

Array Iteration

Objectives

- Use a for loop to iterate over an array
- Use forEach to iterate over an array
- Compare and contrast for loops and forEach

For Loop

To loop over an array using a for loop, we need to make use of the array's *length* property

```
var colors = ["red", "orange", "yellow", "green"];

for(var i = 0; i < colors.length; i++) {
  console.log(colors[i]);
}
```

ForEach

JavaScript provides an easy built-in way of iterating over an array: ForEach

```
arr.forEach(someFunction)
```

```
var colors = ["red", "orange", "yellow", "green"];  
  
colors.forEach(function(color){  
  //color is a placeholder, call it whatever you want  
  console.log(color);  
});
```

For vs. ForEach

The following 2 code snippets do the same thing:

```
//with a for loop
var colors = ["red", "orange", "yellow", "green"];

for(var i = 0; i < colors.length; i++) {
    console.log(colors[i]);
}
```

```
//using forEach
var colors = ["red", "orange", "yellow", "green"];

colors.forEach(function(color){
    console.log(color);
});

colors.forEach(myfunction)
```

Exercise

What does the following code print out?

```
var numbers = [1,2,3,4,5,6,7,8,9,10];
var colors = ["red", "orange", "yellow", "green"];

numbers.forEach(function(color) {
  if(color% 3 === 0) {
    console.log(color);
  }
});

function myForEach(arr, func){
  //loop through the array
  for(var i = 0; i < arr.length; i++) {
    //call func for each item in array
    func(arr[i]);
  }
}

myForEach(myarr, function(element)
{ alert(element); });

myForEach(myarr, alert);

//function without () is just the function name without execution
```