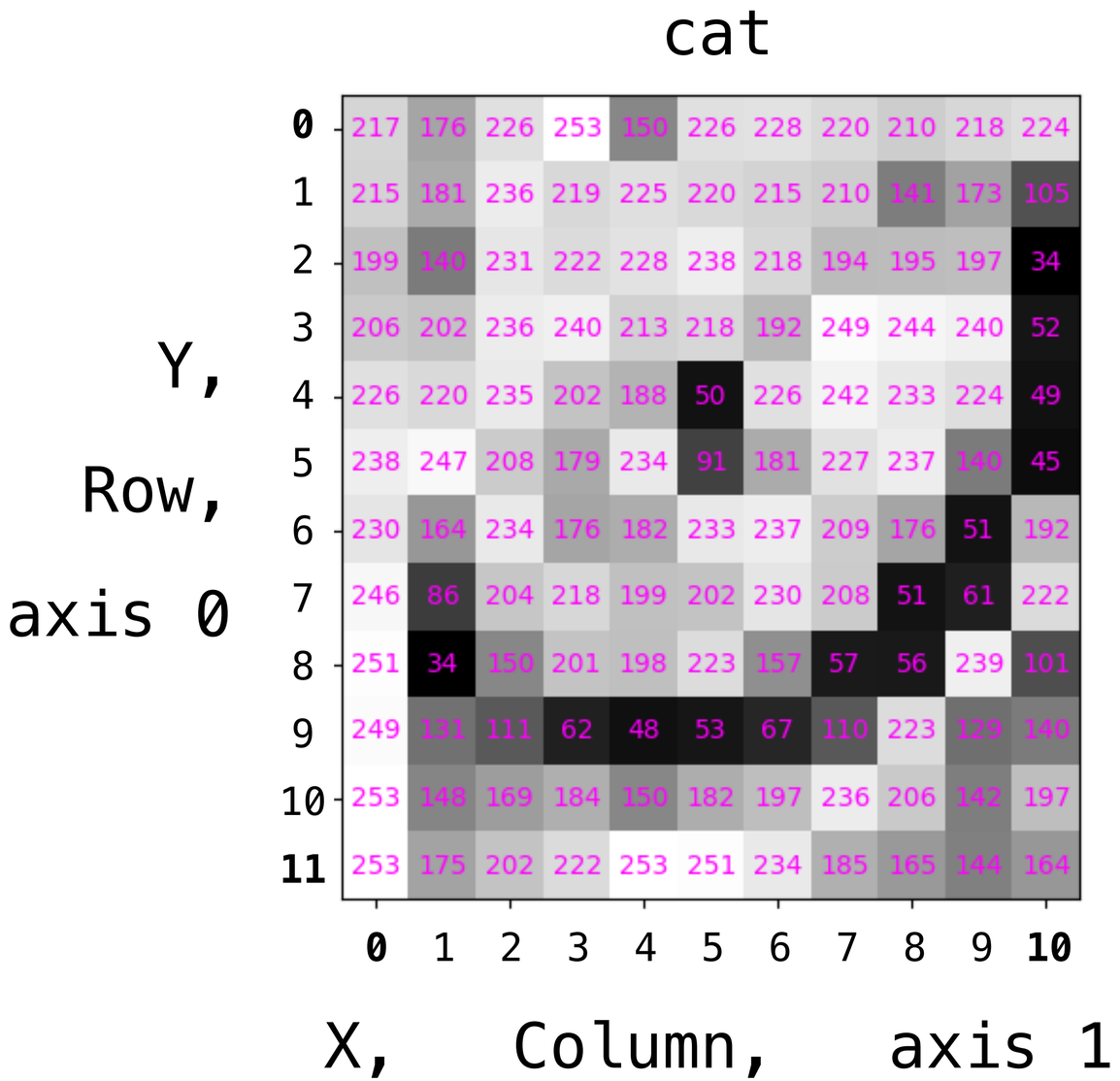
Python for bioimage analysis

The data

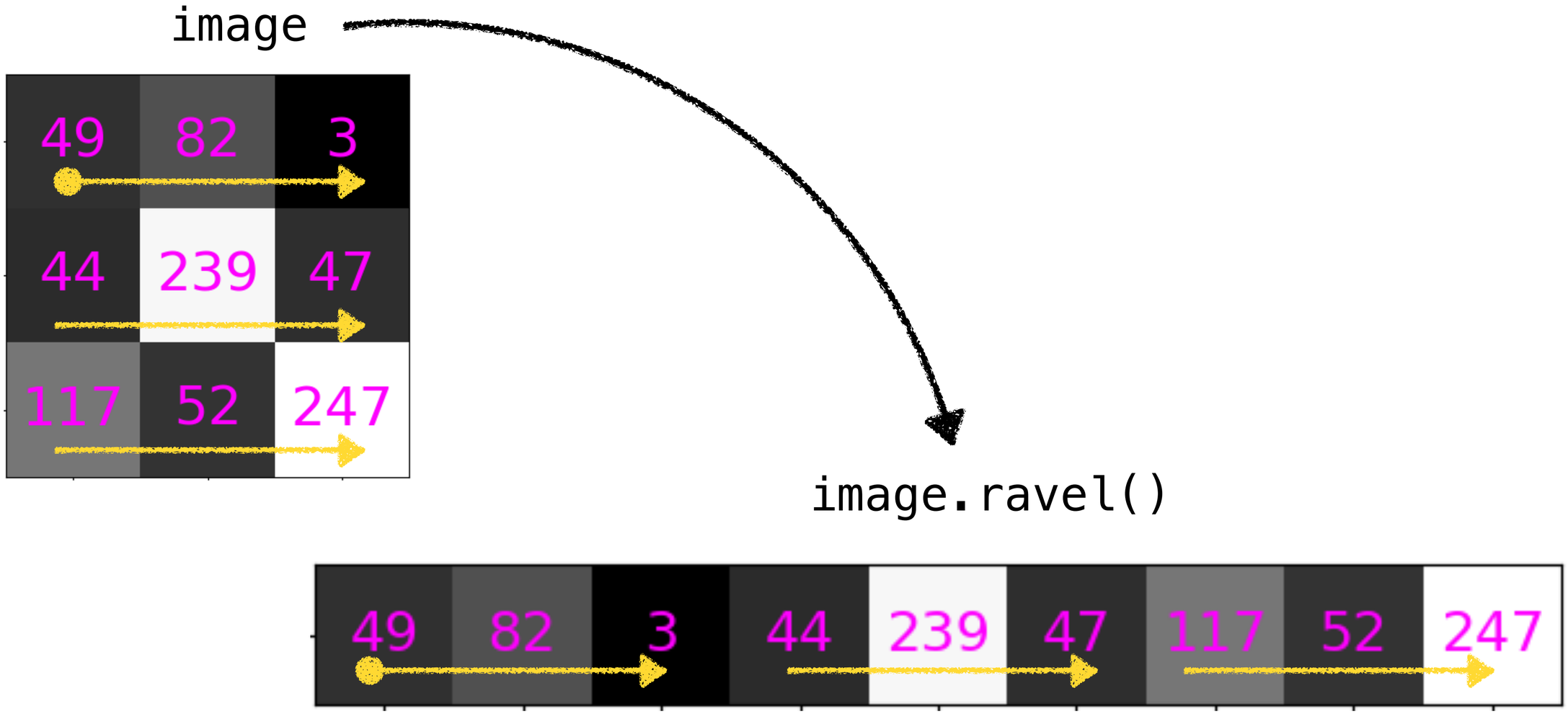


The data

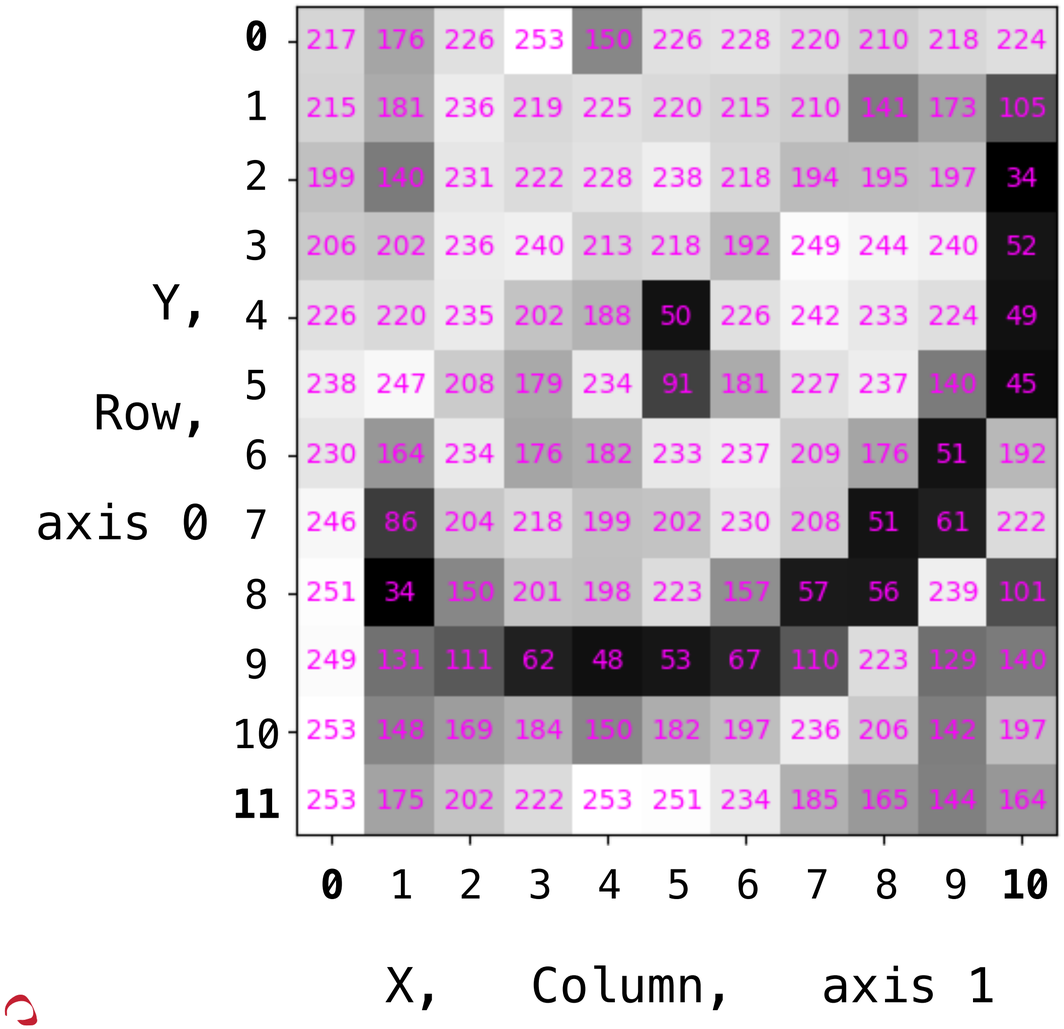
```
Type of the image: <class 'numpy.ndarray'>

Datatype of the image: uint8

Shape of the image: (12, 11)

Minimum pixel value: 34

Maximum pixel value: 253

Mean pixel value: 184.17
```

