FUNCTIONAL PROGRAMMING

1) Basics of Functional Programming

OLD CODE	NEW CODE

// Function that returns a string representing a cup of green tea
const prepareTea = () => 'greenTea';
/*
Given a function (representing the tea type) and number of cups needed, the
following function returns an array of strings (each representing a cup of
a specific type of tea).
*/
const getTea = (numOfCups) => {
  const teaCups = [];
 	
  for(let cups = 1; cups <= numOfCups; cups += 1) {
    const teaCup = prepareTea();
    teaCups.push(teaCup);
  }
  return teaCups;
};
 	
// Only change code below this line
const tea4TeamFCC = null;
// Only change code above this line
	//Function that returns a string representing a cup of green tea
	const prepareTea = () => 'greenTea';
	/*
	Given a function (representing the tea type) and number of cups needed, the
	following function returns an array of strings (each representing a cup of
	a specific type of tea).
	*/
	const getTea = (numOfCups) => {
	  const teaCups = [];
	
	  for(let cups = 1; cups <= numOfCups; cups += 1) {
	    const teaCup = prepareTea();
	    teaCups.push(teaCup);
	  }
	  return teaCups;
	};
	// Only change code below this line
	const tea4TeamFCC = getTea(40);
	// Only change code above this line

2) Terminology

OLD CODE	NEW CODE
// Function that returns a string representing a cup of green tea	// Function that returns a string representing a cup of green tea

```javascript
const prepareGreenTea = () => 'greenTea';

// Function that returns a string representing a cup of black tea
const prepareBlackTea = () => 'blackTea';

/*
Given a function (representing the tea type) and number of cups needed, the
following function returns an array of strings (each representing a cup of
a specific type of tea).
*/
const getTea = (prepareTea, numOfCups) => {
  const teaCups = [];

  for(let cups = 1; cups <= numOfCups; cups += 1) {
    const teaCup = prepareTea();
    teaCups.push(teaCup);
  }
  return teaCups;
};

// Only change code below this line
const tea4GreenTeamFCC = null;
const tea4BlackTeamFCC = null;
// Only change code above this line

console.log(
  tea4GreenTeamFCC,
  tea4BlackTeamFCC
);
```

```javascript
const prepareGreenTea = () => 'greenTea';

// Function that returns a string representing a cup of black tea
const prepareBlackTea = () => 'blackTea';

/*
Given a function (representing the tea type) and number of cups needed, the
following function returns an array of strings (each representing a cup of
a specific type of tea).
*/
const getTea = (prepareTea, numOfCups) => {
  const teaCups = [];

  for(let cups = 1; cups <= numOfCups; cups += 1) {
    const teaCup = prepareTea();
    teaCups.push(teaCup);
  }
  return teaCups;
};

// Only change code below this line
const tea4GreenTeamFCC = getTea(prepareGreenTea, 27);
const tea4BlackTeamFCC = getTea(prepareBlackTea, 13);
// Only change code above this line

console.log(
  tea4GreenTeamFCC,
  tea4BlackTeamFCC
);
```

3) Hazards of using imperative code

OLD CODE	NEW CODE
`// tabs is an array of titles of each site open within the window`	`// tabs is an array of titles of each site open within the window`
`var Window = function(tabs) {`	`var Window = function(tabs) {`
`  this.tabs = tabs; // We keep a record of the array inside the object`	`  this.tabs = tabs; // We keep a record of the array inside the object`
`};`	`};`
`// When you join two windows into one window`	`// When you join two windows into one window`
`Window.prototype.join = function (otherWindow) {`	`Window.prototype.join = function (otherWindow) {`
`  this.tabs = this.tabs.concat(otherWindow.tabs);`	`  this.tabs = this.tabs.concat(otherWindow.tabs);`
`  return this;`	`  return this;`
`};`	`};`
`// When you open a new tab at the end`	`// When you open a new tab at the end`
`Window.prototype.tabOpen = function (tab) {`	`Window.prototype.tabOpen = function (tab) {`
`  this.tabs.push('new tab'); // Let's open a new tab for now`	`  this.tabs.push('new tab'); // Let's open a new tab for now`
`  return this;`	`  return this;`
`};`	`};`
`// When you close a tab`	`// When you close a tab`
`Window.prototype.tabClose = function (index) {`	`Window.prototype.tabClose = function (index) {`
` // Only change code below this line`	`// Only change code below this line`
`  var tabsBeforeIndex = this.tabs.splice(0, index); // Get the tabs before the tab`	`  var tabsBeforeIndex = this.tabs.splice(0, index); // Get the tabs before the tab`
`  var tabsAfterIndex = this.tabs.splice(index + 1); // Get the tabs after the tab`	`  var tabsAfterIndex = this.tabs.splice(1); // Get the tabs after the tab`
`  this.tabs = tabsBeforeIndex.concat(tabsAfterIndex); // Join them together`	`  this.tabs = tabsBeforeIndex.concat(tabsAfterIndex); // Join them together`

