

# 1 Setting width and number of decimal places in NumPy print output

The function below may be used to set both the number of decimal places and the fixed width of NumPy print out. If no width is given it defaults to zero (no extra padding added). Setting width will ensure alignment of output.

```
import numpy as np

# We can set NumPy to print to a fixed number of decimal places:
# e.g. the 8 decimal places that are standard, with no set width

np.set_printoptions(formatter={'float': lambda x: "{0:0.8f}".format(x)})

# Or to make things more flexible/clearer in the code we can set up a function.
# This function will define both fixed width and decimal places/
# If no width is given it will default to zero, with no extra spacing added

def set_numpy_decimal_places(places, width=0):
    set_np = '{0:' + str(width) + '.' + str(places) + 'f}'
    np.set_printoptions(formatter={'float': lambda x: set_np.format(x)})
```

Let's generate some random number data:

```
x = np.random.rand(3,3)
```

The array as originally printed:

```
print (x)
```

OUT:

```
[[0.50873497 0.60515928 0.41612647]
 [0.56755605 0.58378952 0.11467147]
 [0.49378493 0.99466011 0.70781679]]
```

Setting the number of decimal places:

```
set_numpy_decimal_places(3)
print (x)
```

OUT:

```
set_numpy_decimal_places(3)
```

```
print (x)
```

```
[[0.509 0.605 0.416]
 [0.568 0.584 0.115]
 [0.494 0.995 0.708]]
```

Setting the number of decimal places and a fixed width:

```
set_numpy_decimal_places(3, 6)
print (x)
```

OUT:

```
set_numpy_decimal_places(3, 6)
```

```
print (x)
```

```
[[ 0.509  0.605  0.416]
 [ 0.568  0.584  0.115]
 [ 0.494  0.995  0.708]]
```