

# How can I normalize values in a NumPy array between 0 and 1?

Authored by  
**stats writer**

June 25, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I normalize values in a NumPy array between 0 and 1?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152356>

Normalizing values in a NumPy array between 0 and 1 refers to the process of rescaling the data in a given array to fit within the range of 0 to 1. This can be achieved by subtracting the minimum value in the array from each value and then dividing by the difference between the maximum and minimum value. This results in a new array where the minimum value is 0 and the maximum value is 1, with all other values falling in between. This technique is commonly used in data analysis and machine learning to standardize the data and make it easier to compare and interpret.

## Normalize Values in NumPy Array Between 0 and 1

To normalize the values in a NumPy array to be between 0 and 1, you can use one of the following methods:

### Method 1: Use NumPy

```
import numpy as np
```

```
x_norm = (x-np.min(x))/(np.max(x)-np.min(x))
```

### Method 2: Use Sklearn

```
from sklearn import preprocessing as pre
```

```
x = x.reshape(-1, 1)
```

```
x_norm = pre.MinMaxScaler().fit_transform(x)
```

Both methods assume `x` is the name of the NumPy array you would like to normalize.

The following examples show how to use each method in practice.

### Example 1: Normalize Values Using NumPy

Suppose we have the following NumPy array:

```
import numpy as np
```

```
#create NumPy array
```

```
x = np.array()
```

We can use the following code to normalize each value in the array to be between 0 and 1:

```
#normalize all values to be between 0 and 1
```

```
x_norm = (x-np.min(x))/(np.max(x)-np.min(x))
```

```
#view normalized array
```

```
print(x_norm)
```

Each value in the NumPy array has been normalized to be between 0 and 1.

Here's how it worked:

The minimum value in the dataset is 13 and the maximum value is 71.

To normalize the first value of 13, we would apply the formula shared earlier:

$$z_i = (x_i - \min(x)) / (\max(x) - \min(x)) = (13 - 13) / (71 - 13) = 0$$
$$z_i = (x_i - \min(x)) / (\max(x) - \min(x)) = (16 - 13) / (71 - 13) = .0517$$

To normalize the third value of 19, we would use the same formula:

$$z_i = (x_i - \min(x)) / (\max(x) - \min(x)) = (19 - 13) / (71 - 13) = .1034$$

We use this same formula to normalize each value in the original NumPy array to be between 0 and 1.

**Example 2: Normalize Values Using sklearn**

Once again, suppose we have the following NumPy array:

```
import numpy as np
```

```
#create NumPy array
```

```
x = np.array()
```

**We can use the `MinMaxScaler()` function from `sklearn` to normalize each value in the array to be between 0 and 1:**

```
from sklearn import preprocessing as pre
```

```
#reshape array so that it works with sklearn
```

```
x = x.reshape(-1, 1)
```

```
#normalize all values to be between 0 and 1
```

```
x_norm = pre.MinMaxScaler().fit_transform(x)
```

```
#view normalized array
```

```
print(x_norm)
```

```
]
```

**Each value in the NumPy array has been normalized to be between 0 and 1.**

**Notice that these normalized values match the ones calculated using the previous method.**

**The following tutorials explain how to perform other**

## common tasks in NumPy:

ARABPSYCHOLOGY.COM