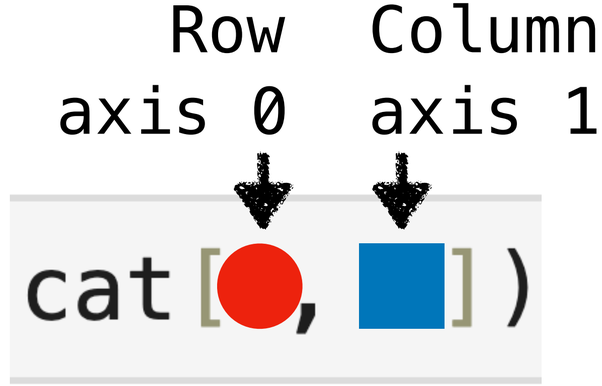


Plot a Histogram

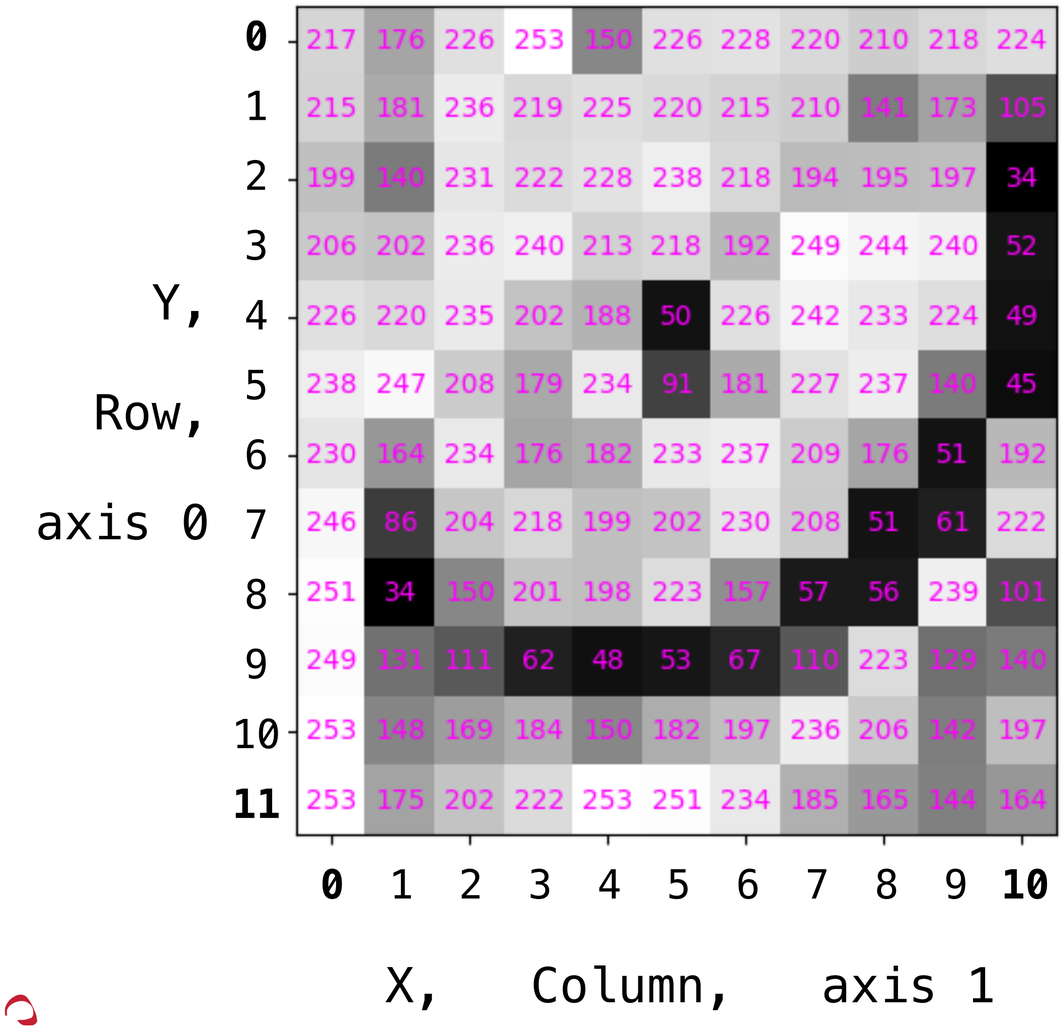


Indexing: individual entries





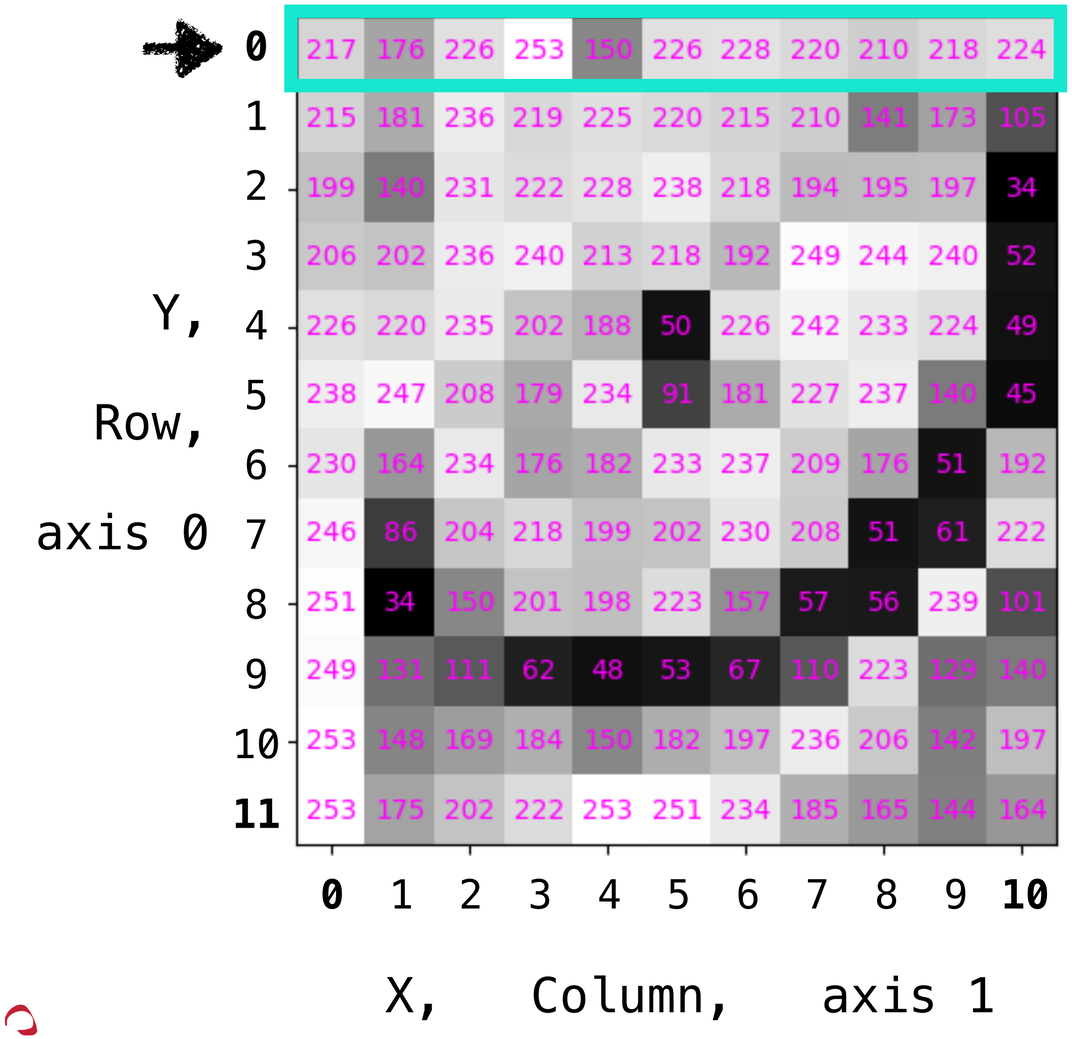
Indexing: individual entries



0th row 1st column

```
print(cat[0, 1])
```

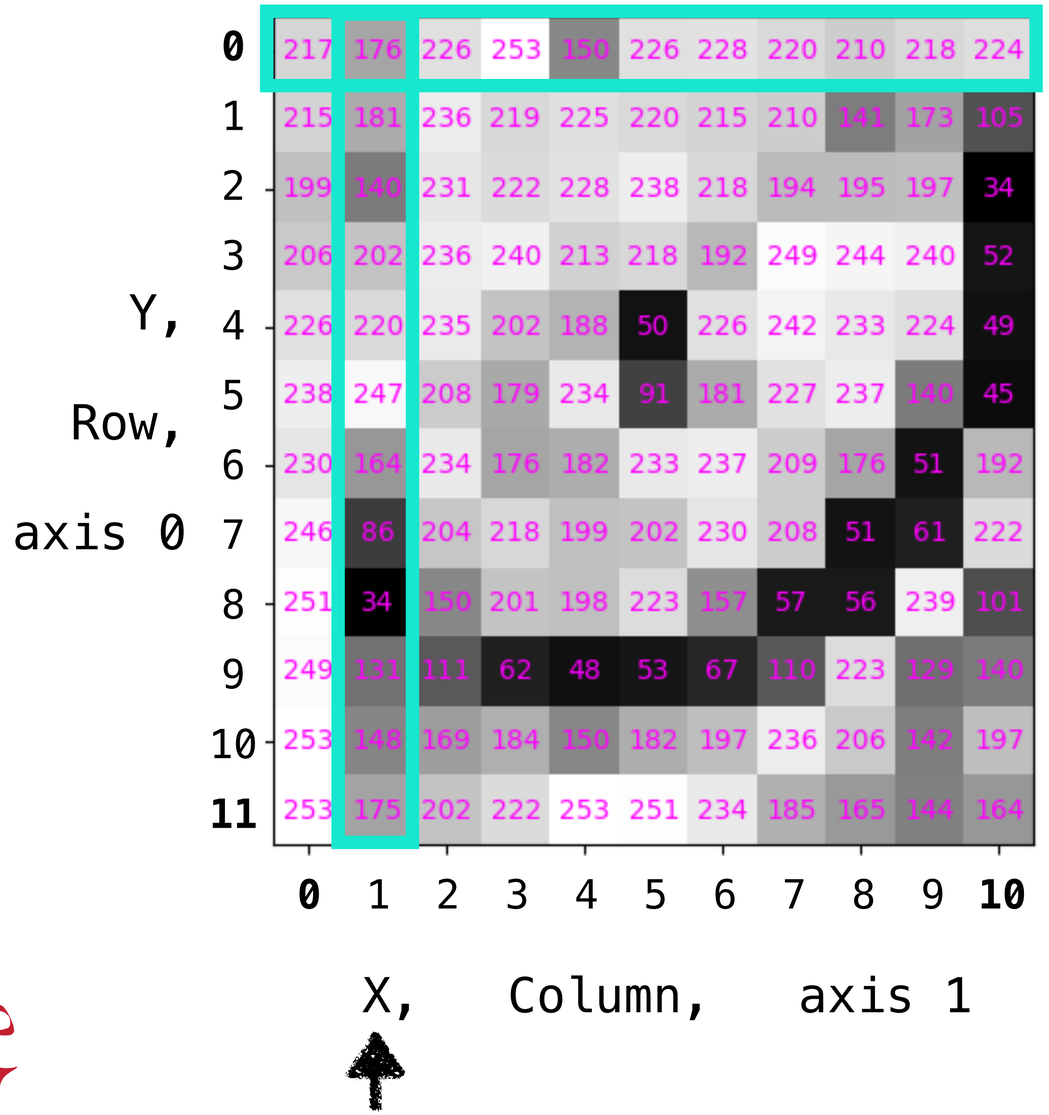
Indexing: individual entries



0th row 1st column

```
print(cat[0, 1])
```

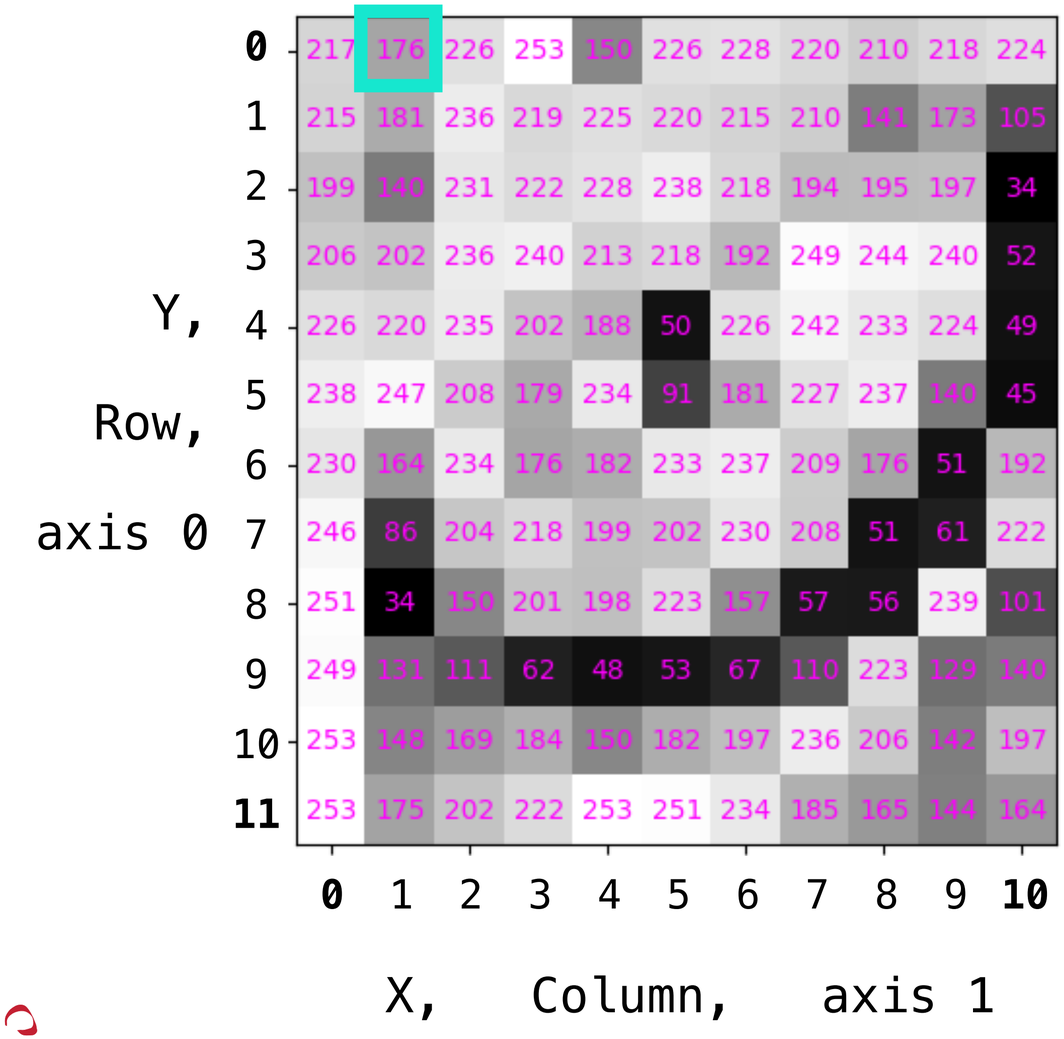
Indexing: individual entries



0th row 1st column

```
print(cat[0, 1])
```

Indexing: individual entries

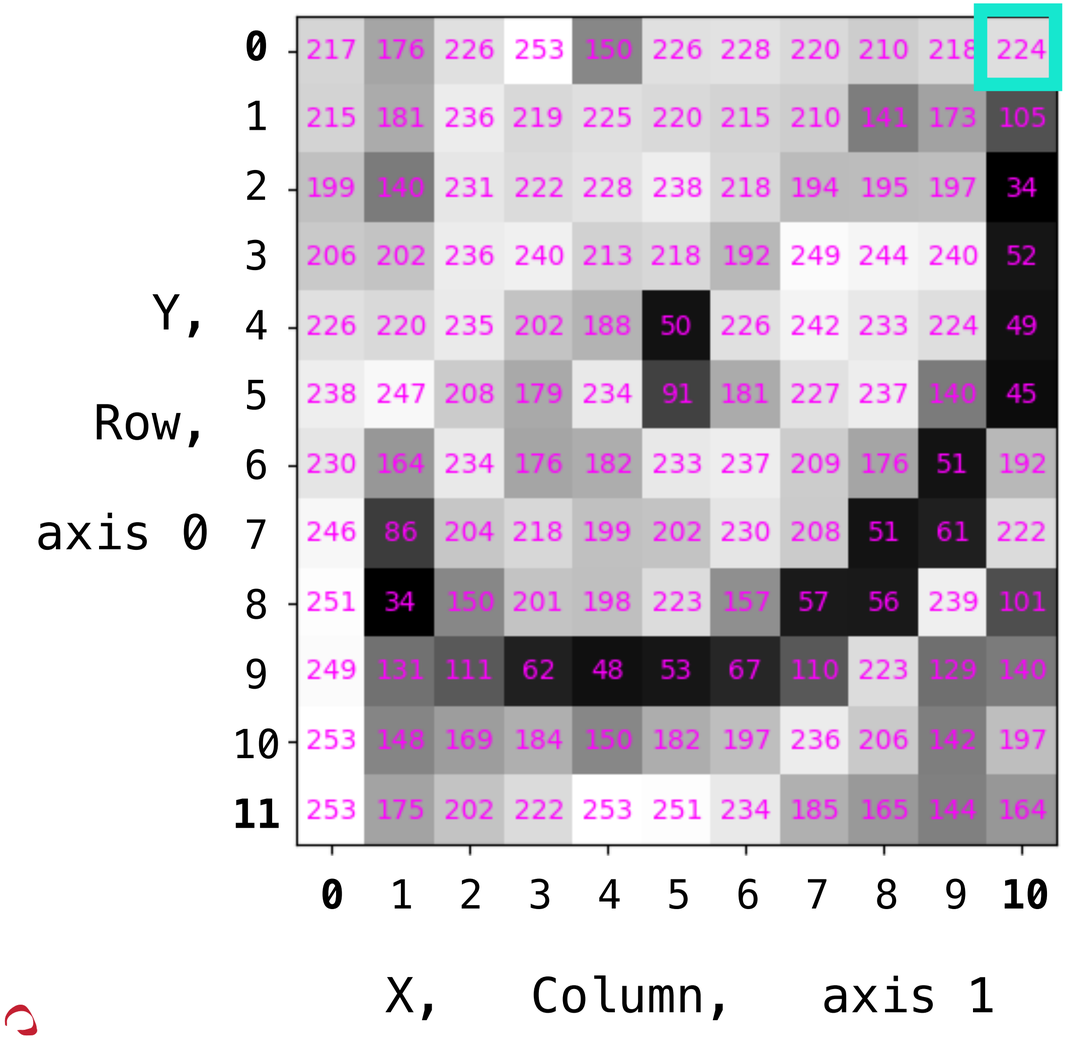


0th row 1st column

```
print(cat[0, 1])
```

176

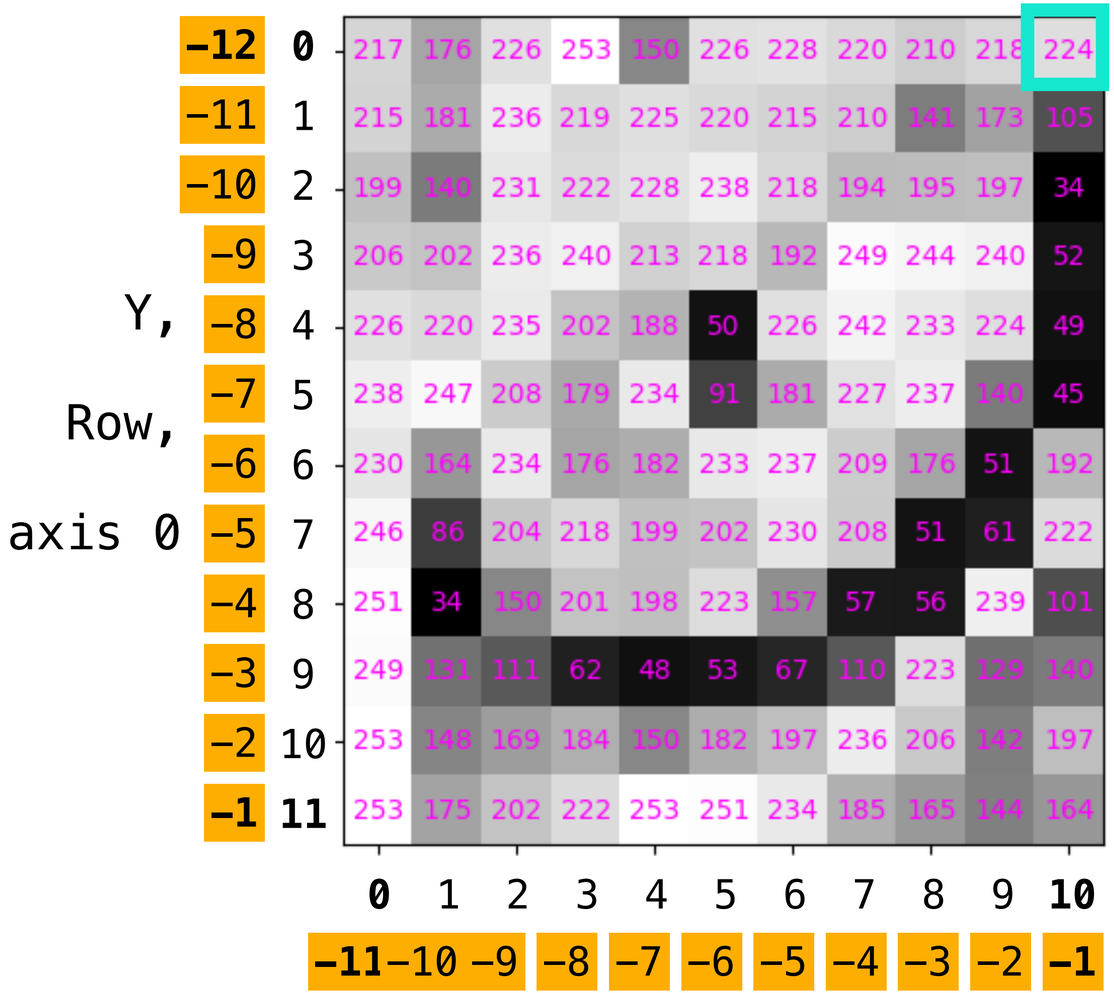
Indexing: individual entries




```
print(cat[0, 10])
```

```
224
```