// Only change code above this line
 	
return this;
};
 	
// Let's create three browser windows
var workWindow = new Window(['GMail', 'Inbox', 'Work mail', 'Docs', 'freeCodeCamp']); // Your mailbox, drive, and other work sites
var socialWindow = new Window(['FB', 'Gitter', 'Reddit', 'Twitter', 'Medium']); // Social sites
var videoWindow = new Window(['Netflix', 'YouTube', 'Vimeo', 'Vine']); // Entertainment sites
 	
// Now perform the tab opening, closing, and other operations
var finalTabs = socialWindow
.tabOpen() // Open a new tab for cat memes
.join(videoWindow.tabClose(2)) // Close third tab in video window, and join
.join(workWindow.tabClose(1).tabOpen());
console.log(finalTabs.tabs);
	// Only change code above this line
	
	return this;
	};
	
	// Let's create three browser windows
	var workWindow = new Window(['GMail', 'Inbox', 'Work mail', 'Docs', 'freeCodeCamp']); // Your mailbox, drive, and other work sites
	var socialWindow = new Window(['FB', 'Gitter', 'Reddit', 'Twitter', 'Medium']); // Social sites
	var videoWindow = new Window(['Netflix', 'YouTube', 'Vimeo', 'Vine']); // Entertainment sites
	
	// Now perform the tab opening, closing, and other operations
	var finalTabs = socialWindow
	.tabOpen() // Open a new tab for cat memes
	.join(videoWindow.tabClose(2)) // Close third tab in video window, and join
	.join(workWindow.tabClose(1).tabOpen());
	console.log(finalTabs.tabs);

4) Avoiding mutations and side effects

OLD CODE	NEW CODE

// The global variable
var fixedValue = 4;
 	
function incrementer () {
// Only change code below this line
	// The global variable
	var fixedValue = 4;
	
	function incrementer () {
	// Only change code below this line

// Only change code above this line
}
	var total = fixedValue+1;
	return total;
	// Only change code above this line
	}

5) Passing arguments to avoid external dependence in functions

OLD CODE	NEW CODE

// The global variable
var fixedValue = 4;
 	
// Only change code below this line
function incrementer () {
 	
// Only change code above this line
}
	// The global variable
	var fixedValue = 4;
	
	// Only change code below this line
	function incrementer (num) {
	return num+1;
	
	// Only change code above this line
	}

6) Refactoring global variables out of functions

OLD CODE	NEW CODE

// The global variable
var bookList = ["The Hound of the Baskervilles", "On The Electrodynamics of Moving Bodies", "Philosophiæ Naturalis Principia Mathematica", "Disquisitiones Arithmeticae"];
 	
// Change code below this line
function add (bookName) {
 	
bookList.push(bookName);
return bookList;
 	
// Change code above this line
}
// Change code below this line
function remove (bookName) {
	// The global variable
	var bookList = ["The Hound of the Baskervilles", "On The Electrodynamics of Moving Bodies", "Philosophiæ Naturalis Principia Mathematica", "Disquisitiones Arithmeticae"];
	
	// Change code below this line
	function add (arr, bookName) {
	var newbooks = [...arr];
	newbooks.push(bookName);
	return newbooks;
	
	// Change code above this line
	}
	// Change code below this line
	function remove (arr, bookName)

var book_index = bookList.indexOf(bookName);
  if (book_index >= 0) {
 	
    bookList.splice(book_index, 1);
    return bookList;
 	
    // Change code above this line
    }
}
 	
var newBookList = add(bookList, 'A Brief History of Time');
var newerBookList = remove(bookList, 'On The Electrodynamics of Moving Bodies');
var newestBookList = remove(add(bookList, 'A Brief History of Time'), 'On The Electrodynamics of Moving Bodies');
 	
console.log(bookList);
	{
	  var lessbooks = [...arr];
	  var book_index = lessbooks.indexOf(bookName);
	  if (book_index >= 0) {
	
	    lessbooks.splice(book_index, 1);
	    return lessbooks;
	// Change code above this line
	    }
	}
	
	var newBookList = add(bookList, 'A Brief History of Time');
	var newerBookList = remove(bookList, 'On The Electrodynamics of Moving Bodies');
	var newestBookList = remove(add(bookList, 'A Brief History of Time'), 'On The Electrodynamics of Moving Bodies');
	
	console.log(bookList);

