```
import numpy as np
```

```
import numpy as np

zeros_array = np.zeros(10)
print(zeros_array)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
import numpy as np

ones_array = np.ones(10)
print(ones_array)
```

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

```
import numpy as np

fives_array = np.full(10, 5)
print(fives_array)
```

```
[5 5 5 5 5 5 5 5 5 5]
```

```
integer_array = [x for x in range(10, 51)]


print(integer_array)
```

```
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
44, 45, 46, 47, 48, 49, 50]
```

```
even_integer_array = [x for x in range(10, 51) if x % 2 == 0]


print(even_integer_array)
```

```
[10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
44, 46, 48, 50]
```

```
import numpy as np


matrix = np.arange(9).reshape(3, 3)


print(matrix)
```

```
[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

```
import numpy as np

# Create a 3x3 identity matrix using NumPy
identity_matrix = np.eye(3)

# Printing the identity matrix
print(identity_matrix)
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

```
import numpy as np

# Generate a random number between 0 and 1
random_number = np.random.random()

# Print the random number
print(random_number)
```

```
0.7353342677627976
```

```
import numpy as np

# Generate an array of 25 random numbers from a standard normal
distribution
random_numbers = np.random.randn(25)

# Print the array of random numbers
print(random_numbers)
```

```
[-0.39106013  1.45155633  1.88092782  1.17072436 -1.03477743 -
0.67918309
 -0.50202457 -0.10556039 -1.39827079 -0.77724164 -1.15754751 -
0.38958856
 -0.30568667 -0.92713656 -0.49414303 -0.46496396 -0.96621074 -
0.04511306
  0.70213826 -1.60616452  1.37197971  1.60339412  0.33266995 -
0.86813093
  3.22666192]
```

```
import numpy as np

# Create the desired matrix
matrix = np.arange(0.01, 1.01, 0.01).reshape(10, 10)

# Print the matrix
print(matrix)
```

```
[[0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 ]
 [0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19 0.2 ]
 [0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29 0.3 ]
```

```
 [0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4 ]
 [0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.5 ]
 [0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59 0.6 ]
 [0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7 ]
 [0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 ]
 [0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.9 ]
 [0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 1.  ]]
```

```
import numpy as np

# Create the array of 20 linearly spaced points between 0 and 1
points = np.linspace(0, 1, 20)

# Print the array
print(points)
```

```
[0.         0.05263158 0.10526316 0.15789474 0.21052632 0.26315789
 0.31578947 0.36842105 0.42105263 0.47368421 0.52631579 0.57894737
 0.63157895 0.68421053 0.73684211 0.78947368 0.84210526 0.89473684
 0.94736842 1.        ]
```

```
mat = np.arange(1,26).reshape(5,5)
mat
```

```
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10],
       [11, 12, 13, 14, 15],
       [16, 17, 18, 19, 20],
       [21, 22, 23, 24, 25]])
```

```
# Original matrix
mat = np.arange(1, 26).reshape(5, 5)

# Desired output
result = mat[2:, 1:]

# Print the result
print(result)
```

```
[[12 13 14 15]
 [17 18 19 20]
 [22 23 24 25]]
```

```
# Original matrix
mat = np.arange(1, 26).reshape(5, 5)

# Get the value at row 4 (index 3) and column 5 (index 4)
output_value = mat[3, 4]

# Print the output value
print(output_value)
```

```
20
```

```
# Original matrix
mat = np.arange(1, 26).reshape(5, 5)

# Extract the desired subarray
output_array = mat[:3, 1:2]

# Print the output array
print(output_array)
```

```
[[ 2]
 [ 7]
 [12]]
```

```
# Original matrix
mat = np.arange(1, 26).reshape(5, 5)

# Extract the desired row (the last row)
output_array = mat[4, :]

# Print the output array
print(output_array)
```

```
[21 22 23 24 25]
```

```
# Original matrix
mat = np.arange(1, 26).reshape(5, 5)

# Calculate the sum of all values in mat
total_sum = np.sum(mat)

# Print the sum
print(total_sum)
```

```
325
```

```
# Original matrix
mat = np.arange(1, 26).reshape(5, 5)

# Calculate the standard deviation of the values in mat
std_deviation = np.std(mat)

# Print the standard deviation
print(std_deviation)
```

```
7.211102550927978
```

```
# Original matrix
mat = np.arange(1, 26).reshape(5, 5)

# Calculate the sum of each column in mat
```

```
column_sums = np.sum(mat, axis=0)

# Print the column sums
print(column_sums)
```

```
[55 60 65 70 75]
```