Indexing: individual entries




```
print(cat[0, 10])
```

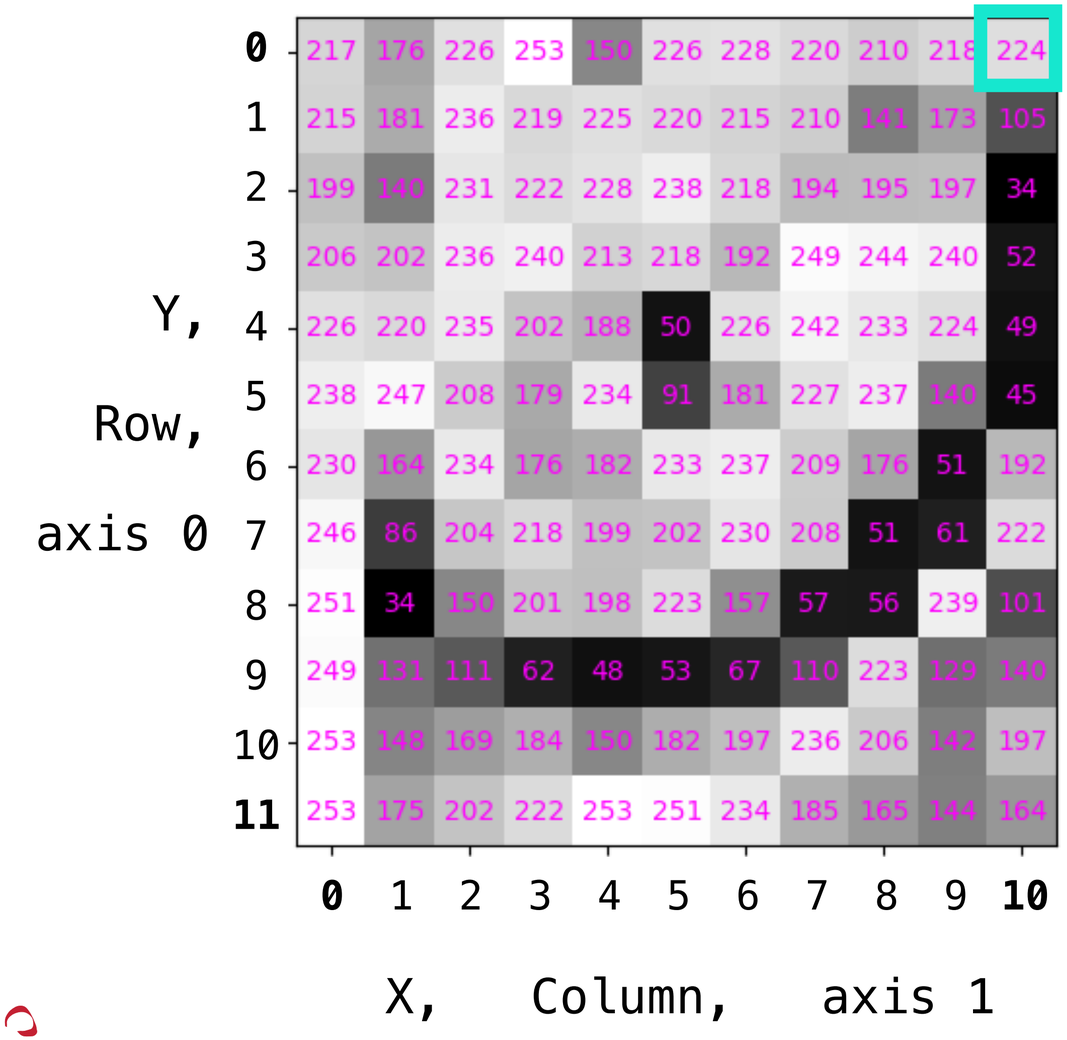
```
224
```

Negative indexing

```
print(cat[0, -1])
```

```
224
```

Indexing: individual entries

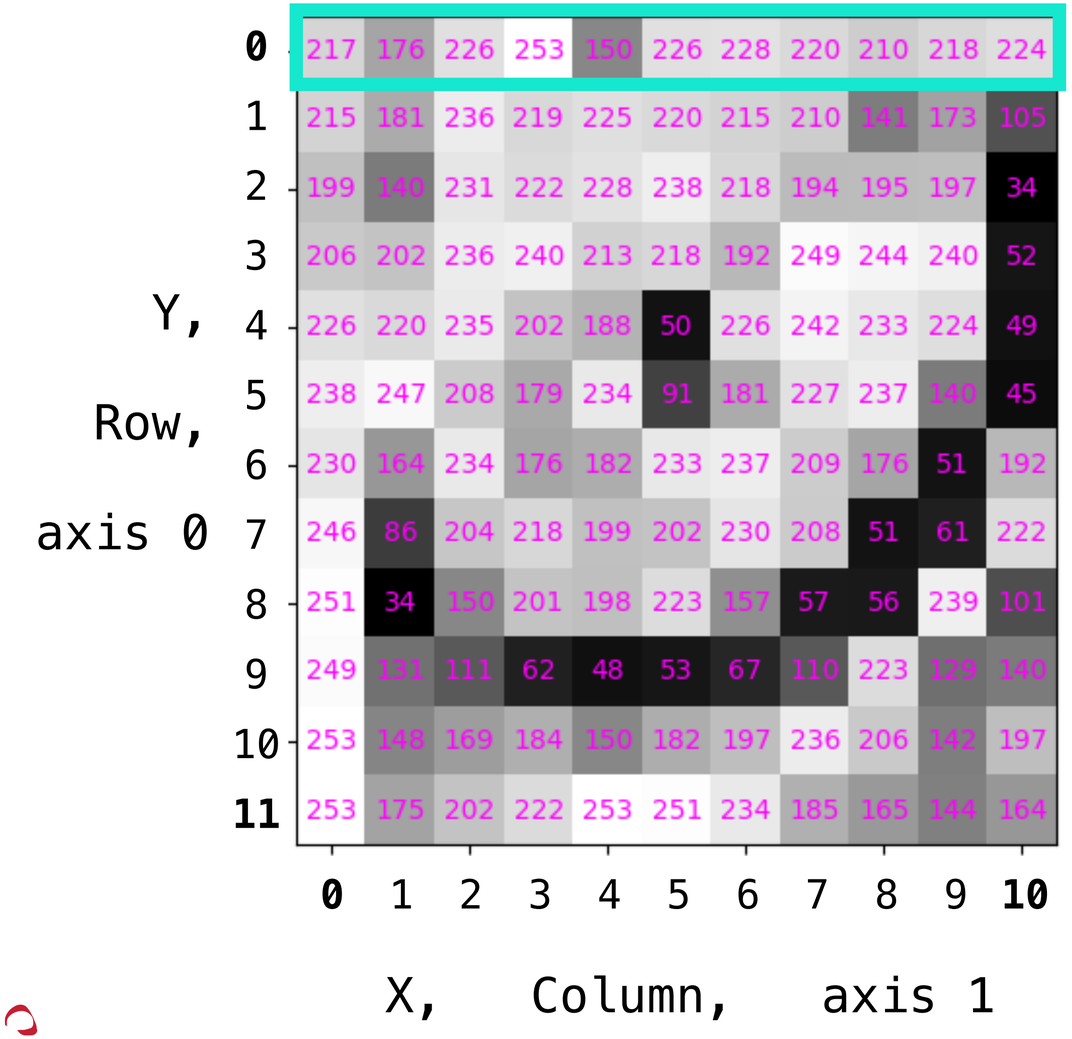


Exercise:

Explore indexing individual entries

```
row, col = 0, 10
valueplot(cat, indices=str([row, col]))
```

Indexing: rows

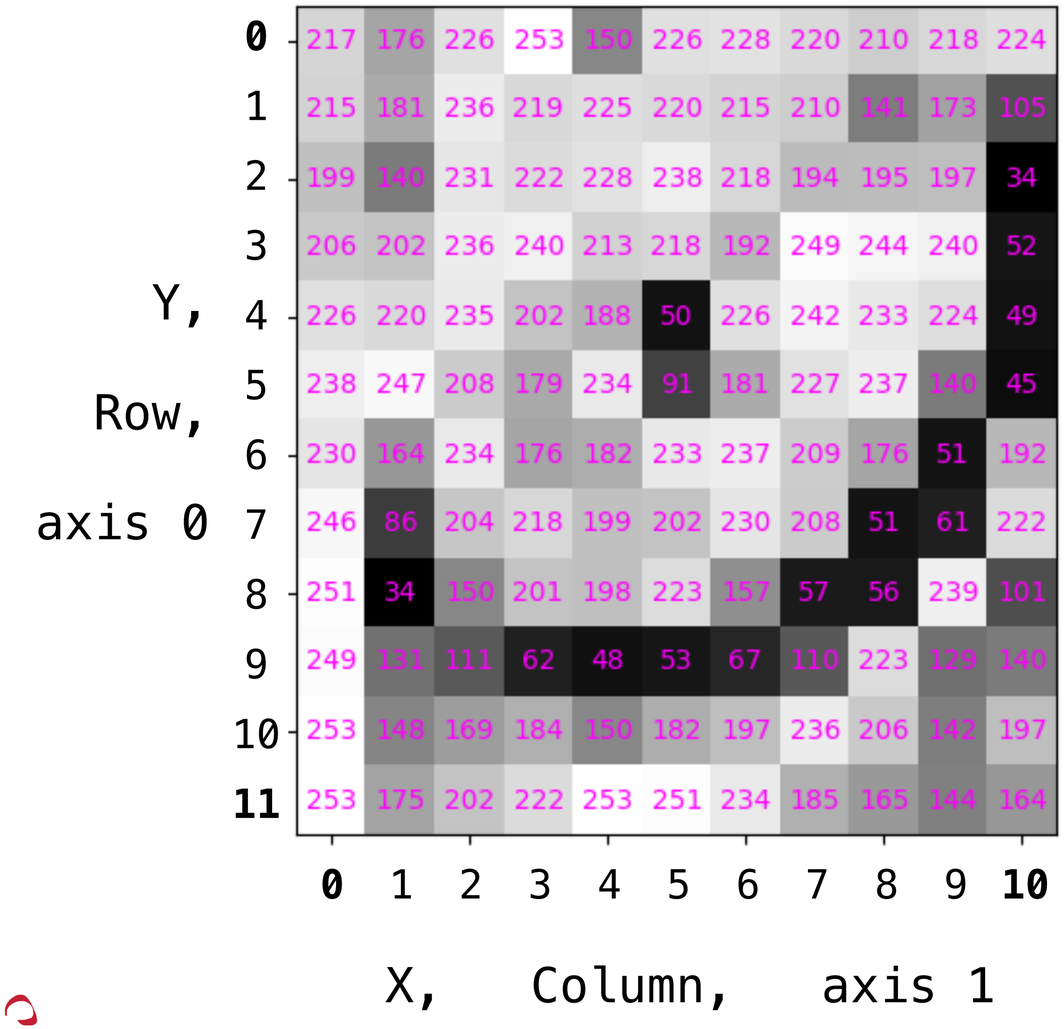


0th row all column

```
print(cat[0, :])
```

```
[217 176 226 253 150 226 228 220 210 218 224]
```

Indexing: rows



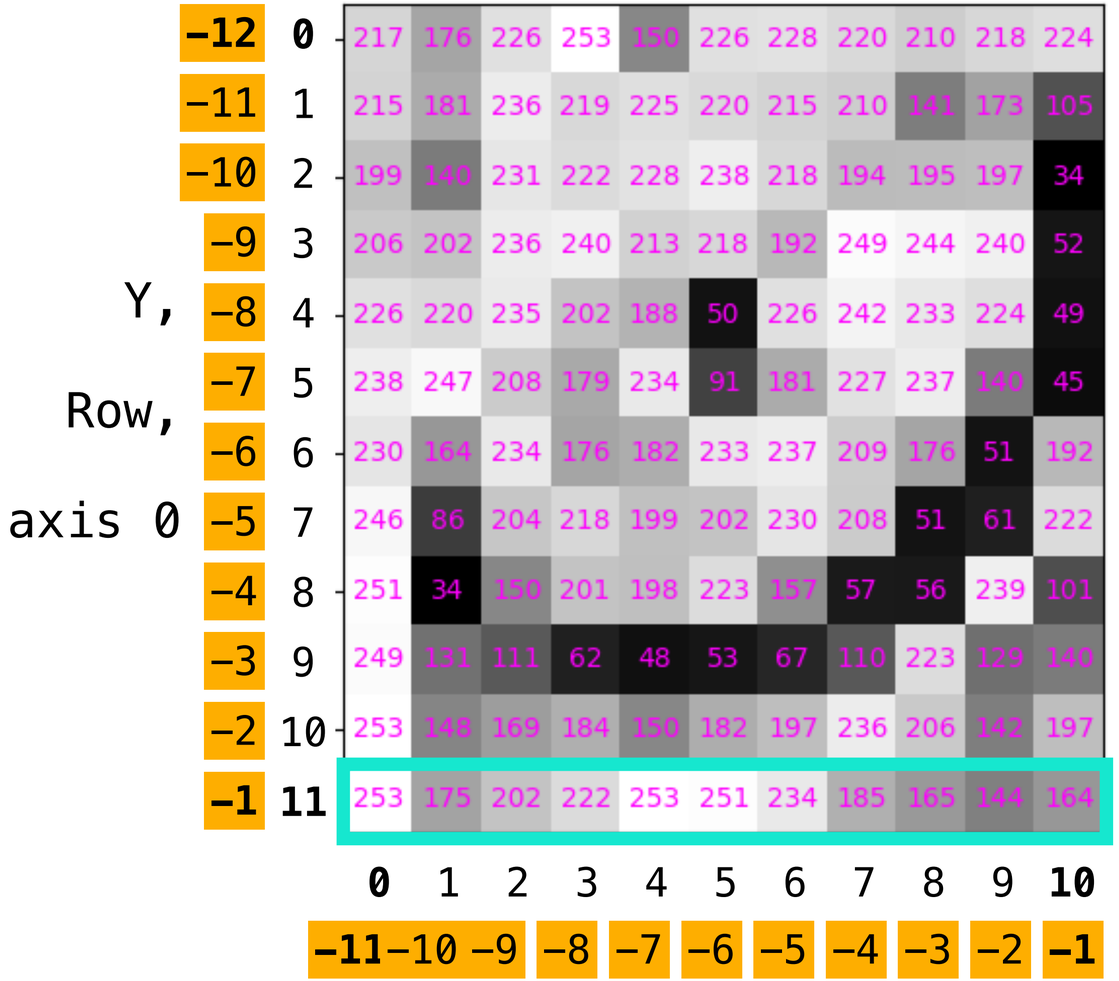

```
row = 1
print(cat[row, :])
print(cat[row,])
print(cat[row])
```

← axis 0

✓ 0.0s

```
[215 181 236 219 225 220 215 210 141 173 105]

[215 181 236 219 225 220 215 210 141 173 105]

[215 181 236 219 225 220 215 210 141 173 105]
```

Indexing: rows

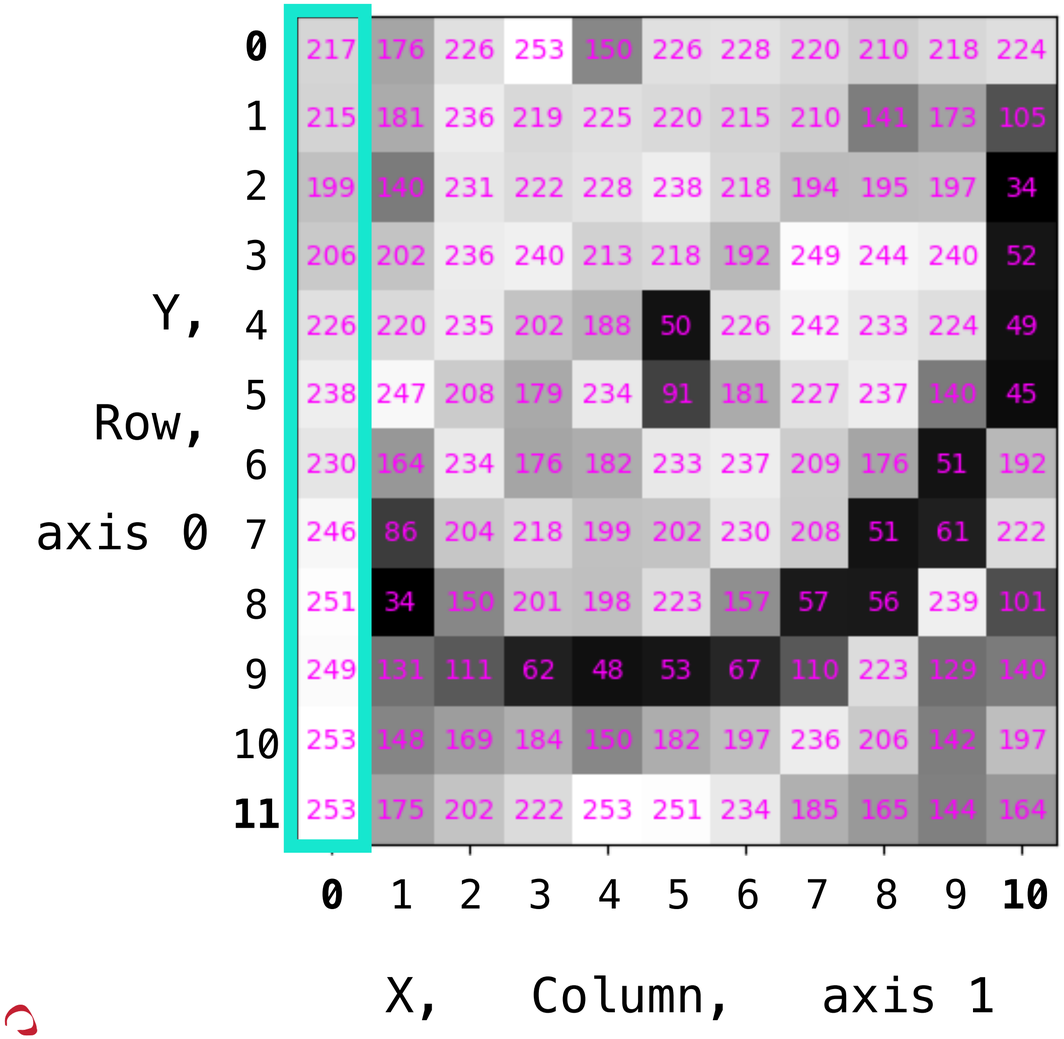



```
row = -1
print(cat[row, :])
print(cat[row,])
print(cat[row])
```

✓ 0.0s

```
[253 175 202 222 253 251 234 185 165 144 164]

[253 175 202 222 253 251 234 185 165 144 164]

[253 175 202 222 253 251 234 185 165 144 164]
```