7) Using map method to extract data from an array

OLD CODE	NEW CODE

// The global variable
var watchList = [
  {
    "Title": "Inception",
    "Year": "2010",
    "Rated": "PG-13",
    "Released": "16 Jul 2010",
    "Runtime": "148 min",
    "Genre": "Action, Adventure, Crime",
    "Director": "Christopher Nolan",
    "Writer": "Christopher Nolan",
    "Actors": "Leonardo DiCapri
	// The global variable
	var watchList = [
	  {
	    "Title": "Inception",
	    "Year": "2010",
	    "Rated": "PG-13",
	    "Released": "16 Jul 2010",
	    "Runtime": "148 min",
	    "Genre": "Action, Adventure, Crime",
	    "Director": "Christopher Nolan",
	    "Writer": "Christopher Nolan",
	    "Actors": "Leonardo DiCaprio

```
o, Joseph Gordon-Levitt, Ellen Page, Tom Hardy",
    "Plot": "A thief, who steals corporate secrets through use of dream-sharing technology, is given the inverse task of planting an idea into the mind of a CEO.",
    "Language": "English, Japanese, French",
    "Country": "USA, UK",
    "Awards": "Won 4 Oscars. Another 143 wins & 198 nominations.",
    "Poster": "http://ia.media-imdb.com/images/M/MV5BMjAxMzY3NjcxNF5BMl5BanBnXkFtZTcwNTI5OTM0Mw@@._V1_SX300.jpg",
    "Metascore": "74",
    "imdbRating": "8.8",
    "imdbVotes": "1,446,708",
    "imdbID": "tt1375666",
    "Type": "movie",
    "Response": "True"
  },
  {
    "Title": "Interstellar",
    "Year": "2014",
    "Rated": "PG-13",
    "Released": "07 Nov 2014",
    "Runtime": "169 min",
    "Genre": "Adventure, Drama, Sci-Fi",
    "Director": "Christopher Nolan",
    "Writer": "Jonathan Nolan, Christopher Nolan",
    "Actors": "Ellen Burstyn, Matthew McConaughey, Mackenzie Foy, John Lithgow",
    "Plot": "A team of explorers travel through a wormhole in space in an attempt to ensure h
```

```
, Joseph Gordon-Levitt, Ellen Page, Tom Hardy",
    "Plot": "A thief, who steals corporate secrets through use of dream-sharing technology, is given the inverse task of planting an idea into the mind of a CEO.",
    "Language": "English, Japanese, French",
    "Country": "USA, UK",
    "Awards": "Won 4 Oscars. Another 143 wins & 198 nominations.",
    "Poster": "http://ia.media-imdb.com/images/M/MV5BMjAxMzY3NjcxNF5BMl5BanBnXkFtZTcwNTI5OTM0Mw@@._V1_SX300.jpg",
    "Metascore": "74",
    "imdbRating": "8.8",
    "imdbVotes": "1,446,708",
    "imdbID": "tt1375666",
    "Type": "movie",
    "Response": "True"
  },
  {
    "Title": "Interstellar",
    "Year": "2014",
    "Rated": "PG-13",
    "Released": "07 Nov 2014",
    "Runtime": "169 min",
    "Genre": "Adventure, Drama, Sci-Fi",
    "Director": "Christopher Nolan",
    "Writer": "Jonathan Nolan, Christopher Nolan",
    "Actors": "Ellen Burstyn, Matthew McConaughey, Mackenzie Foy, John Lithgow",
    "Plot": "A team of explorers travel through a wormhole in space in an attempt to ensure huma
```

```
umanity's survival.",
    "Language": "English",
    "Country": "USA, UK",
    "Awards": "Won 1 Oscar. Another 39 wins & 132 nominations.",
    "Poster": "http://ia.media-imdb.com/images/M/MV5BMjIxNTU4MzY4MF5BMl5BanBnXkFtZTgwMzM4ODI3MjE@._V1_SX300.jpg",
    "Metascore": "74",
    "imdbRating": "8.6",
    "imdbVotes": "910,366",
    "imdbID": "tt0816692",
    "Type": "movie",
    "Response": "True"
  },
  {
    "Title": "The Dark Knight",
    "Year": "2008",
    "Rated": "PG-13",
    "Released": "18 Jul 2008",
    "Runtime": "152 min",
    "Genre": "Action, Adventure, Crime",
    "Director": "Christopher Nolan",
    "Writer": "Jonathan Nolan (screenplay), Christopher Nolan (screenplay), Christopher Nolan (story), David S. Goyer (story), Bob Kane (characters)",
    "Actors": "Christian Bale, Heath Ledger, Aaron Eckhart, Michael Caine",
    "Plot": "When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, the caped crusader must come to terms with one of the greatest psychological tests of his ability to fight injustice.",
    "Language": "English, Manda
```

```
nity's survival.",
    "Language": "English",
    "Country": "USA, UK",
    "Awards": "Won 1 Oscar. Another 39 wins & 132 nominations.",
    "Poster": "http://ia.media-imdb.com/images/M/MV5BMjIxNTU4MzY4MF5BMl5BanBnXkFtZTgwMzM4ODI3MjE@._V1_SX300.jpg",
    "Metascore": "74",
    "imdbRating": "8.6",
    "imdbVotes": "910,366",
    "imdbID": "tt0816692",
    "Type": "movie",
    "Response": "True"
  },
  {
    "Title": "The Dark Knight",
    "Year": "2008",
    "Rated": "PG-13",
    "Released": "18 Jul 2008",
    "Runtime": "152 min",
    "Genre": "Action, Adventure, Crime",
    "Director": "Christopher Nolan",
