In [1]:
```
import numpy as np
```

1. Extract all odd numbers from the arr.

Input:

In [ ]:
```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Desired output:

In [ ]:
```
array([1, 3, 5, 7, 9])
```

Solution:

In [4]:
```
# Input
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

# Solution
arr[arr % 2 == 0]
```

Out[4]:
```
array([0, 2, 4, 6, 8])
```

2. Replace all odd numbers in arr with -1.

Input:

In [ ]:
```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Desired Output:

In [ ]:
```
array([ 0, -1,  2, -1,  4, -1,  6, -1,  8, -1])
```

Solutions

In [5]:
```
arr[arr % 2 == 1] = -7
arr
```

Out[5]:
```
array([ 0, -7,  2, -7,  4, -7,  6, -7,  8, -7])
```

3. Get the common items between a and b.

Input:

In [ ]:
```
a = np.array([1,2,3,2,3,4,3,4,5,6])
b = np.array([7,2,10,2,7,4,9,4,9,8])
```

Desired output:

In [12]:
```
array([2, 4])
```

Solution:

In [13]:
```
a = np.array([1,2,3,2,3,4,3,4,5,6])
b = np.array([7,2,10,2,7,4,9,4,9,8])
np.intersect1d(a,b)
```

Out[13]:
```
array([2, 4])
```

4. Create as many differrent arrays as you can with the numbers 1 to 12 inclusive as the array's element.

Input:

In [33]:
```
np.arange(1,13)
```

Out[33]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

One possible output:

In [ ]:
```
array([[1, 2, 3, 4, 5, 6],
       [7, 8, 9, 10, 11, 12]])
#2x6 array
```

Solution:

In [11]:
```
arr = np.arange(1,13)
arr.reshape(6, -1)  # Setting to -1 automatically decides the number of col
```

Out[11]:
```
array([[ 1,  2],
       [ 3,  4],
       [ 5,  6],
       [ 7,  8],
       [ 9, 10],
       [11, 12]])
```

5. Convert the function maxx that works on two scalars, to work on two arrays.

Input:

In [1]:
```
def maxx(x, y):
    """Get the maximum of two items"""
    if x >= y:
        return x
    else:
        return y

maxx(1, 5)
#> 5
```

Out[1]:
```
5
```

In [ ]:
```
a = np.array([5, 7, 9, 8, 6, 4, 5])
b = np.array([6, 3, 4, 8, 9, 7, 1])
```

Desired output:

In [ ]:
```
pair_max(a, b)
#> array([ 6.,  7.,  9.,  8.,  9.,  7.,  5.])
```

Solution:

In [8]:
```
def maxx(x, y):
    """Get the maximum of two items"""
    if x >= y:
        return x
    else:
        return y

pair_max = np.vectorize(maxx)

a = np.array([5, 7, 9, 8, 6, 4, 5])
b = np.array([6, 3, 4, 8, 9, 7, 1])

pair_max(a, b)
```

Out[8]:
```
array([6, 7, 9, 8, 9, 7, 5])
```

6. Reverse the rows of a 3x3 array.

In [13]:
```
arr = np.arange(9).reshape(3,3)
arr1 = arr
print(arr)
print(arr1[::-1])
```

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

7. Create a one-dimensional array out of the maximum values for each row in the given array.

Input:

In [59]:
```
np.random.seed(100)
a = np.random.randint(1,10, [5,3])
a
```

Out[59]:
```
array([[9, 9, 4],
       [8, 8, 1],
       [5, 3, 6],
       [3, 3, 3],
       [2, 1, 9]])
```

Desired output:

In [ ]:
```
array([9, 8, 6, 3, 9])
```

Solution:

In [62]:
```
np.random.seed(100)
a = np.random.randint(1,10, [5,3])
a

# Solution 1
np.amax(a, axis=1)

# Solution 2
np.apply_along_axis(np.max, arr=a, axis=1)
```

Out[62]:
```
array([9, 8, 6, 3, 9])
```