Indexing: columns

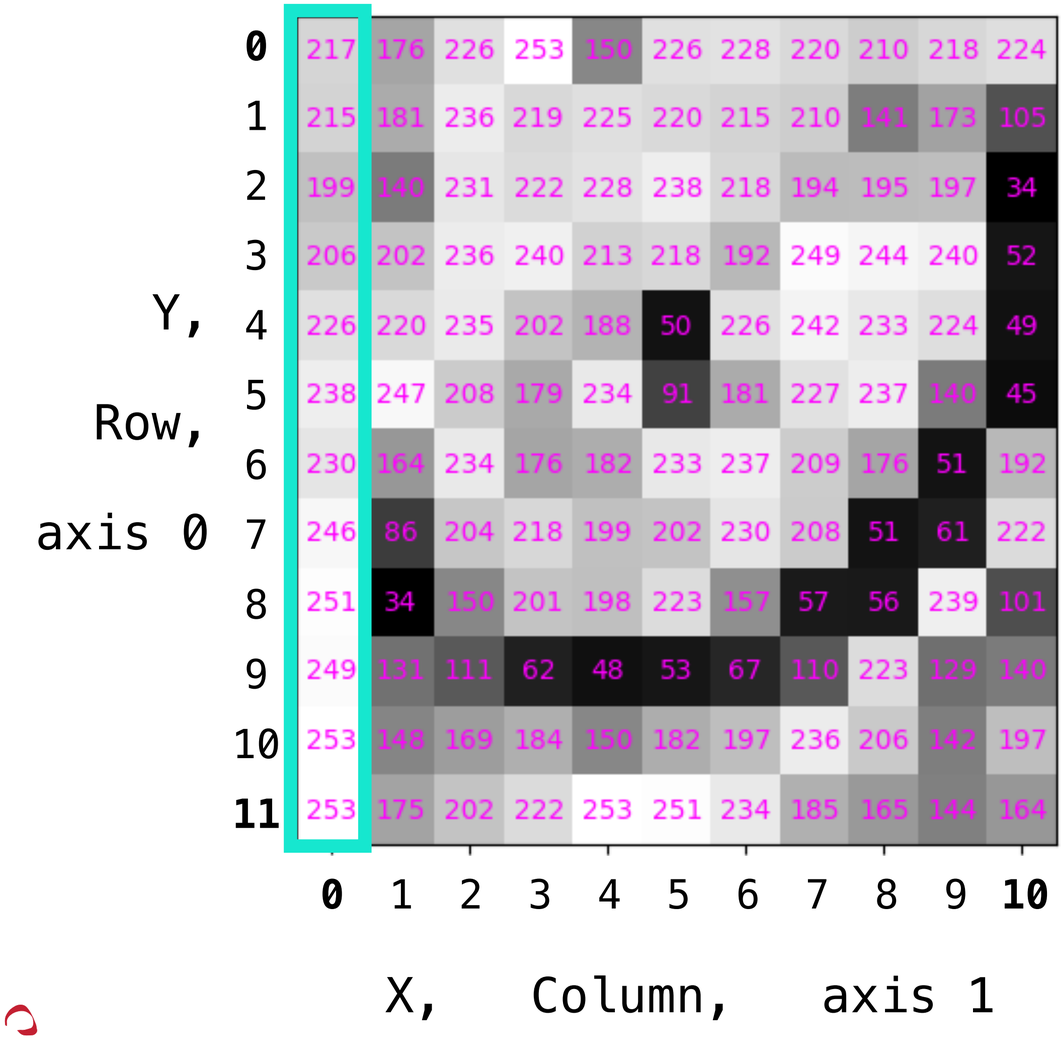


all rows column 0

```
print(cat[:, 0])
```

```
[217 215 199 206 226 238 230 246 251 249 253 253]
```

Indexing: columns

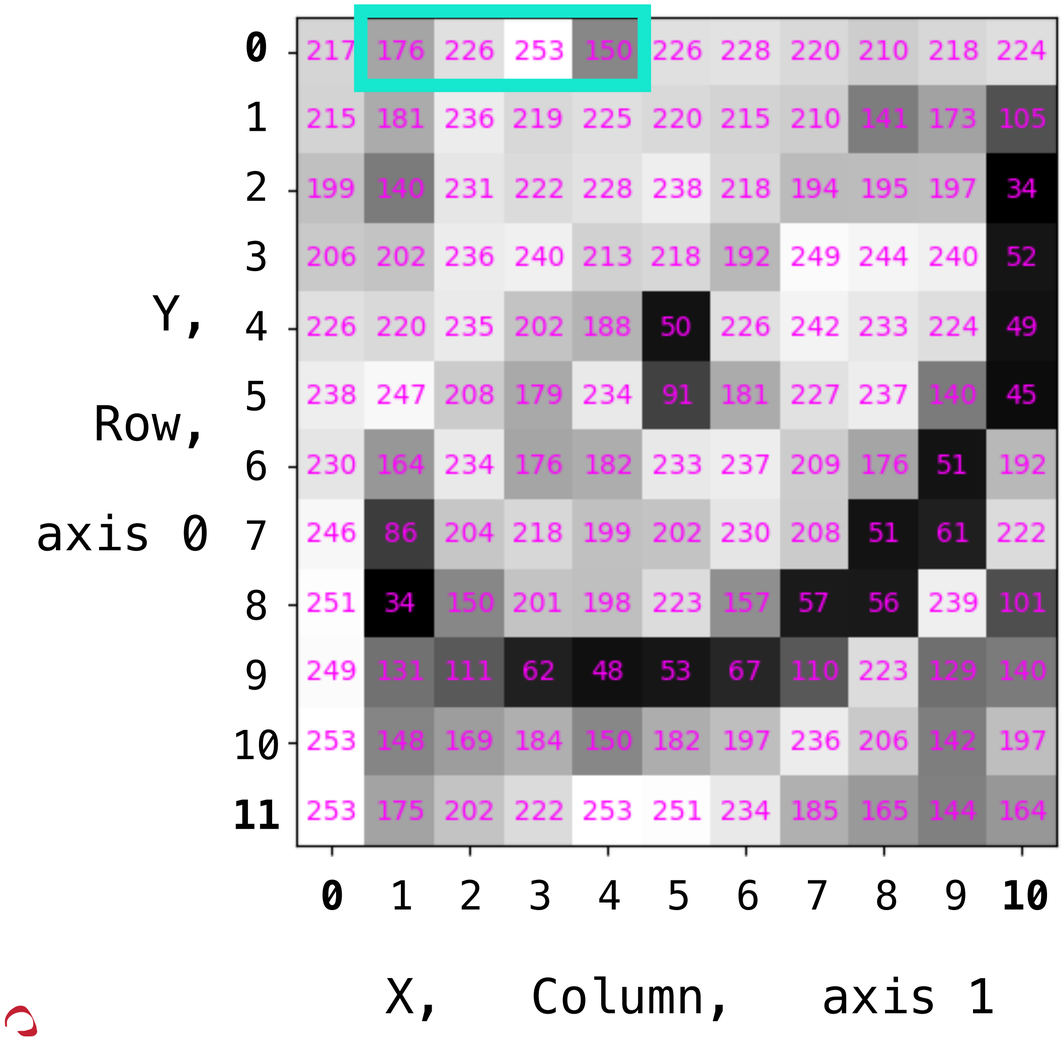


axis 0 axis 1

```
print(cat[:, 0])
```

-> Exercises

Indexing: rows and columns

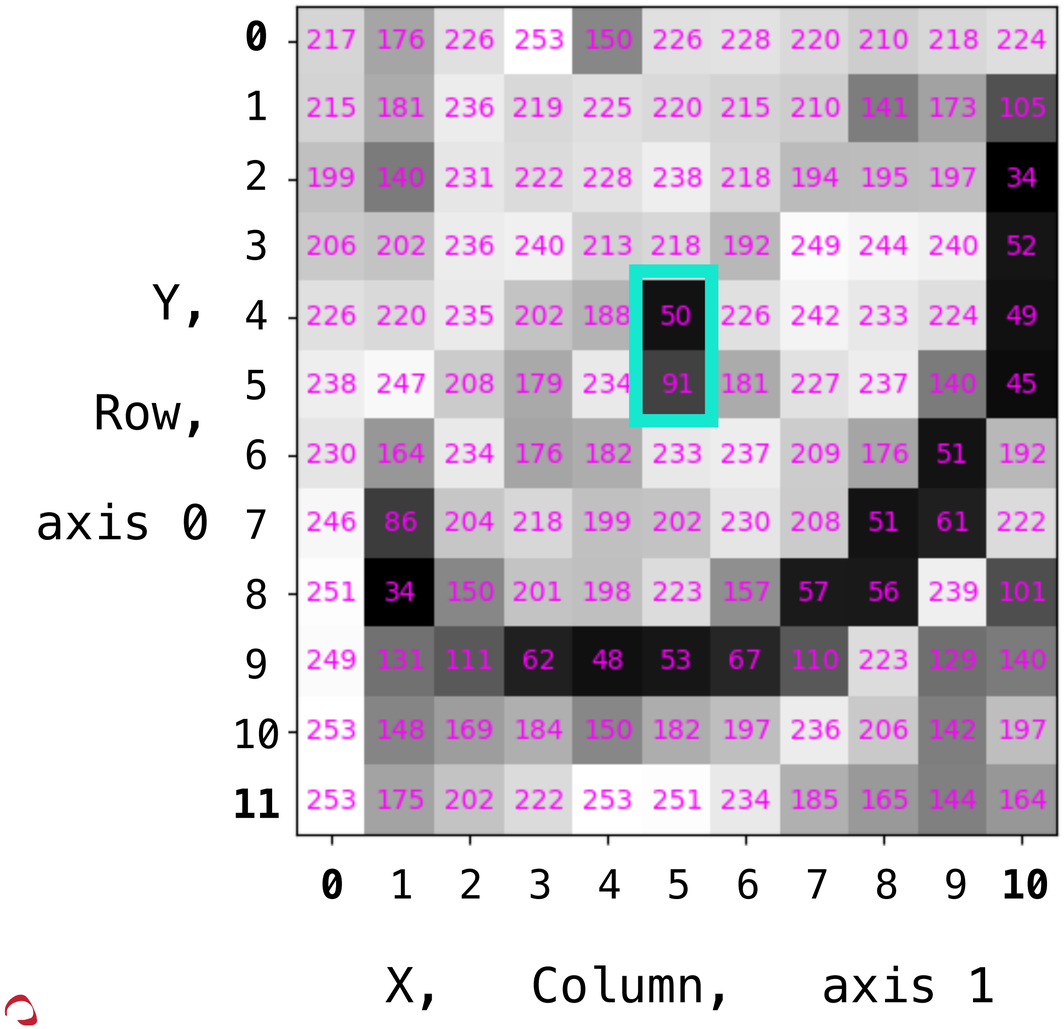


"Column 1 (inclusive) to 5 (exclusive)"

```
print(cat[0, 1:5])
```

```
[176 226 253 150]
```

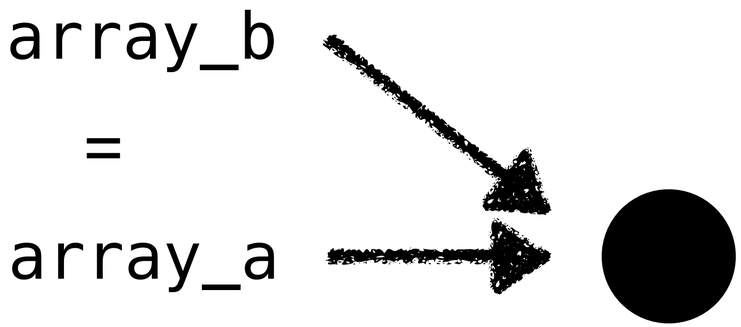
Indexing: rows and columns



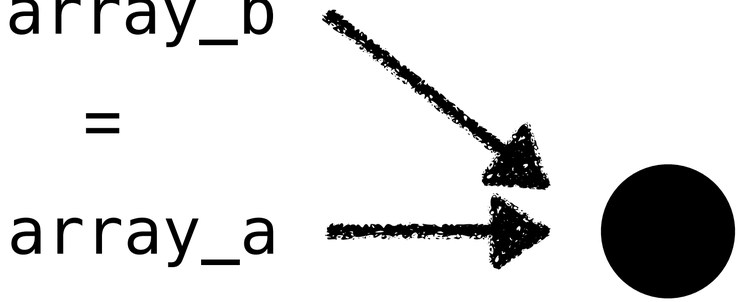
Exercise:

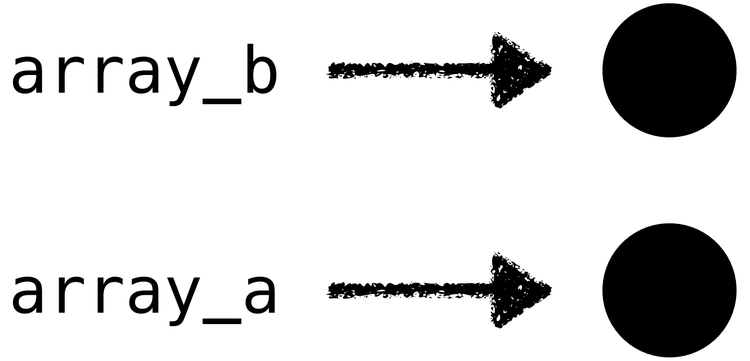
highlight these values using function `valueplot()`


array_a




```
array_b = array_a
```

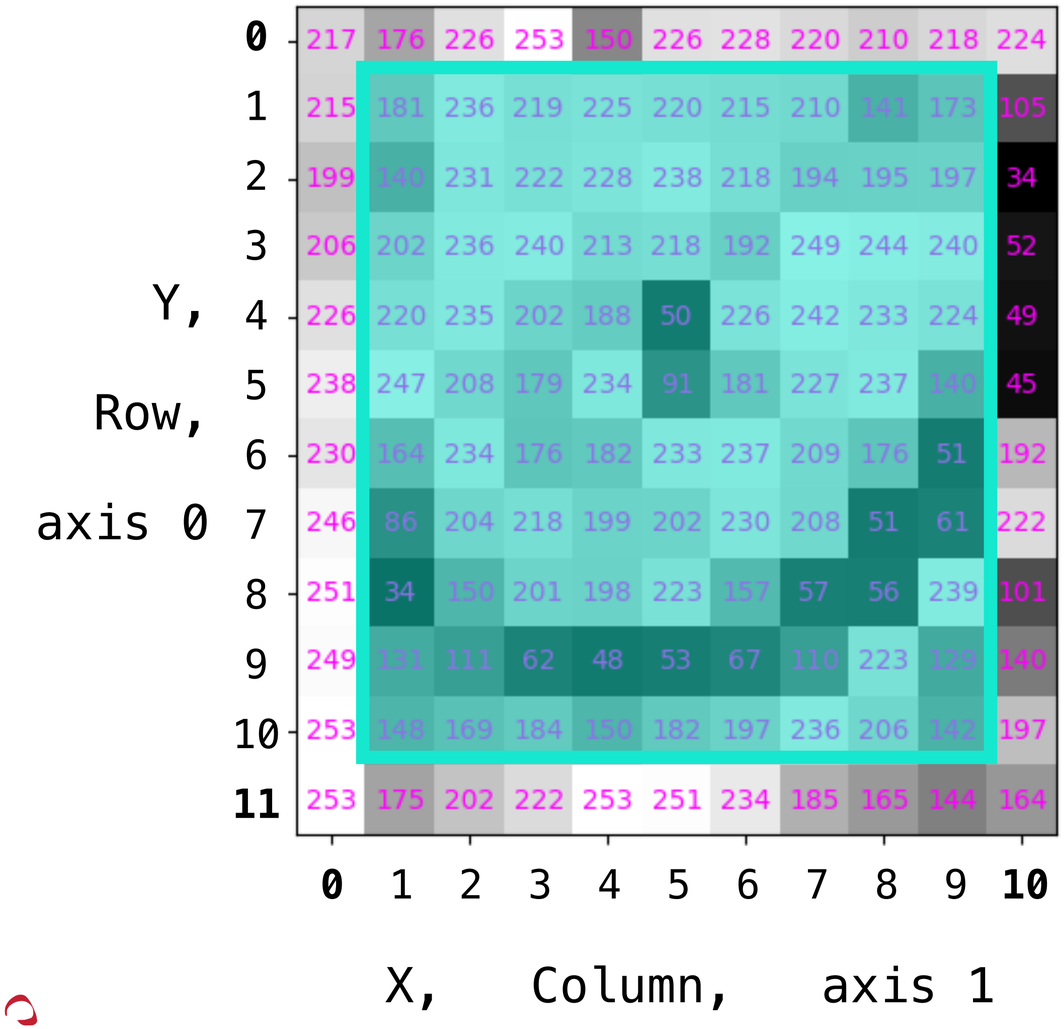





```
array_b = array_a
```

```
array_b = array_a.copy()
```

Indexing: rows and columns



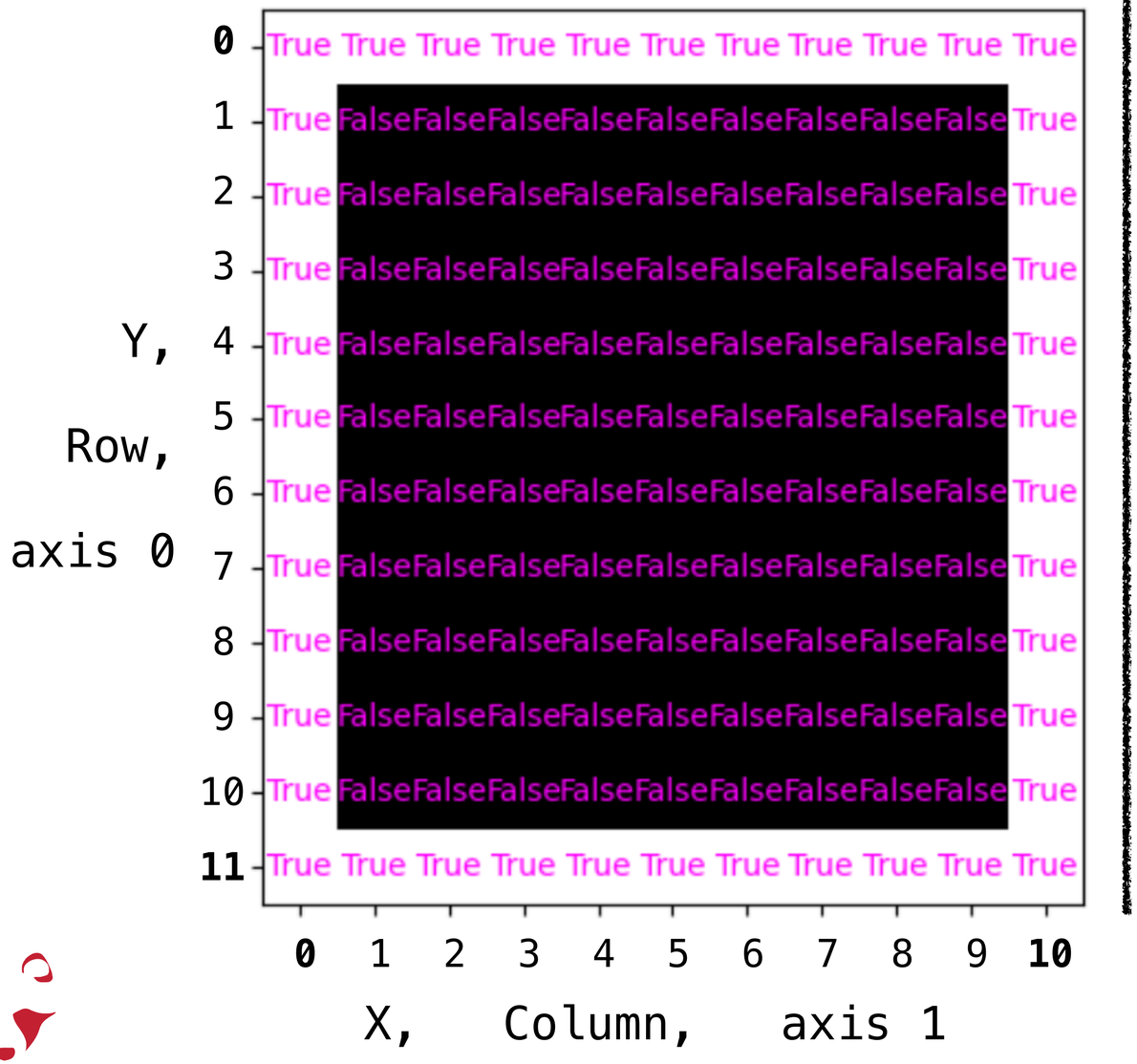
Exercise:

Make a copy of cat and name it `"pirate"`.

