find/findIndex

Goals

- Understand what find and findIndex do
- Write your own version of find and findIndex

Find

- Iterates through an array
- Runs a callback on each value in the array
- If the callback returns true at any point, return the value in the array that we're iterating over
- Otherwise, return undefined

An Example

```js
let arr = [1,2,3];

arr.find(function(value, index, array){
  return value === 2;
});

// 2
let arr = [1,2,3];

arr.find(function(value, index, array){
  return value < 1;
});

// undefined
```

How Does It Work?

- Iterates through an array
- Runs a callback on each value in the array
- If the callback returns true at any point, return the value we're iterating over
- Otherwise, return undefined

```js
function find(array, callback){
  for(let i = 0; i < array.length; i++){
    if(callback(array[i], i, array) === true){
      return array[i]
    }
```

```
    }
}
```

Using Find In A Function

```
function findOddNumber(arr){
  return arr.find(function(value){
    return value % 2 !== 0
  });
}

findOddNumber([2,3,4,5]) // 3
findOddNumber([4,6,8,10]) // undefined
```

```
function ensureNoSubArrays(arr){
  return arr.find(Array.isArray) === undefined
}

ensureNoSubArrays([1, 2, [3,4]]); // false
ensureNoSubArrays([1,2,3,4]); // true
```

When You Would Use Find

- You need to determine if a value in an array exists and you have to determine this by using a callback
- A simple alternative to using filter and accessing the first element of the filtered array

findIndex

- Iterates through an array
- Runs a callback on each value in the array
- If the callback returns true for any single value, return the index at which that value is found
- Otherwise, return -1

An Example

```
let arr = [1,2,3];

arr.findIndex(function(value, index, array){
  return value < 2;
}); // 0

let arr = [1,2,3];

arr.findIndex(function(value, index, array){
  return value > 3;
}); // -1
```

How Does It Work?

- Iterates through an array
- Runs a callback on each value in the array
- If the callback returns true for any single value, return the index at which that value is found
- Otherwise, return -1

```javascript
function findIndex(array, callback){
  for(let i = 0; i < array.length; i++){
    if(callback(array[i], i, array) === true){
      return i;
    }
  }
  return -1;
}
```

Using Findindex In A Function

```javascript
function findPositionOfBoolean(list){
  return list.findIndex(function(value, index, array){
    return typeof value === "boolean"
  });
}

findPositionOfBoolean([1,3,false,true]) // 2
findPositionOfBoolean(["no booleans", "around", "these parts"]) // -1
```

When You Would Use Findindex

- You need to determine the index of a value in an array if it exists and you have to determine this by using a callback
- A better version of indexOf to be used when a callback is necessary

Recap

- find and findIndex iterate through an array and runs a callback on each value
- if the callback at any time returns true, find returns the value or undefined if not found
- if the callback at any time returns true, findIndex returns the index or -1 if not found