<http://www.ruby-doc.org/core-2.1.5/Array.html>

The [map](http://www.ruby-doc.org/core-2.1.5/Array.html#method-i-map) method can be used to create a new array based on the original array, but with the values modified by the supplied block:

arr.map { |a| 2\*a } **#=> [2, 4, 6, 8, 10]**

arr **#=> [1, 2, 3, 4, 5]**

arr.map! { |a| a\*\*2 } **#=> [1, 4, 9, 16, 25]**

arr **#=> [1, 4, 9, 16, 25]**

Selecting Items from an [Array](http://www.ruby-doc.org/core-2.1.5/Array.html)¶ ↑

Elements can be selected from an array according to criteria defined in a block. The selection can happen in a destructive or a non-destructive manner. While the destructive operations will modify the array they were called on, the non-destructive methods usually return a new array with the selected elements, but leave the original array unchanged.

Non-destructive Selection¶ ↑

arr = [1, 2, 3, 4, 5, 6]

arr.select { |a| a > 3 } **#=> [4, 5, 6]**

arr.reject { |a| a < 3 } **#=> [3, 4, 5, 6]**

arr.drop\_while { |a| a < 4 } **#=> [4, 5, 6]**

arr **#=> [1, 2, 3, 4, 5, 6]**

Destructive Selection¶ ↑

[select!](http://www.ruby-doc.org/core-2.1.5/Array.html#method-i-select-21) and [reject!](http://www.ruby-doc.org/core-2.1.5/Array.html#method-i-reject-21) are the corresponding destructive methods to [select](http://www.ruby-doc.org/core-2.1.5/Array.html#method-i-select) and [reject](http://www.ruby-doc.org/core-2.1.5/Array.html#method-i-reject)

Similar to [select](http://www.ruby-doc.org/core-2.1.5/Array.html#method-i-select) vs. [reject](http://www.ruby-doc.org/core-2.1.5/Array.html#method-i-reject), [delete\_if](http://www.ruby-doc.org/core-2.1.5/Array.html#method-i-delete_if) and [keep\_if](http://www.ruby-doc.org/core-2.1.5/Array.html#method-i-keep_if) have the exact opposite result when supplied with the same block:

arr.delete\_if { |a| a < 4 } **#=> [4, 5, 6]**

arr **#=> [4, 5, 6]**

arr = [1, 2, 3, 4, 5, 6]

arr.keep\_if { |a| a < 4 } **#=> [1, 2, 3]**

arr **#=> [1, 2, 3]**