Assign to all pixels but the rim-pixels a value of 0.

Plot to verify

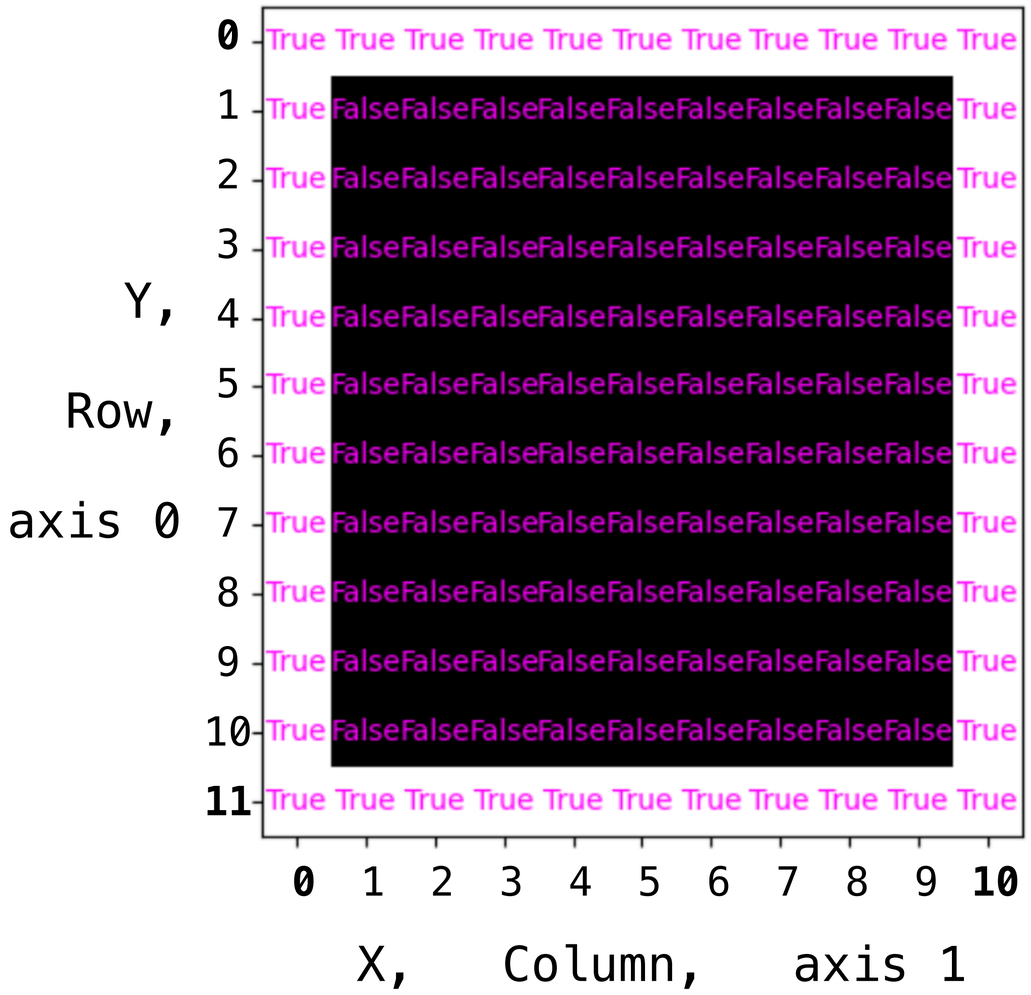
Indexing: Boolean



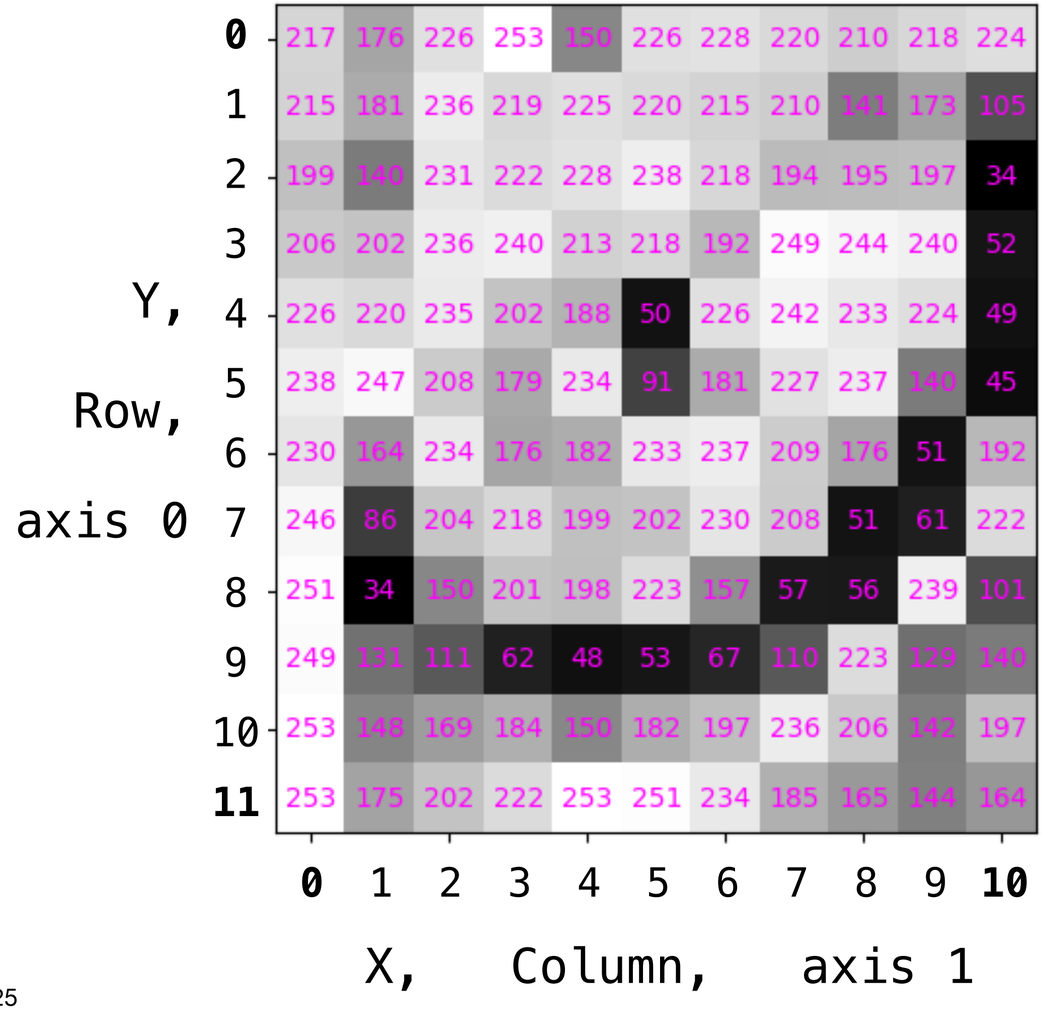
numpy arrays can contain boolean entries