    "Writer": "Jonathan Nolan (screenplay), Christopher Nolan (screenplay), Christopher Nolan (story), David S. Goyer (story), Bob Kane (characters)",
    "Actors": "Christian Bale, Heath Ledger, Aaron Eckhart, Michael Caine",
    "Plot": "When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, the caped crusader must come to terms with one of the greatest psychological tests of his ability to fight injustice.",
    "Language": "English, Mandarin",
rin",
    "Country": "USA, UK",
    "Awards": "Won 2 Oscars. An
other 146 wins & 142 nomination
s.",
    "Poster": "http://ia.media-
imdb.com/images/M/MV5BMTMxNTMwO
DM0NF5BMl5BanBnXkFtZTcwODAyMTk2
Mw@@._V1_SX300.jpg",
    "Metascore": "82",
    "imdbRating": "9.0",
    "imdbVotes": "1,652,832",
    "imdbID": "tt0468569",
    "Type": "movie",
    "Response": "True"
  },
  {
    "Title": "Batman Begins",
    "Year": "2005",
    "Rated": "PG-13",
    "Released": "15 Jun 2005",
    "Runtime": "140 min",
    "Genre": "Action, Adventure
",
    "Director": "Christopher No
lan",
    "Writer": "Bob Kane (charac
ters), David S. Goyer (story),
Christopher Nolan (screenplay),
 David S. Goyer (screenplay)",
    "Actors": "Christian Bale,
Michael Caine, Liam Neeson, Kat
ie Holmes",
    "Plot": "After training wit
h his mentor, Batman begins his
 fight to free crime-
ridden Gotham City from the cor
ruption that Scarecrow and the
League of Shadows have cast upo
n it.",
    "Language": "English, Urdu,
 Mandarin",
    "Country": "USA, UK",
```

```
    "Country": "USA, UK",
    "Awards": "Won 2 Oscars. Ano
ther 146 wins & 142 nominations.
",
    "Poster": "http://ia.media-
imdb.com/images/M/MV5BMTMxNTMwOD
M0NF5BMl5BanBnXkFtZTcwODAyMTk2Mw
@@._V1_SX300.jpg",
    "Metascore": "82",
    "imdbRating": "9.0",
    "imdbVotes": "1,652,832",
    "imdbID": "tt0468569",
    "Type": "movie",
    "Response": "True"
  },
  {
    "Title": "Batman Begins",
    "Year": "2005",
    "Rated": "PG-13",
    "Released": "15 Jun 2005",
    "Runtime": "140 min",
    "Genre": "Action, Adventure"
,
    "Director": "Christopher Nol
an",
    "Writer": "Bob Kane (charact
ers), David S. Goyer (story), Ch
ristopher Nolan (screenplay), Da
vid S. Goyer (screenplay)",
    "Actors": "Christian Bale, M
ichael Caine, Liam Neeson, Katie
 Holmes",
    "Plot": "After training with
 his mentor, Batman begins his f
ight to free crime-
ridden Gotham City from the corr
uption that Scarecrow and the Le
ague of Shadows have cast upon i
t.",
    "Language": "English, Urdu,
Mandarin",
    "Country": "USA, UK",
    "Awards": "Nominated for 1 O
    "Awards": "Nominated for 1 Oscar. Another 15 wins & 66 nominations.",
    "Poster": "http://ia.media-imdb.com/images/M/MV5BNTM3OTc0MzM2OV5BMl5BanBnXkFtZTYwNzUwMTI3._V1_SX300.jpg",
    "Metascore": "70",
    "imdbRating": "8.3",
    "imdbVotes": "972,584",
    "imdbID": "tt0372784",
    "Type": "movie",
    "Response": "True"
  },
  {
    "Title": "Avatar",
    "Year": "2009",
    "Rated": "PG-13",
    "Released": "18 Dec 2009",
    "Runtime": "162 min",
    "Genre": "Action, Adventure, Fantasy",
    "Director": "James Cameron",
    "Writer": "James Cameron",
    "Actors": "Sam Worthington, Zoe Saldana, Sigourney Weaver, Stephen Lang",
    "Plot": "A paraplegic marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home.",
    "Language": "English, Spanish",
    "Country": "USA, UK",
    "Awards": "Won 3 Oscars. Another 80 wins & 121 nominations.",
    "Poster": "http://ia.media-imdb.com/images/M/MV5BMTYwOTEwNjAzMl5BMl5BanBnXkFtZTcwODc5MTUw
```

```
scar. Another 15 wins & 66 nominations.",
    "Poster": "http://ia.media-imdb.com/images/M/MV5BNTM3OTc0MzM2OV5BMl5BanBnXkFtZTYwNzUwMTI3._V1_SX300.jpg",
    "Metascore": "70",
    "imdbRating": "8.3",
    "imdbVotes": "972,584",
    "imdbID": "tt0372784",
    "Type": "movie",
    "Response": "True"
  },
  {
    "Title": "Avatar",
    "Year": "2009",
    "Rated": "PG-13",
    "Released": "18 Dec 2009",
    "Runtime": "162 min",
    "Genre": "Action, Adventure, Fantasy",
    "Director": "James Cameron",
    "Writer": "James Cameron",
    "Actors": "Sam Worthington, Zoe Saldana, Sigourney Weaver, Stephen Lang",
    "Plot": "A paraplegic marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home.",
    "Language": "English, Spanish",
    "Country": "USA, UK",
    "Awards": "Won 3 Oscars. Another 80 wins & 121 nominations.",
    "Poster": "http://ia.media-imdb.com/images/M/MV5BMTYwOTEwNjAzMl5BMl5BanBnXkFtZTcwODc5MTUwMw@@._V1_SX300.jpg",
    "Metascore": "83",
```

