Replace all odd numbers in arr with -1 without changing `arr`

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

Ausgabe:

```
#>  array([ 0, -1,  2, -1,  4, -1,  6, -1,  8, -1])
arr
#>  array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Lsg:

```
arr = np.arange(10)
out = np.where(arr % 2 == 1, -1, arr)
print(arr)
out
#> [0 1 2 3 4 5 6 7 8 9]
array([ 0, -1,  2, -1,  4, -1,  6, -1,  8, -1])
```

Get the common items between `a` and `b`

Input:

```
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:

```
array([2, 4])
```

LSG

```
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)
#> array([2, 4])
```

Stack arrays `a` and `b` vertically

Input

```
a = np.arange(10).reshape(2,-1)
b = np.repeat(1, 10).reshape(2,-1)
```

Desired Output:

```
#> array([[0, 1, 2, 3, 4],
#>        [5, 6, 7, 8, 9],
#>        [1, 1, 1, 1, 1],
#>        [1, 1, 1, 1, 1]])
```

LSG

```
a = np.arange(10).reshape(2,-1)
b = np.repeat(1, 10).reshape(2,-1)

# Answers
# Method 1:
np.concatenate([a, b], axis=0)

# Method 2:
np.vstack([a, b])

# Method 3:
np.r_[a, b]
#> array([[0, 1, 2, 3, 4],
#>        [5, 6, 7, 8, 9],
#>        [1, 1, 1, 1, 1],
#>        [1, 1, 1, 1, 1]])
```

Stack the arrays `a` and `b` horizontally.

Input

```
a = np.arange(10).reshape(2,-1)

b = np.repeat(1, 10).reshape(2,-1)
```

Desired Output:

```
#> array([[0, 1, 2, 3, 4, 1, 1, 1, 1, 1],
#>        [5, 6, 7, 8, 9, 1, 1, 1, 1, 1]])
```

LSG

```
a = np.arange(10).reshape(2,-1)
b = np.repeat(1, 10).reshape(2,-1)

# Answers
# Method 1:
np.concatenate([a, b], axis=1)

# Method 2:
np.hstack([a, b])

# Method 3:
np.c_[a, b]
#> array([[0, 1, 2, 3, 4, 1, 1, 1, 1, 1],
#>        [5, 6, 7, 8, 9, 1, 1, 1, 1, 1]])
```