Indexing: Boolean

monocle_bool

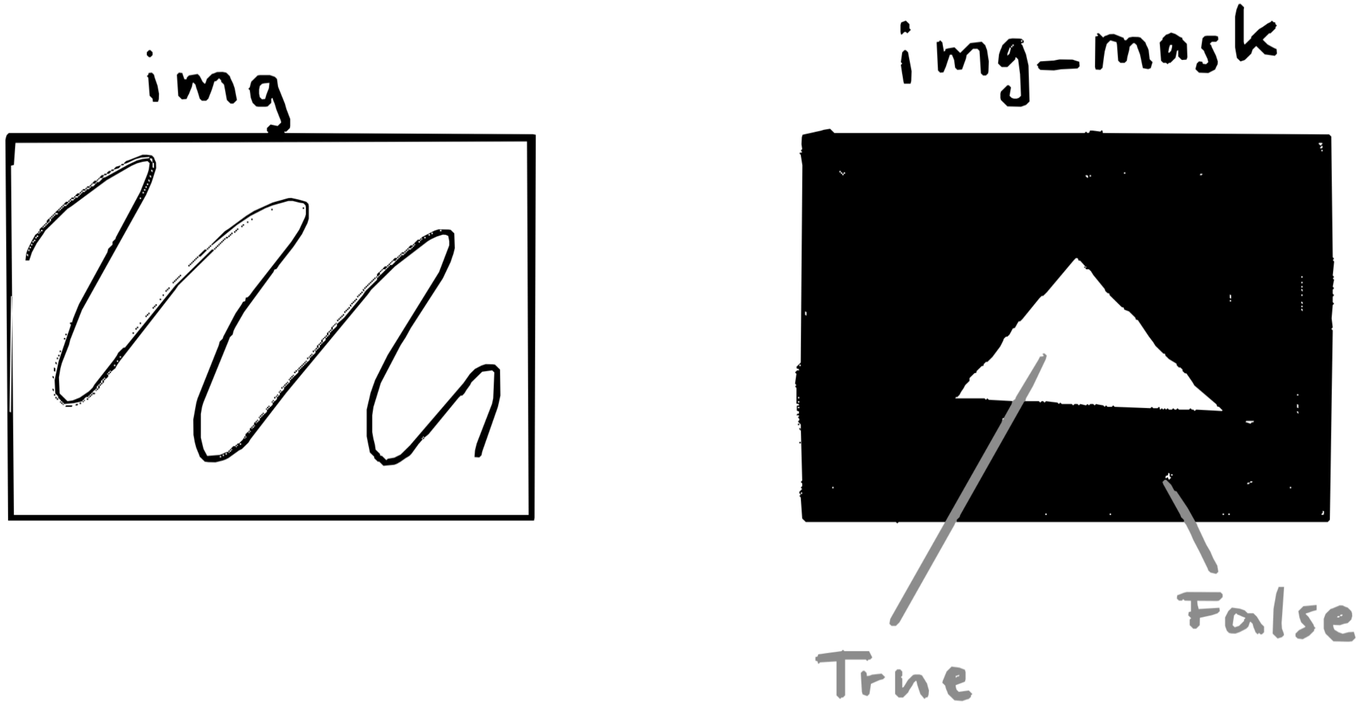


cat

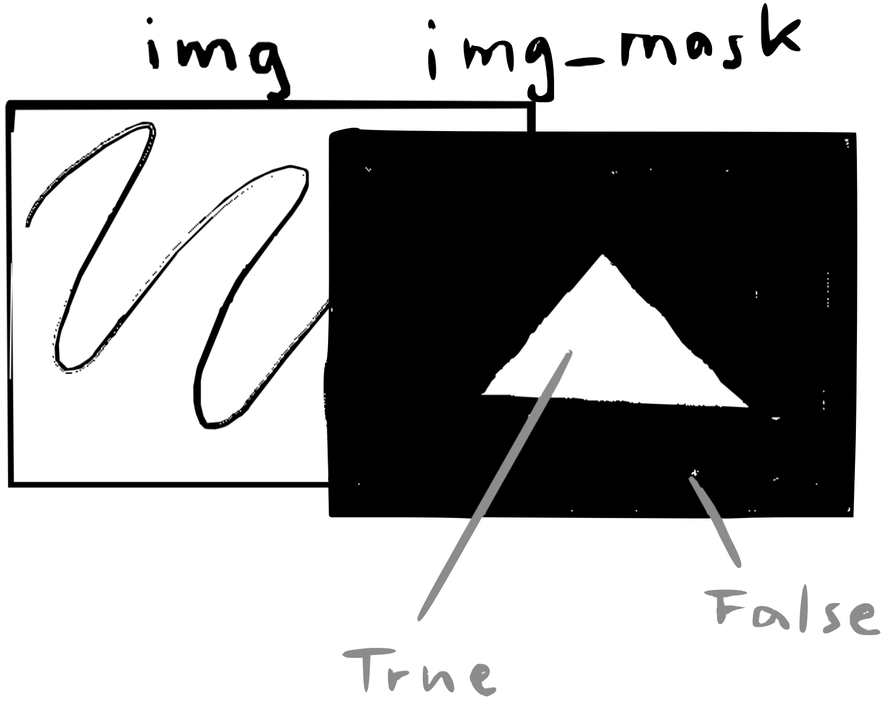


Indexing: Boolean

Indexing: Boolean

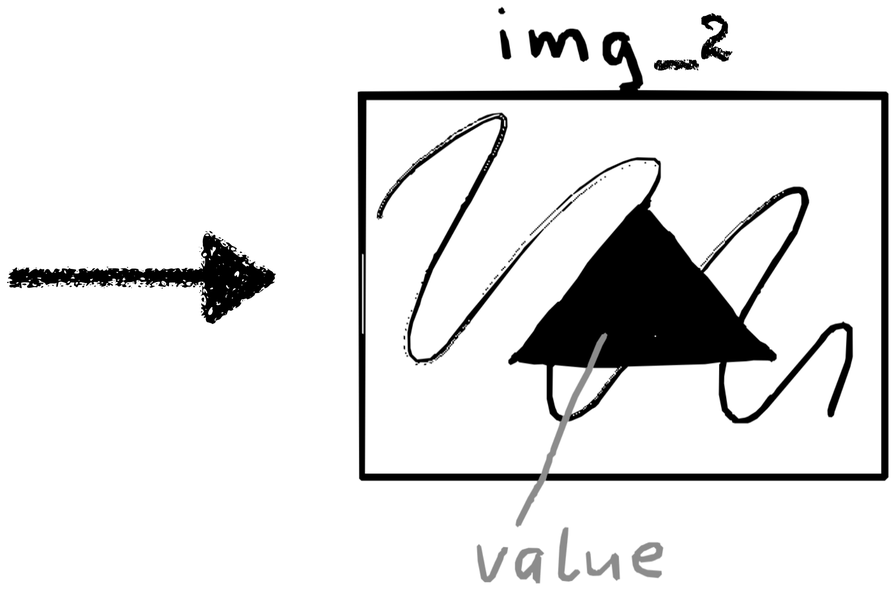


Indexing: Boolean

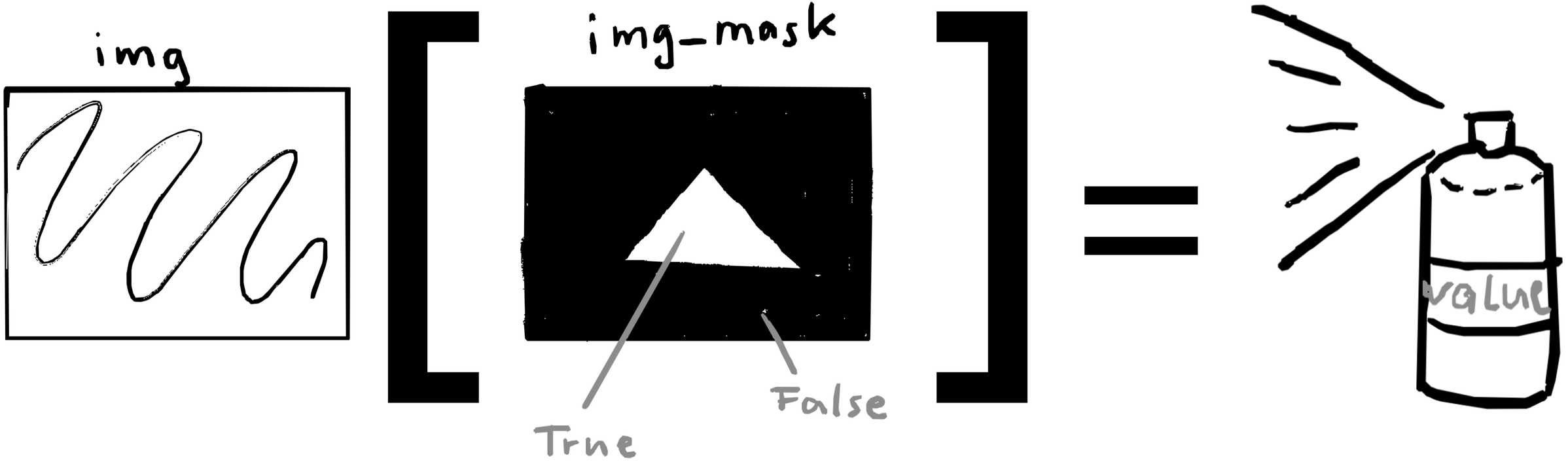


Indexing: Boolean

Indexing: Boolean

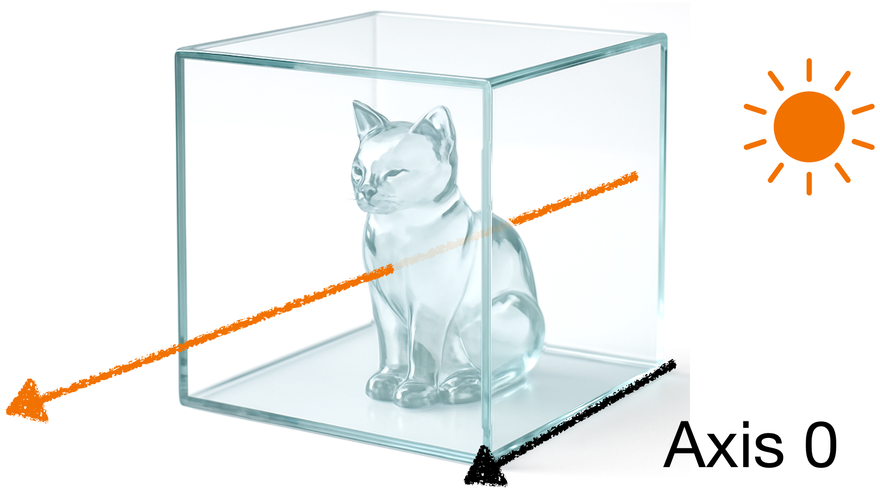


Indexing: Boolean

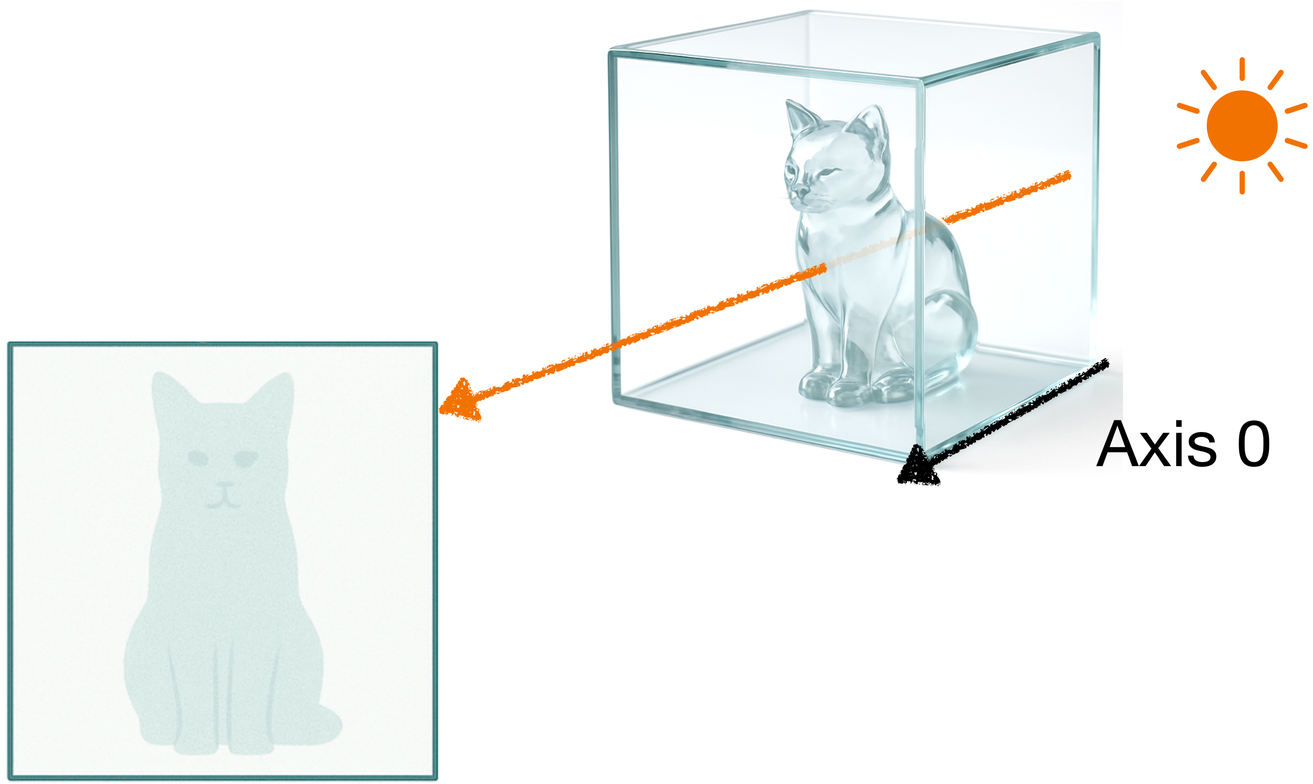


Projections

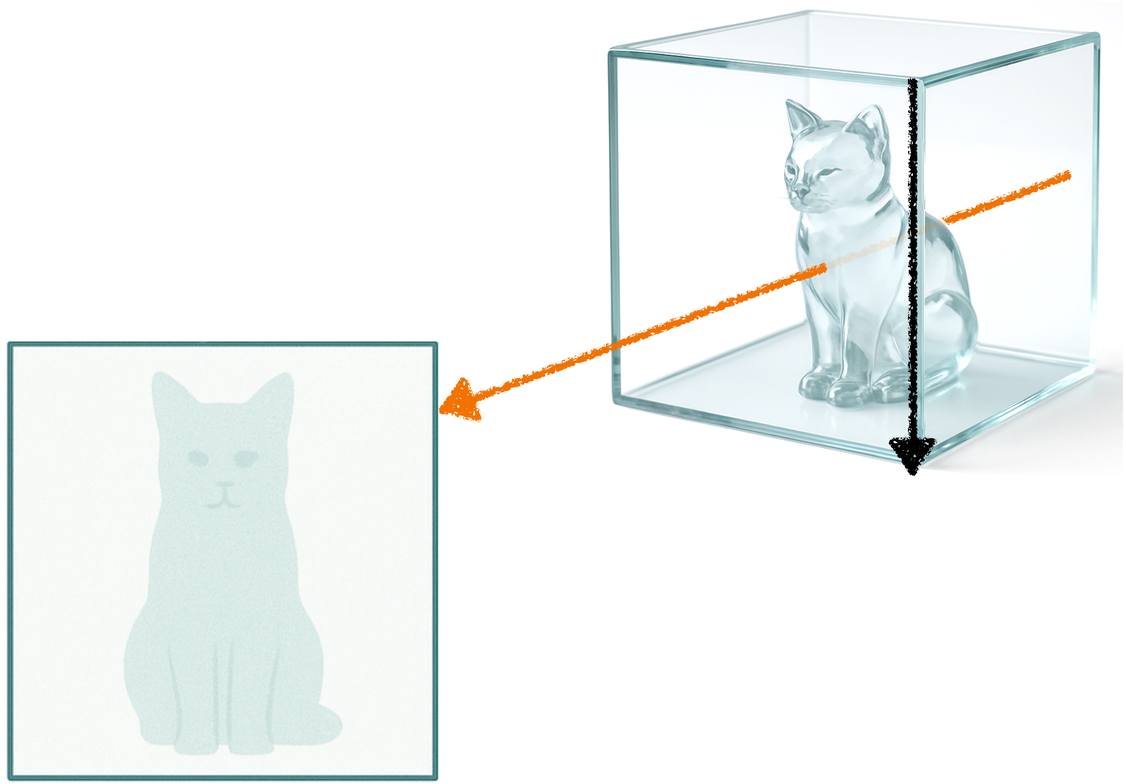
Projections



Projections

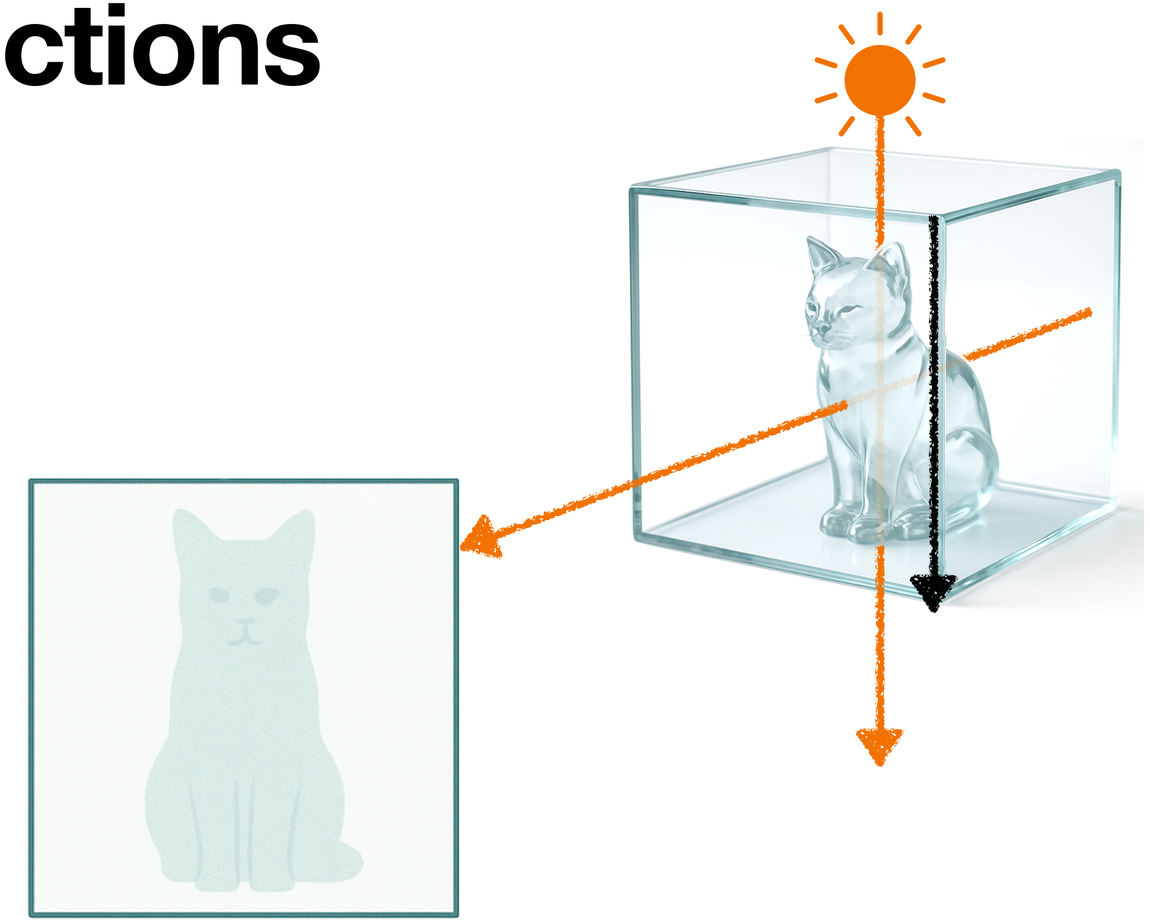


Projections



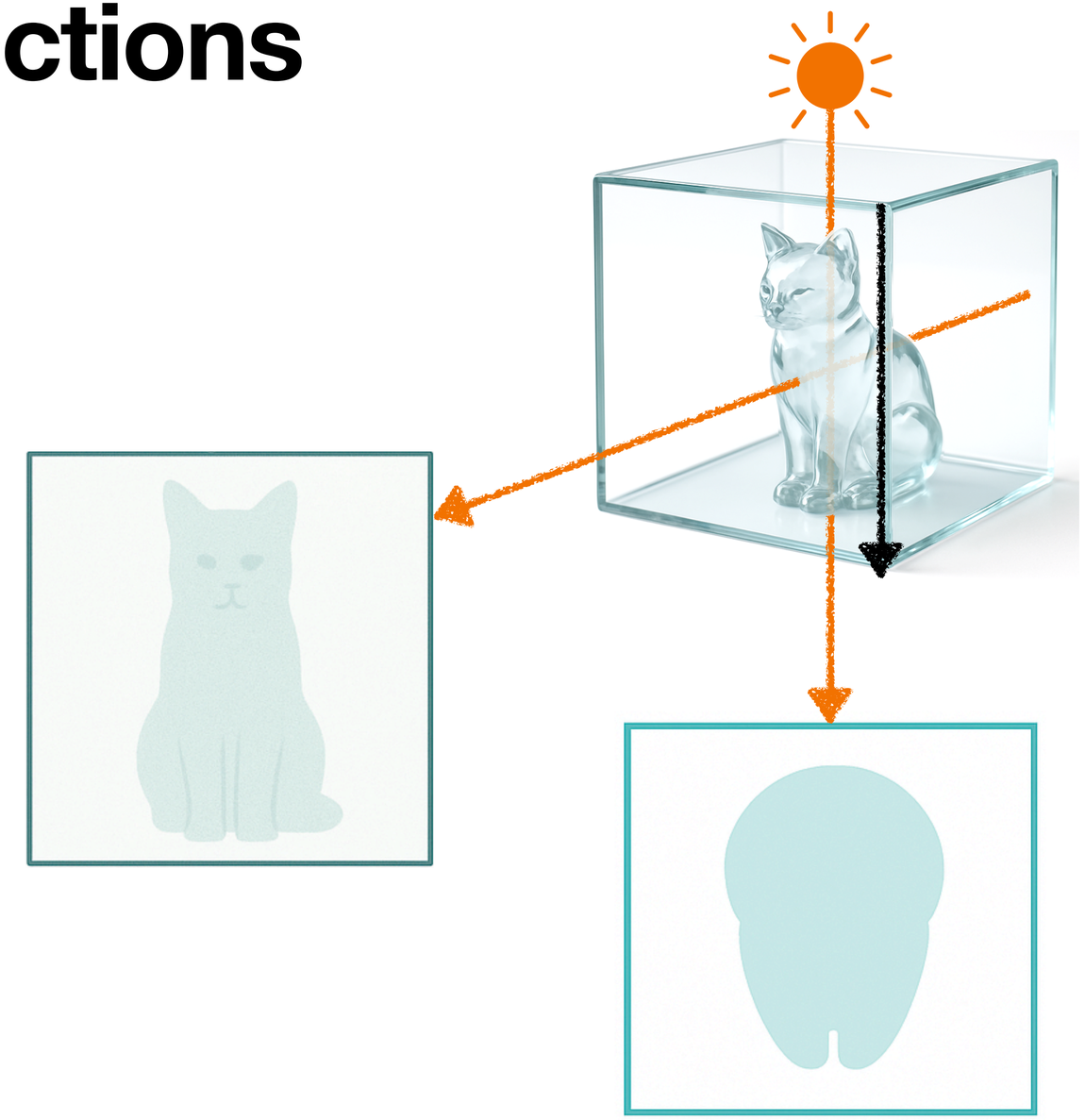
Axis 1

Projections



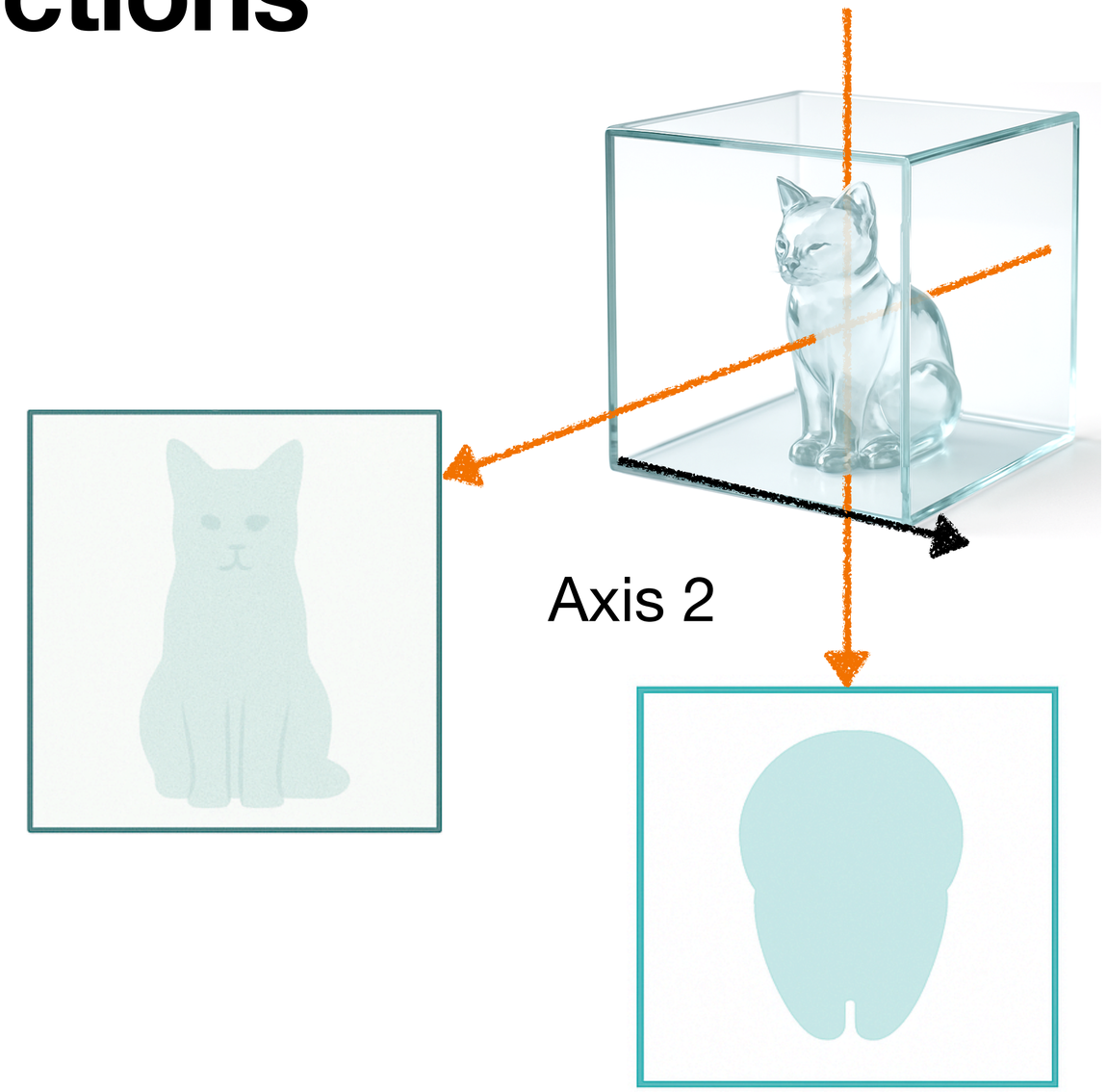