Mw@@._V1_SX300.jpg",
    "Metascore": "83",
    "imdbRating": "7.9",
    "imdbVotes": "876,575",
    "imdbID": "tt0499549",
    "Type": "movie",
    "Response": "True"
  }
];
 	
// Only change code below this line
 	
var ratings = [];
for(var i=0; i < watchList.length; i++){
  ratings.push({title: watchList[i]["Title"], rating: watchList[i]["imdbRating"]});
}
 	
// Only change code above this line
 	
console.log(JSON.stringify(ratings));
	"imdbRating": "7.9",
	    "imdbVotes": "876,575",
	    "imdbID": "tt0499549",
	    "Type": "movie",
	    "Response": "True"
	  }
	];
	
	// Only change code below this line
	
	var ratings = [];
	  ratings = watchList.map(item => ({
	    title: item["Title"],
	    rating: item["imdbRating"]
	  }));
	  //ratings = watchList.map(movie2 => movie2.imdbRating);
	  //ratings.push({title: watchList[i]["Title"], rating: watchList[i]["imdbRating"]});
	
	// Only change code above this line
	
	console.log(JSON.stringify(ratings));

8) Implementing map on a prototype

OLD CODE	NEW CODE

// The global variable
var s = [23, 65, 98, 5];
 	
Array.prototype.myMap = function(callback){
  var newArray = [];
  // Only change code below this line
 	
  // Only change code above this line
	// The global variable
	var s = [23, 65, 98, 5];
	
	Array.prototype.myMap = function(callback){
	  var newArray = [];
	  // Only change code below this line
	  for(var i=0; i<s.length; i++){
	    newArray[i] = 2*s[i];

return newArray;
 	
};
 	
var new_s = s.myMap(function(item){
  return item * 2;
});
	}
	  // Only change code above this line
	  return newArray;
	
	};
	
	var new_s = s.myMap(function(item){
	  return item * 2;
	});
	console.log(new_s);

