

ARRAY METHODS

ARRAY.MAP(CALLBACK)

What does it do?

```
let arr = [1, 2, 3, 4, 5]
```

The map method will iterate through every item in the array

arr.map(function(element, index, array

...

It will invoke the callback function for each round in the iteration.

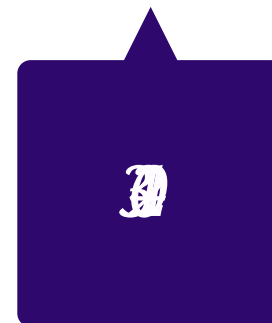
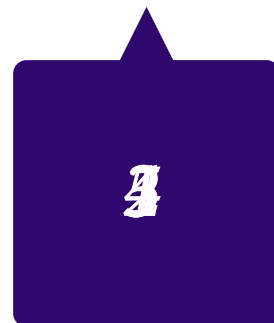
It will return a new array. The elements in this new array will be whatever the callback function returns with each iteration.

ARRAY.MAP(CALLBACK)



```
let arr = [1, 2, 3, 4, 5]
```

```
arr.map( function( element, index, array ) {  
    ...  
})
```



IN ACTION!

ARRAY.MAP(CALLBACK)

```
let arr = [1, 2, 3, 4, 5]
```

[3, 4, 5, 6, 7]

```
let arr = arr.map(function(element, index, array) {  
    return element + 2  
})
```

4

ARRAY.FILTER(CALLBACK)

What does it do?

```
let arr = [1, 2, 3, 4, 5]
```

The filter method will iterate through every item in the array

and it is being called from `function(element, index, array)`

...

It will invoke the callback function for each round in the iteration.

*It will return a new array. The callback method will return either **truthy** or **falsy**.*

If the return value is truthy, the current element will be included in the new array. If falsy is returned, the current element will not be included in the new array.

IN ACTION!

ARRAY.FILTER(CALLBACK)

```
let arr = [1, 2, 3, 4, 5]
```

[4, 5]

```
let newArr = arr.filter(function(element, index, array) {  
    return element > 3  
})
```

False

ARRAY.REDUCE(CALLBACK)

What does it do?

```
let arr = [1, 2, 3, 4, 5]
```

The reduce method will iterate through every item in the

array that is being called from total, element, index, array

...

It will invoke the callback function for each round in the iteration.

It will return a single value which is the resulting value of the total parameter.

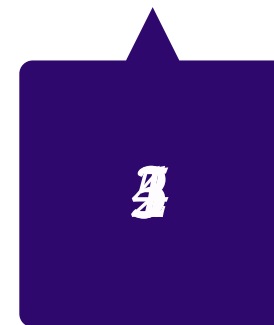
IN ACTION!

ARRAY.REDUCE(CALLBACK)

```
let arr = [1, 2, 3, 4, 5]
```



```
arr.reduce(function(total, element, index, array) {  
    return total + element  
})
```



CLASSES



WHAT IS A CLASS?

A blue print or instructions... to create an object.

A class is a 'special function'.

```
class Blueprint {  
    constructor() {  
        this.number = 10  
    }  
}
```



*A constructor is the 'set up' method.
It builds the object.*

```
let newObject = new Blueprint() => {number: 10}
```