Axis 1

Projections

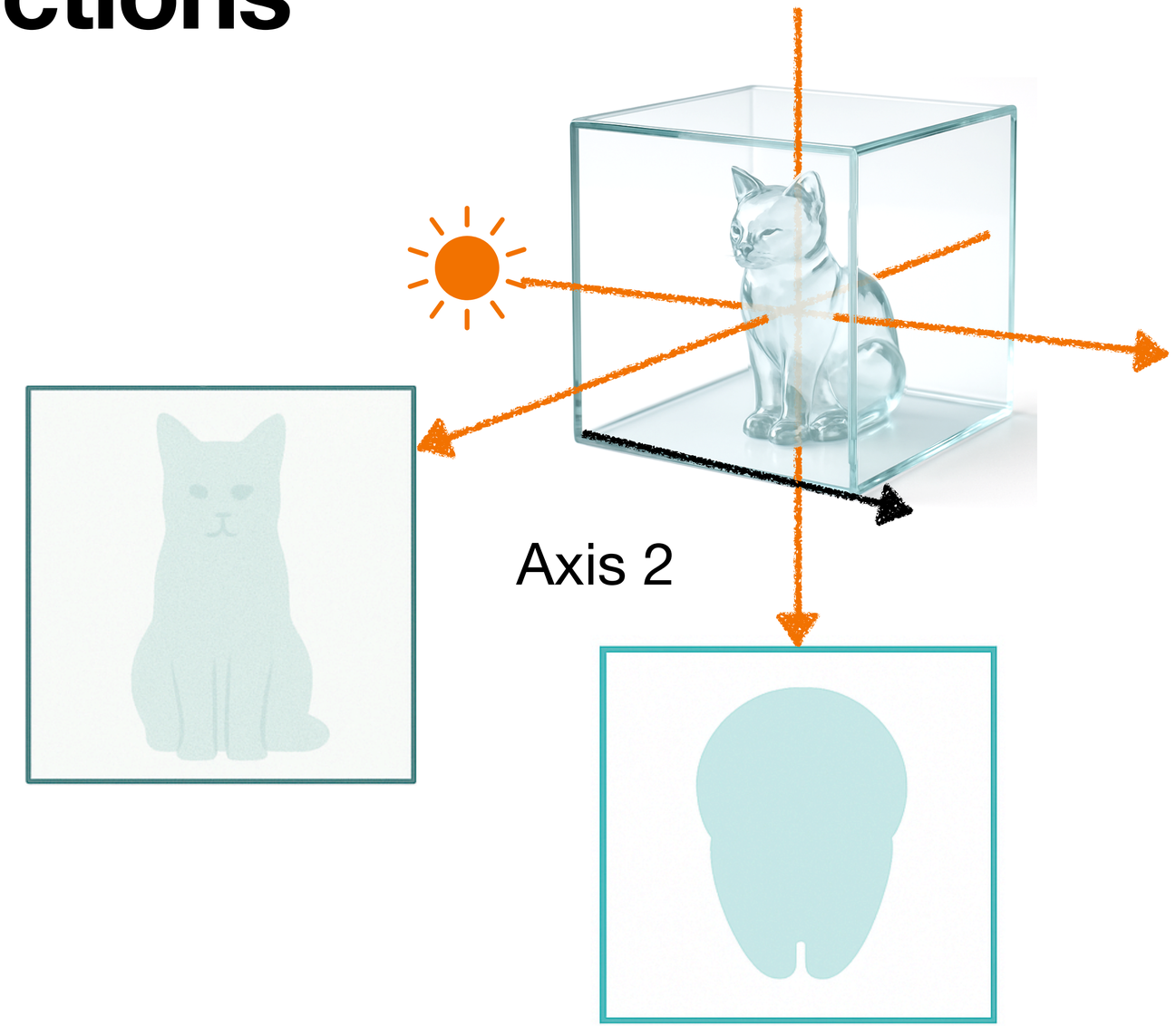


Axis 1

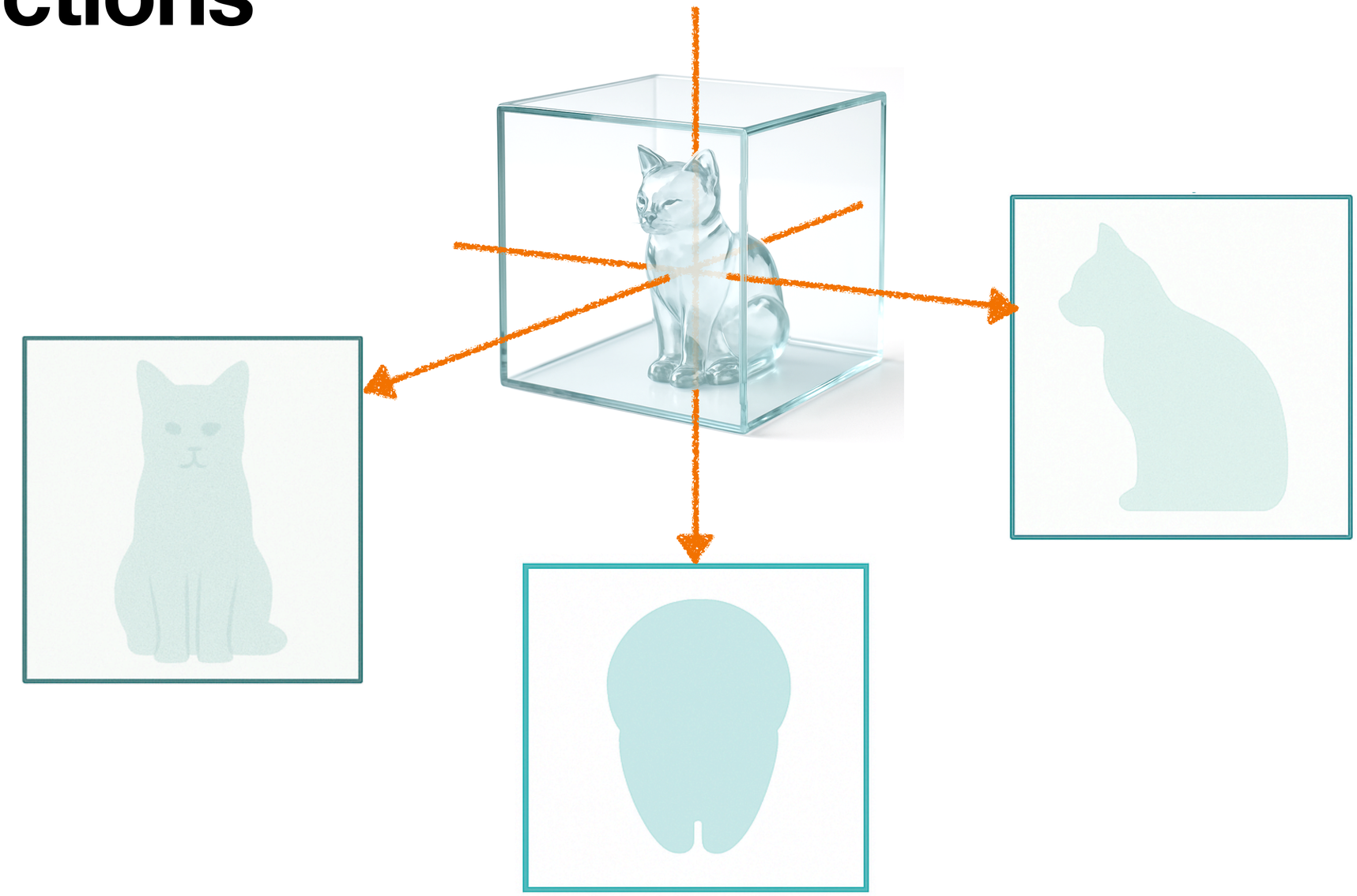
Projections



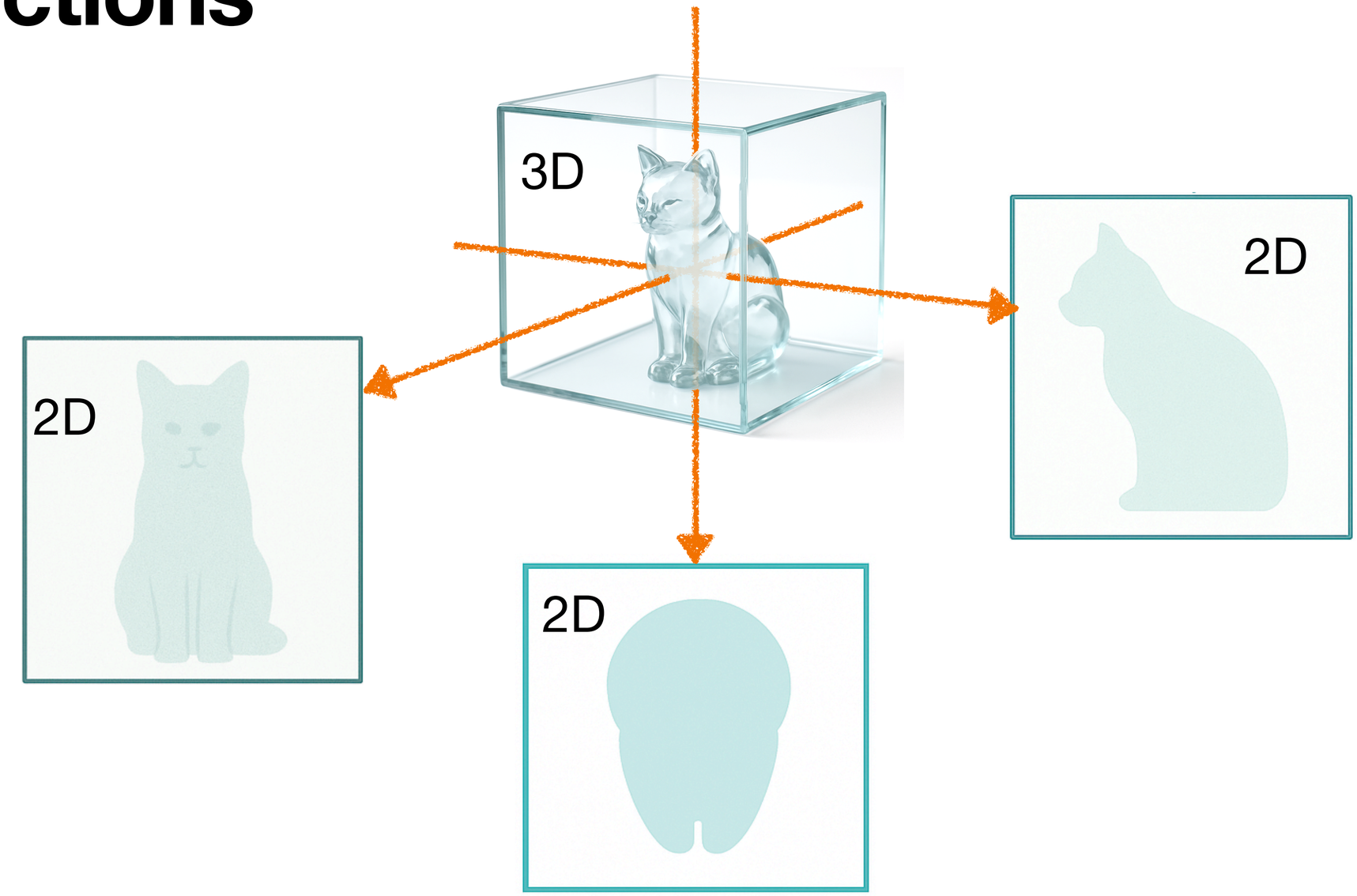
Projections

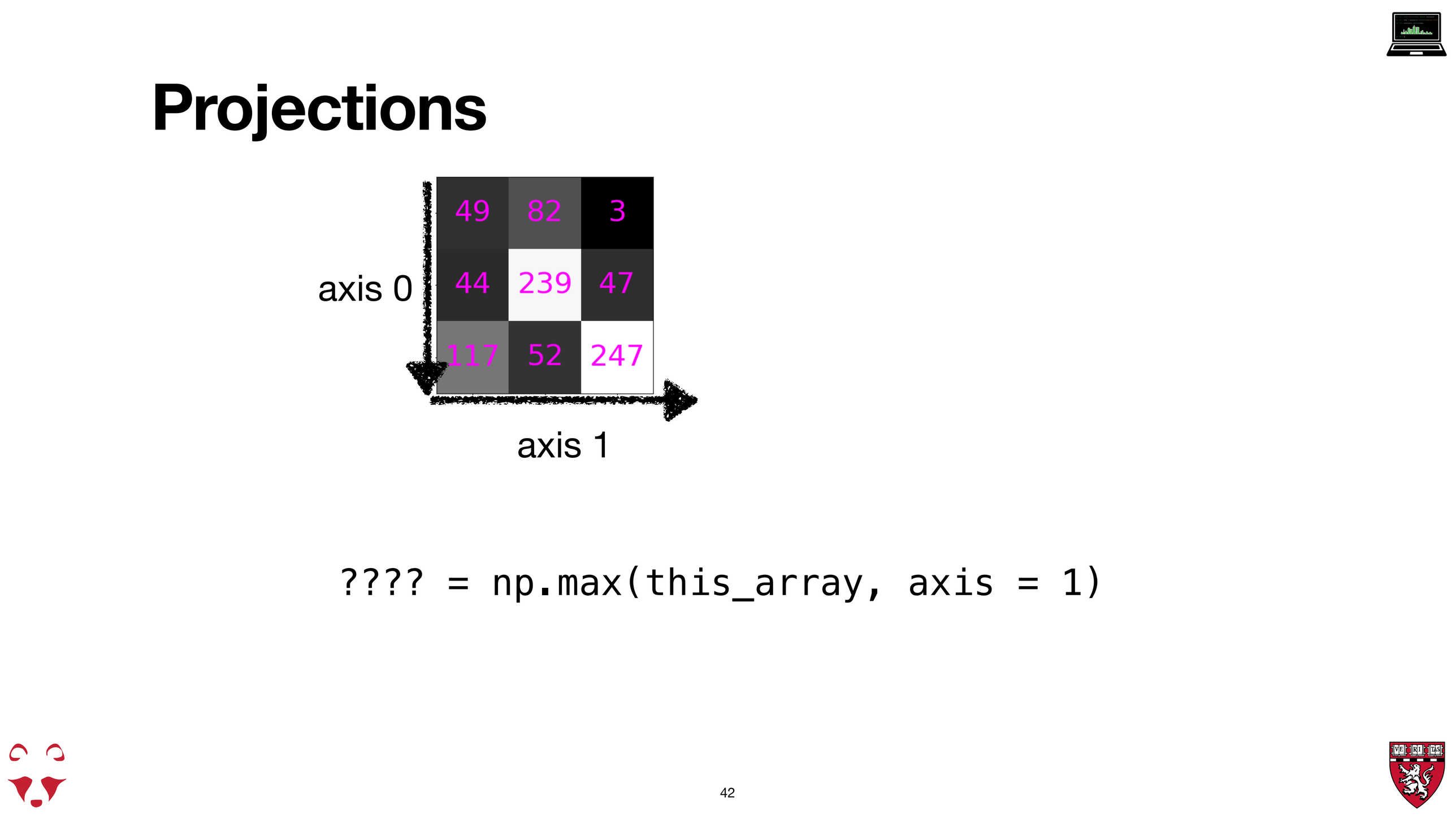


Projections



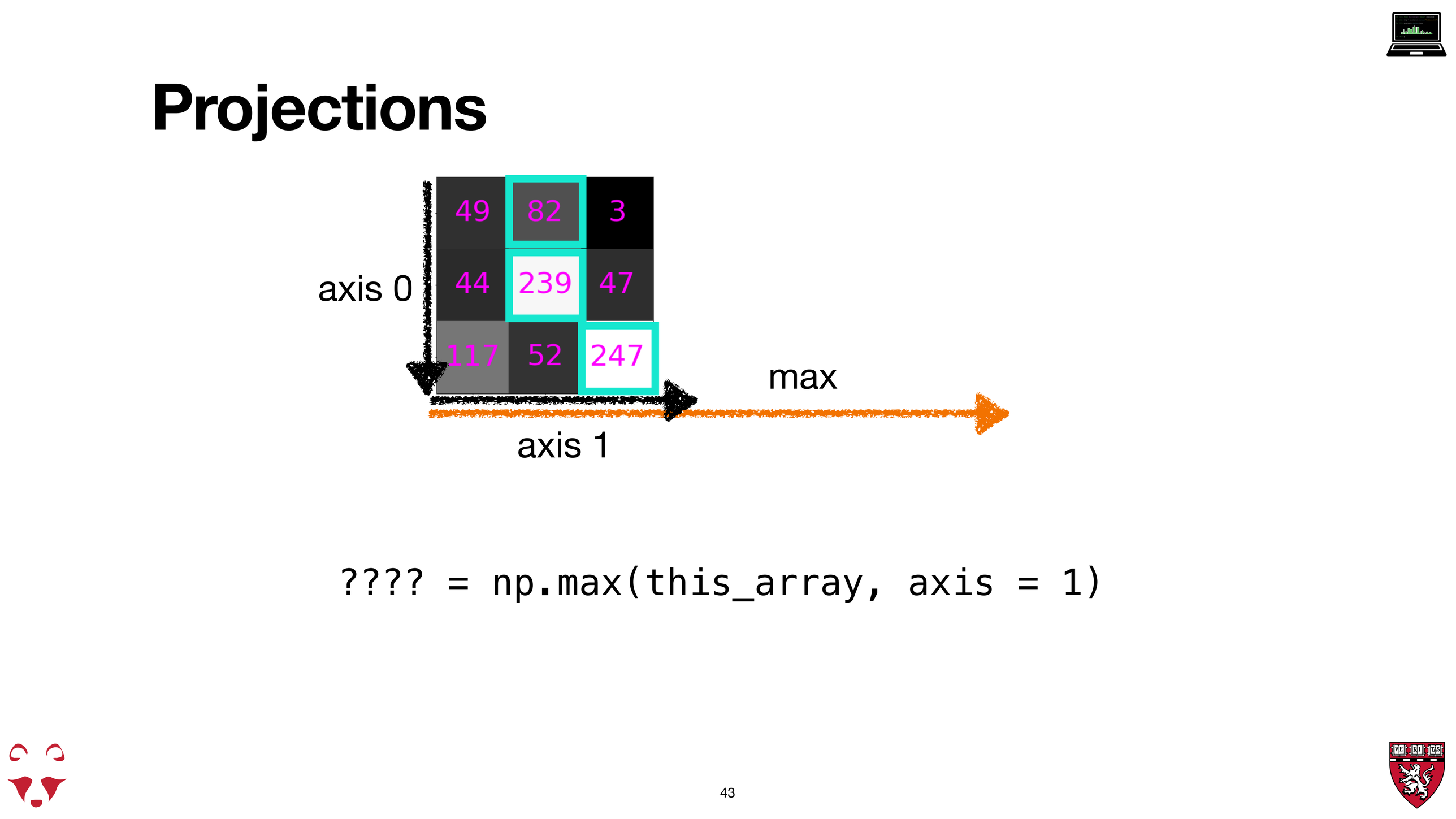
Projections





Projections

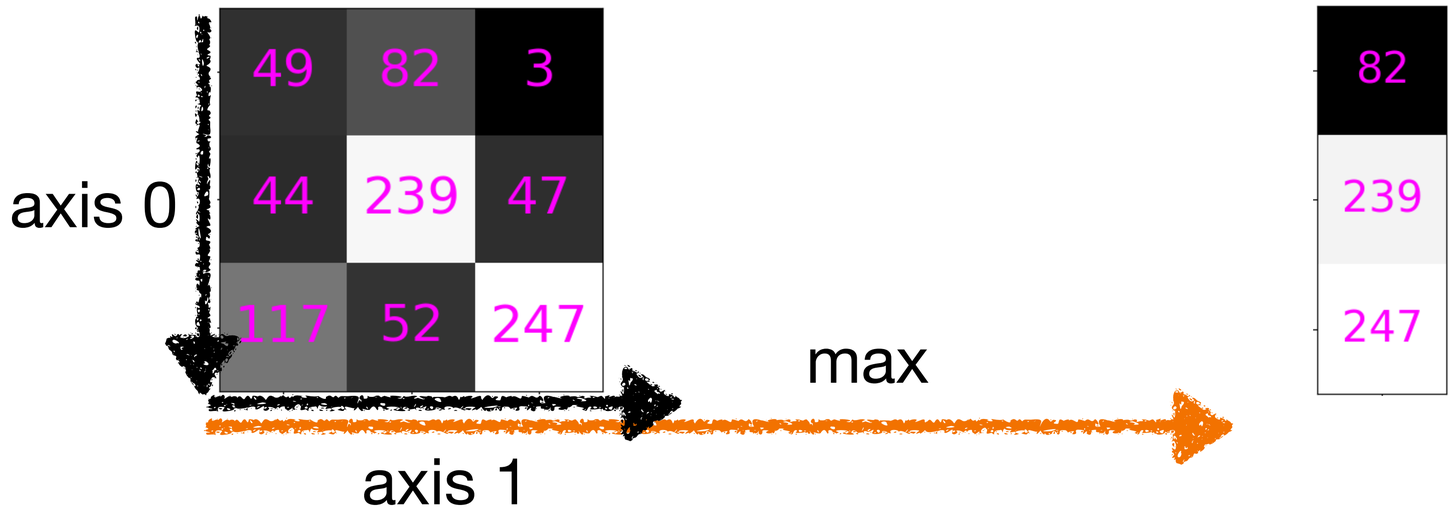

```
???? = np.max(this_array, axis = 1)
```



Projections

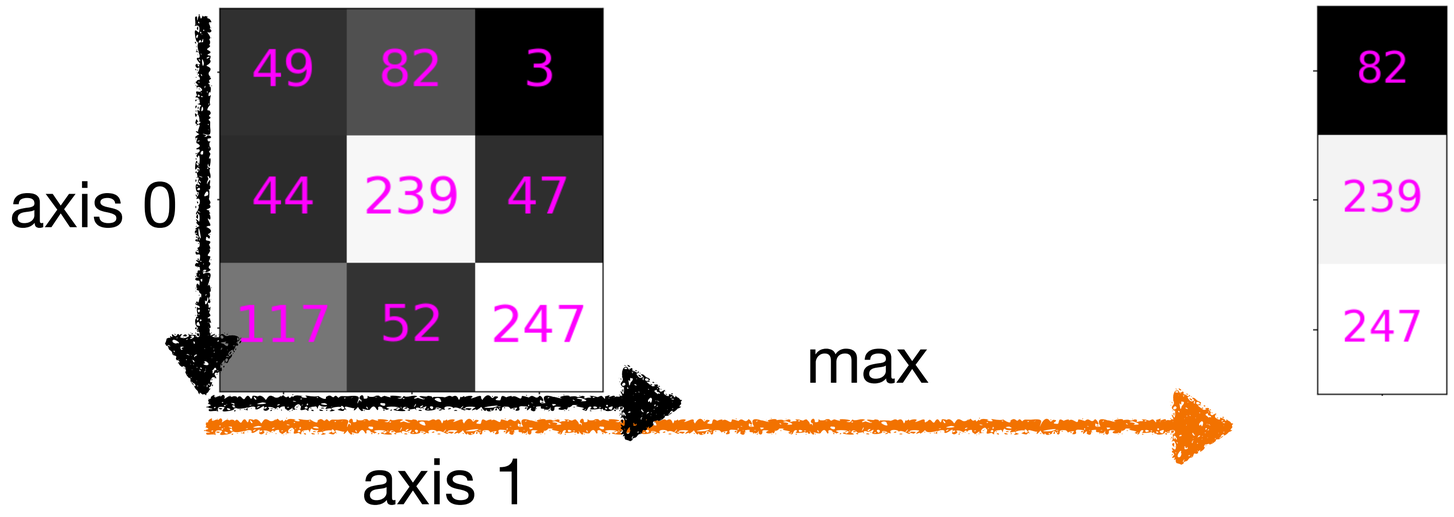

```
???? = np.max(this_array, axis = 1)
```

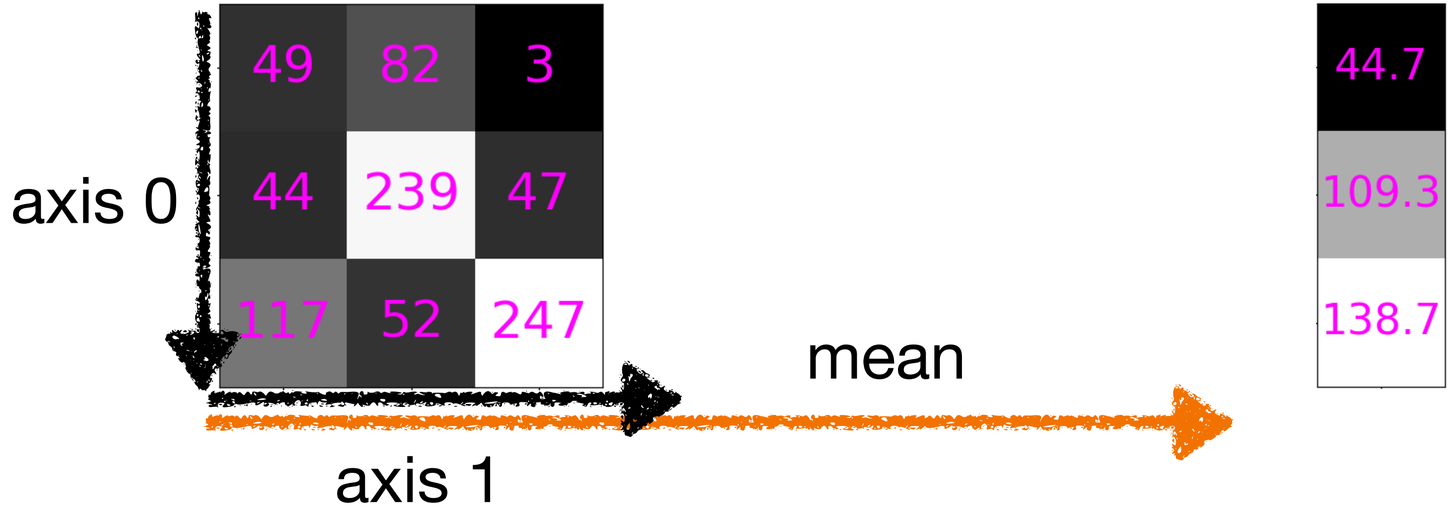
Projections



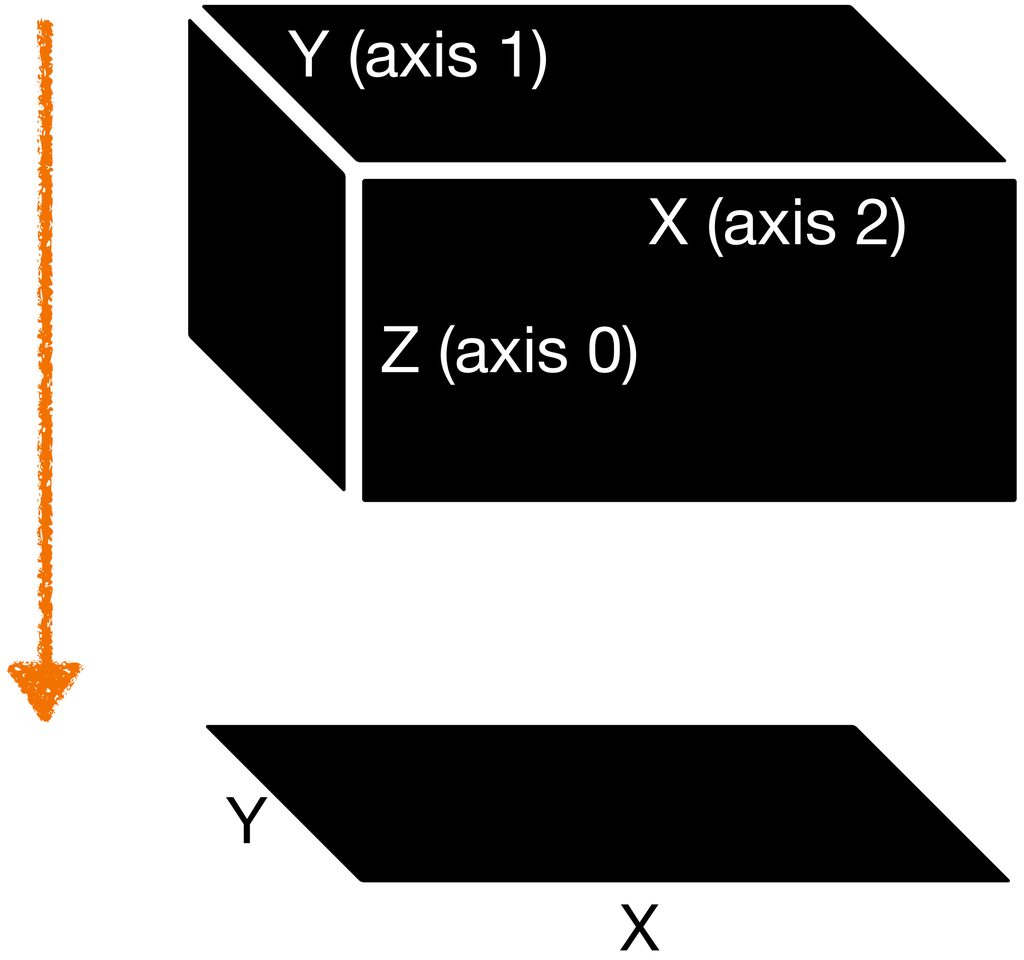

```
???? = np.max(this_array, axis = 1)