9) Using the filter method to extract data from an array

OLD CODE	NEW CODE

// The global variable
var watchList = [
  {
    "Title": "Inception",
    "Year": "2010",
    "Rated": "PG-13",
    "Released": "16 Jul 2010",
    "Runtime": "148 min",
    "Genre": "Action, Adventure, Crime",
    "Director": "Christopher Nolan",
    "Writer": "Christopher Nolan",
    "Actors": "Leonardo DiCaprio, Joseph Gordon-Levitt, Ellen Page, Tom Hardy",
    "Plot": "A thief, who steals corporate secrets through use of dream-sharing technology, is given the inverse task of planting an idea into the mind of a CEO.",
    "Language": "English, Japanese, French",
    "Country": "USA, UK",
	// The global variable
	var watchList = [
	 {
	 "Title": "Inception",
	 "Year": "2010",
	 "Rated": "PG-13",
	 "Released": "16 Jul 2010",
	 "Runtime": "148 min",
	 "Genre": "Action, Adventure, Crime",
	 "Director": "Christopher Nolan",
	 "Writer": "Christopher Nolan",
	 "Actors": "Leonardo DiCaprio, Joseph GordonLevitt, Ellen Page, Tom Hardy",
	 "Plot": "A thief, who steals corporate secrets through use of dreamsharing technology, is given the inverse task of planting an idea into the mind of a CEO.",
	 "Language": "English, Japanese, French",
	 "Country": "USA, UK",
	 "Awards": "Won 4 Oscars. Another 143 wins & 198 nominations.",

"Awards": "Won 4 Oscars. Another 143 wins & 198 nominations.", "Poster": "http://ia.media-imdb.com/images/M/MV5BMjAxMzY3NjcxNF5BMl5BanBnXkFtZTcwNTI5OTM0Mw@@._V1_SX300.jpg", "Metascore": "74", "imdbRating": "8.8", "imdbVotes": "1,446,708", "imdbID": "tt1375666", "Type": "movie", "Response": "True" }, { "Title": "Interstellar", "Year": "2014", "Rated": "PG-13", "Released": "07 Nov 2014", "Runtime": "169 min", "Genre": "Adventure, Drama, Sci-Fi", "Director": "Christopher Nolan", "Writer": "Jonathan Nolan, Christopher Nolan", "Actors": "Ellen Burstyn, Matthew McConaughey, Mackenzie Foy, John Lithgow", "Plot": "A team of explorers travel through a wormhole in space in an attempt to ensure humanity's survival.", "Language": "English", "Country": "USA, UK", "Awards": "Won 1 Oscar. Another 39 wins & 132 nominations.", "Poster": "http://ia.media-imdb.com/images/M/MV5BMjIxNTU4MzY4MF5BMl5BanBnXkFtZTgwMzM4ODI3MjE@._V1_SX300.jpg", "Metascore": "74",	"Poster": "http://ia.mediaimdb.com/images/M/MV5BMjAxMzY3NjcxNF5BMl5BanBnXkFtZTcwNTI5OTM0Mw@@._V1_SX300.jpg", "Metascore": "74", "imdbRating": "8.8", "imdbVotes": "1,446,708", "imdbID": "tt1375666", "Type": "movie", "Response": "True" }, { "Title": "Interstellar", "Year": "2014", "Rated": "PG-13", "Released": "07 Nov 2014", "Runtime": "169 min", "Genre": "Adventure, Drama, Sci-Fi", "Director": "Christopher Nolan", "Writer": "Jonathan Nolan, Christopher Nolan", "Actors": "Ellen Burstyn, Matthew McConaughey, MackenzieFoy, John Lithgow", "Plot": "A team of explorers travel through a wormhole in space in an attempt to ensure humanity's survival.", "Language": "English", "Country": "USA, UK", "Awards": "Won 1 Oscar. Another 39 wins & 132 nominations.", "Poster": "http://ia.mediaimdb.com/images/M/MV5BMjIxNTU4MzY4MF5BMl5BanBnXkFtZTgwMzM4ODI3MjE@._V1_SX300.jpg", "Metascore": "74", "imdbRating": "8.6", "imdbVotes": "910,366", "imdbID": "tt0816692", "Type": "movie",

