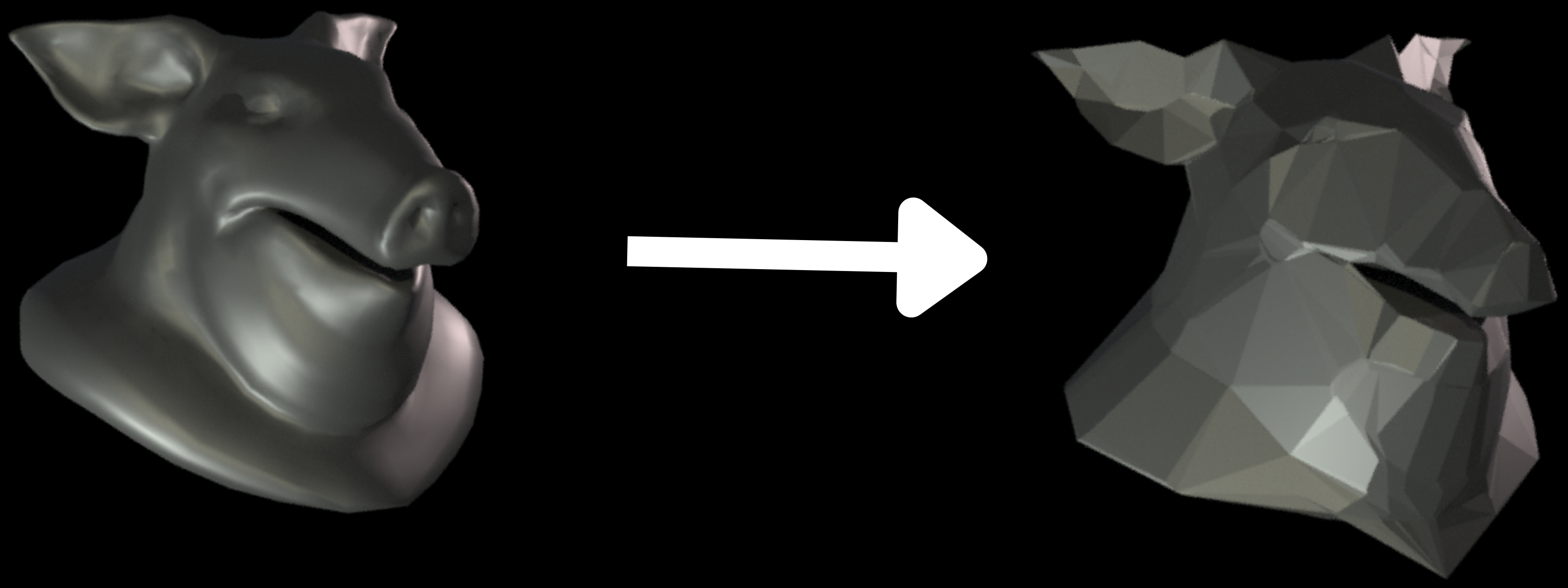
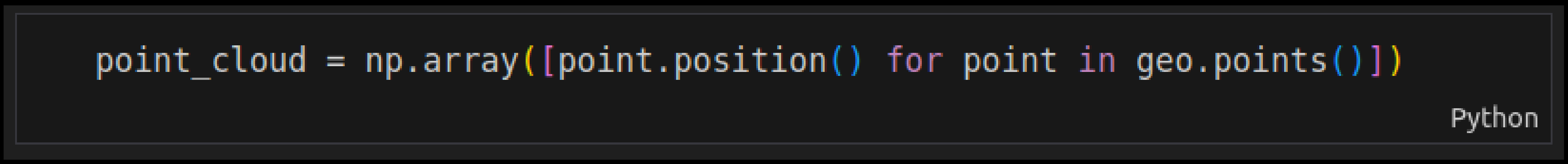
Oriol Manzano
@oriolmanzano.com

Use K-Nearest Neighbour to perform smoothing deformations



Oriol Manzano
@oriolmanzano.com

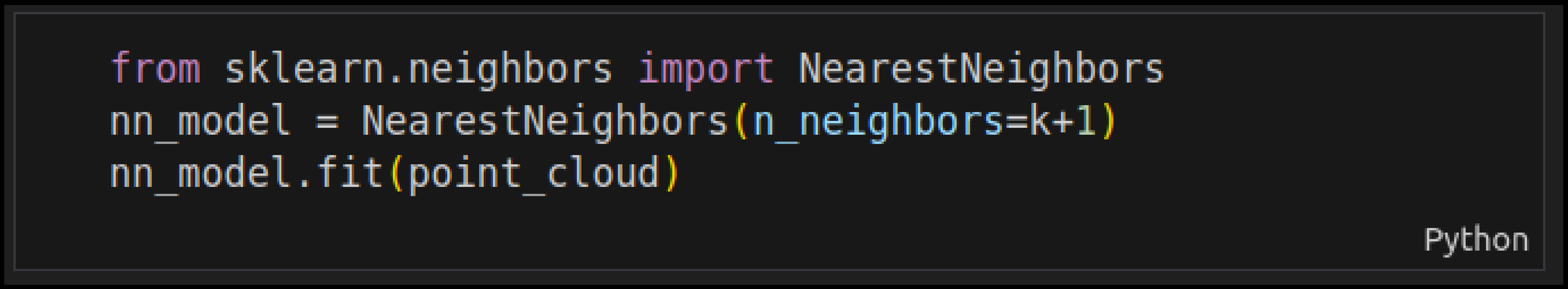
1-Extract the points position of the 3D Shape



```python
point_cloud = np.array([point.position() for point in geo.points()])
```