```

Projections





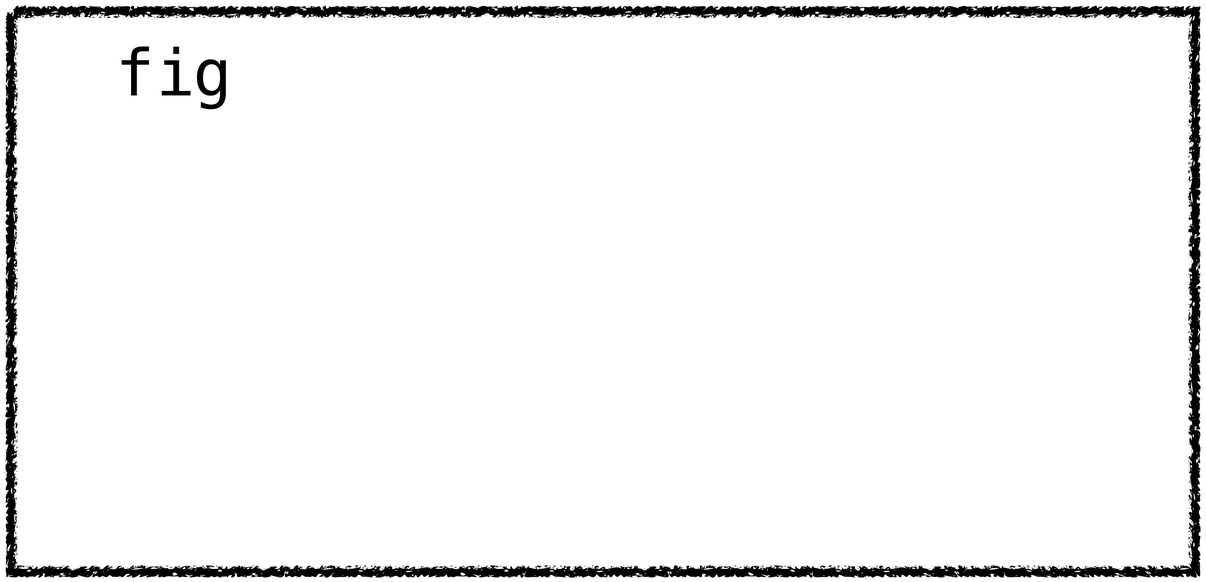
Projections



`ch0`

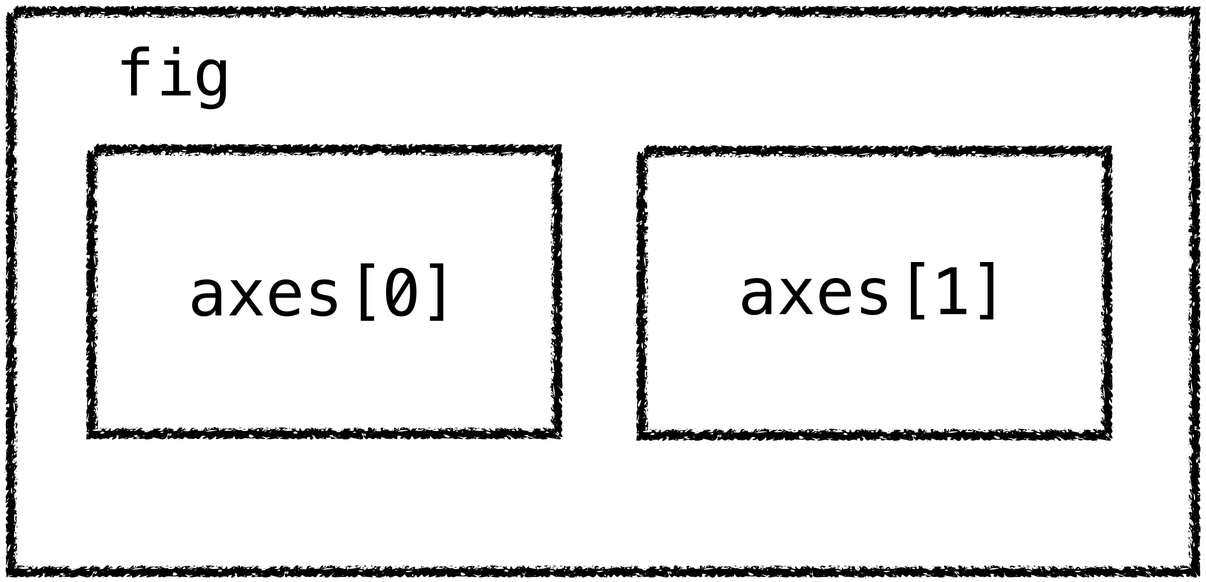
Matplotlib: Plotting

```
fig, axes = plt.subplots(1, 2)
```



Matplotlib: Plotting

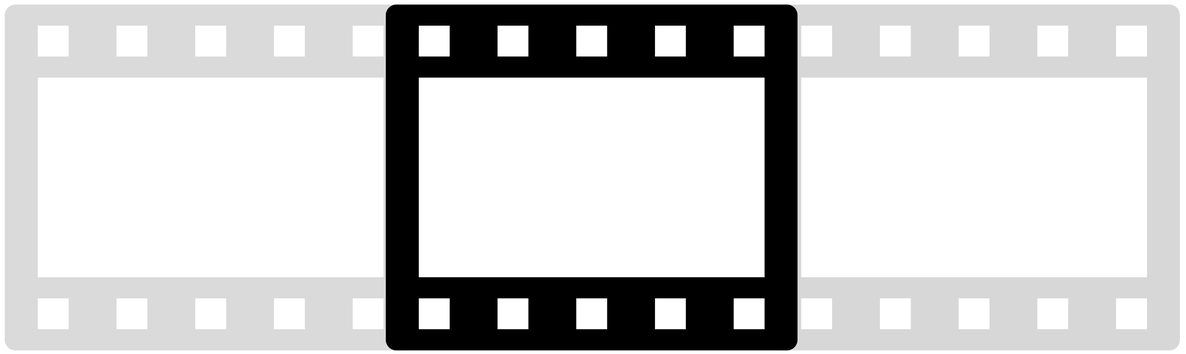
```
fig, axes = plt.subplots(1, 2)
```



Axes of length one

```
print(img.dims)
print(stack.shape)
```

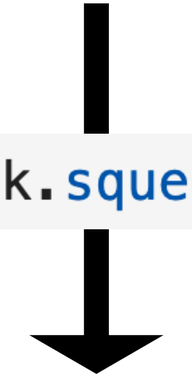
```
<Dimensions [T: 1, C: 2, Z: 25, Y: 400, X: 400]>

(1, 2, 25, 400, 400)
```



Axes of length one

```
print(stack.shape)
```

```
(1, 2, 25, 400, 400)
```



```
stack = stack.squeeze()
```



```
print(stack.shape)
```

```
(2, 25, 400, 400)
```