"imdbRating": "8.6", "imdbVotes": "910,366", "imdbID": "tt0816692", "Type": "movie", "Response": "True" }, { "Title": "The Dark Knight", "Year": "2008", "Rated": "PG-13", "Released": "18 Jul 2008", "Runtime": "152 min", "Genre": "Action, Adventure, Crime", "Director": "Christopher Nolan", "Writer": "Jonathan Nolan (screenplay), Christopher Nolan (screenplay), Christopher Nolan (story), David S. Goyer (story), Bob Kane (characters)", "Actors": "Christian Bale, Heath Ledger, Aaron Eckhart, Michael Caine", "Plot": "When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, the caped crusader must come to terms with one of the greatest psychological tests of his ability to fight injustice.", "Language": "English, Mandarin", "Country": "USA, UK", "Awards": "Won 2 Oscars. Another 146 wins & 142 nominations.", "Poster": "http://ia.media-imdb.com/images/M/MV5BMTMxNTMwODM0NF5BMl5BanBnXkFtZTcwODAyMTk2Mw@@._V1_SX300.jpg", "Metascore": "82", "imdbRating": "9.0",	"Response": "True" }, { "Title": "The Dark Knight", "Year": "2008", "Rated": "PG-13", "Released": "18 Jul 2008", "Runtime": "152 min", "Genre": "Action, Adventure, Crime", "Director": "Christopher Nolan", "Writer": "Jonathan Nolan (screenplay), Christopher Nolan (screenplay), Christopher Nolan (story), David S. Goyer (story), Bob Kane (characters)", "Actors": "Christian Bale, Heath Ledger, Aaron Eckhart,Michael Caine", "Plot": "When the menace known as the Joker wreaks havocand chaos on the people of Gotham, the caped crusader mustcome to terms with one of the greatest psychological tests of his ability to fight injustice.", "Language": "English, Mandarin", "Country": "USA, UK", "Awards": "Won 2 Oscars. Another 146 wins & 142 nominations.", "Poster": "http://ia.mediaimdb.com/images/M/MV5BMTMxNTMwODM0NF5BMl5BanBnXkFtZTcwODAyMTk2Mw@@._V1_SX300.jpg", "Metascore": "82", "imdbRating": "9.0", "imdbVotes": "1,652,832", "imdbID": "tt0468569", "Type": "movie", "Response": "True" },

"imdbVotes": "1,652,832", "imdbID": "tt0468569", "Type": "movie", "Response": "True" }, { "Title": "Batman Begins", "Year": "2005", "Rated": "PG-13", "Released": "15 Jun 2005", "Runtime": "140 min", "Genre": "Action, Adventure", "Director": "Christopher Nolan", "Writer": "Bob Kane (characters), David S. Goyer (story), Christopher Nolan (screenplay), David S. Goyer (screenplay)", "Actors": "Christian Bale, Michael Caine, Liam Neeson, Katie Holmes", "Plot": "After training with his mentor, Batman begins his fight to free crime-ridden Gotham City from the corruption that Scarecrow and the League of Shadows have cast upon it.", "Language": "English, Urdu, Mandarin", "Country": "USA, UK", "Awards": "Nominated for 1 Oscar. Another 15 wins & 66 nominations.", "Poster": "http://ia.mediaimdb.com/images/M/MV5BNTM3OTc0MzM2OV5BMl5BanBnXkFtZTYwNzUwMTI3._V1_SX300.jpg", "Metascore": "70", "imdbRating": "8.3", "imdbVotes": "972,584", "imdbID": "tt0372784",	{ "Title": "Batman Begins", "Year": "2005", "Rated": "PG-13", "Released": "15 Jun 2005", "Runtime": "140 min", "Genre": "Action, Adventure", "Director": "Christopher Nolan", "Writer": "Bob Kane (characters), David S. Goyer (story), Christopher Nolan (screenplay), David S. Goyer (screenplay)", "Actors": "Christian Bale, Michael Caine, Liam Neeson, Katie Holmes", "Plot": "After training with his mentor, Batman begins his fight to free crimeridden Gotham City from the corruption that Scarecrow and the League of Shadows have cast upon it.", "Language": "English, Urdu, Mandarin", "Country": "USA, UK", "Awards": "Nominated for 1 Oscar. Another 15 wins & 66 nominations.", "Poster": "http://ia.mediaimdb.com/images/M/MV5BNTM3OTc0MzM2OV5BMl5BanBnXkFtZTYwNzUwMTI3._V1_SX300.jpg", "Metascore": "70", "imdbRating": "8.3", "imdbVotes": "972,584", "imdbID": "tt0372784", "Type": "movie", "Response": "True" }, { "Title": "Avatar", "Year": "2009", "Rated": "PG-13",

```
    "Type": "movie",
    "Response": "True"
  },
  {
    "Title": "Avatar",
    "Year": "2009",
    "Rated": "PG-13",
    "Released": "18 Dec 2009",
    "Runtime": "162 min",
    "Genre": "Action, Adventure, Fantasy",
    "Director": "James Cameron",
    "Writer": "James Cameron",
    "Actors": "Sam Worthington, Zoe Saldana, Sigourney Weaver, Stephen Lang",
    "Plot": "A paraplegic marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home.",
    "Language": "English, Spanish",
    "Country": "USA, UK",
    "Awards": "Won 3 Oscars. Another 80 wins & 121 nominations.",
    "Poster": "http://ia.media-imdb.com/images/M/MV5BMTYwOTEwNjAzMl5BMl5BanBnXkFtZTcwODc5MTUwMw@@._V1_SX300.jpg",
    "Metascore": "83",
    "imdbRating": "7.9",
    "imdbVotes": "876,575",
    "imdbID": "tt0499549",
    "Type": "movie",
    "Response": "True"
  }
];
// Only change code below this line
```

```
 "Released": "18 Dec 2009",
 "Runtime": "162 min",
 "Genre": "Action, Adventure, Fantasy",
 "Director": "James Cameron",
 "Writer": "James Cameron",
 "Actors": "Sam Worthington, Zoe Saldana, Sigourney Weaver, Stephen Lang",
 "Plot": "A paraplegic marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home.",
 "Language": "English, Spanish",
 "Country": "USA, UK",
 "Awards": "Won 3 Oscars. Another 80 wins & 121 nominations.",
 "Poster": "http://ia.mediaimdb.com/images/M/MV5BMTYwOTEwNjAzMl5BMl5BanBnXkFtZTcwODc5MTUwMw@@._V1_SX300.jpg",
 "Metascore": "83",
 "imdbRating": "7.9",
 "imdbVotes": "876,575",
 "imdbID": "tt0499549",
 "Type": "movie",
 "Response": "True"
 }];
// Only change code below this line
// Add your code below this line
var filteredList = watchList
 .map(function(e) {
 return { title: e["Title"], rating: e["imdbRating"] };
 })
 .filter(e => e.rating >= 8);
console.log(filteredList);
// Only change code above this line
console.log(filteredList);
```