Oriol Manzano
@oriolmanzano.com

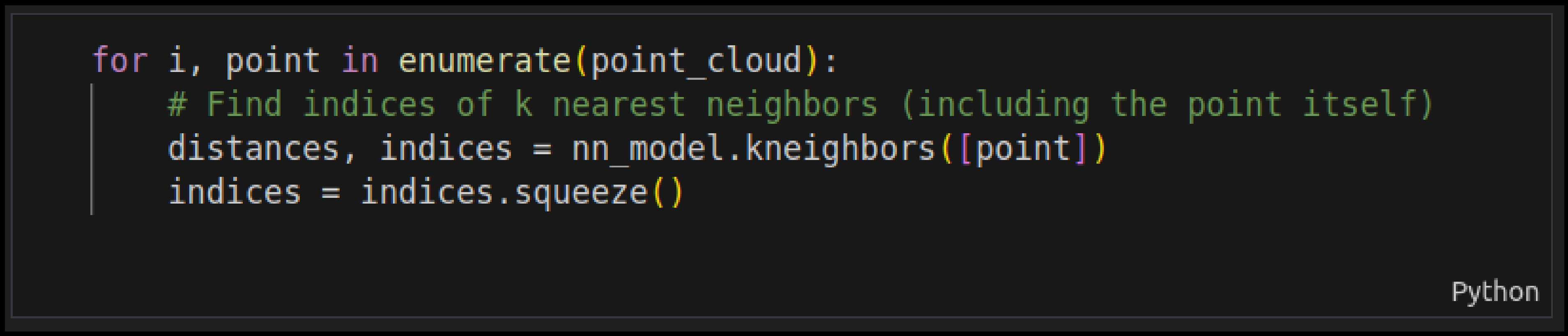
2-Fit the Nearest Neighbours Model with all points data and a specific K value.



```
from sklearn.neighbors import NearestNeighbors
nn_model = NearestNeighbors(n_neighbors=k+1)
nn_model.fit(point_cloud)
```

Oriol Manzano
@oriolmanzano.com

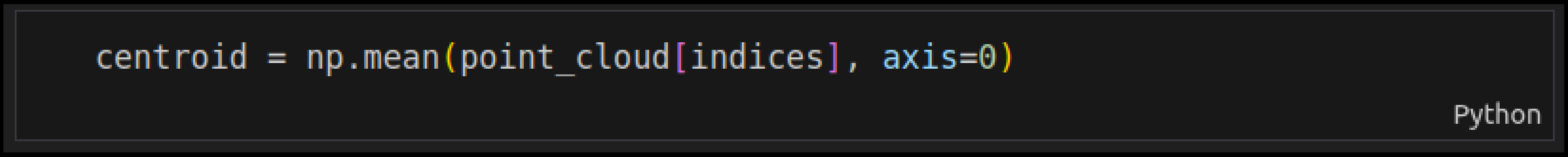
3-For each point find the closest neighbour



```
for i, point in enumerate(point_cloud):
    # Find indices of k nearest neighbors (including the point itself)
    distances, indices = nn_model.kneighbors([point])
    indices = indices.squeeze()
```

Oriol Manzano ✓
@oriolmanzano.com

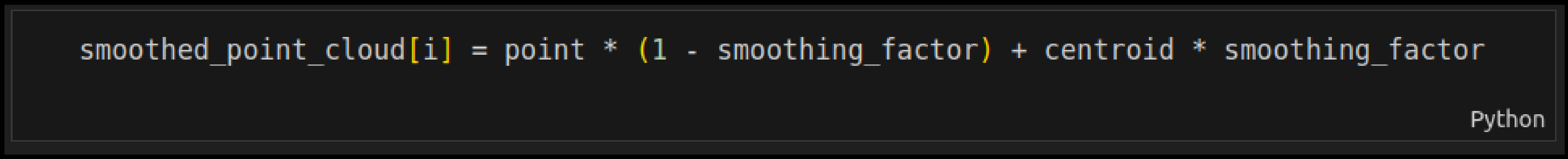
4-Calculate the centroid of k nearest neighbors



```
centroid = np.mean(point_cloud[indices], axis=0)
```

Oriol Manzano ✓
@oriolmanzano.com

5-Update the position of the point based on the centroid



```
smoothed_point_cloud[i] = point * (1 - smoothing_factor) + centroid * smoothing_factor
```