var filteredList;
 	
// Only change code above this line
 	
console.log(filteredList);	

10) Implementing the filter method on a prototype

OLD CODE	NEW CODE

// The global variable
var s = [23, 65, 98, 5];
 	
Array.prototype.myFilter = function(callback){
  // Only change code below this line
  var newArray = [];
  // Only change code above this line
  return newArray;
 	
};
 	
var new_s = s.myFilter(function(item){
  return item % 2 === 1;
});
	let s = [23,65,98,5];
	Array.prototype.myFilter = function(callback) {
	 var newArray = [];
	 // Add your code below this line
	 for (let i = 0; i < this.length; i++) {
	    if (callback(this[i]) === true) {
	      newArray.push(this[i]);
	    }
	 }
	 // Add your code above this line
	 return newArray;
	};
	var new_s = s.myFilter(function(item) {
	 return item % 2 === 1;
	});

11) Returning part of an array using slice method

OLD CODE	NEW CODE

function sliceArray(anim, beginSlice, endSlice) {
  // Only change code below this line
	function sliceArray(anim, beginSlice, endSlice) {
	 // Only change code below this line
	 return anim.slice(beginSlice, e

// Only change code above this line
}
var inputAnim = ["Cat", "Dog", "Tiger", "Zebra", "Ant"];
sliceArray(inputAnim, 1, 3);
	ndSlice);
	// Only change code above this line
	}
	var inputAnim = ["Cat", "Dog", "Tiger", "Zebra", "Ant"];
	sliceArray(inputAnim, 1, 3);

12) Removing elements from an array using slice instead of splice

OLD CODE	NEW CODE

function nonMutatingSplice(cities) {
// Only change code below this line
return cities.splice(3);
 	
// Only change code above this line
}
var inputCities = ["Chicago", "Delhi", "Islamabad", "London", "Berlin"];
nonMutatingSplice(inputCities);
	function nonMutatingSplice(cities) {
	// Add your code below this line
	return cities.slice(0, 3);
	// Add your code above this line
	}
	var inputCities = ["Chicago", "Delhi", "Islamabad", "London", "Berlin"];
	nonMutatingSplice(inputCities);

13) Combining 2 arrays using concat method

OLD CODE	NEW CODE

function nonMutatingConcat(original, attach) {
// Only change code below this line
// Only change code above this line
}
var first = [1, 2, 3];
var second = [4, 5];
nonMutatingConcat(first, second);
	function nonMutatingConcat(original, attach) {
	// Add your code below this line
	return original.concat(attach);
	// Add your code above this line
	}
	var first = [1, 2, 3];
	var second = [4, 5];
	nonMutatingConcat(first, second);

14) Adding elements to the end of an array using concat method instead of push

OLD CODE	NEW CODE

function nonMutatingPush(original, newItem) {
  // Only change code below this line
  return original.push(newItem);
 // Only change code above this line
}
var first = [1, 2, 3];
var second = [4, 5];
nonMutatingPush(first, second);
	function nonMutatingPush(original, newItem) {
	 // Add your code below this line
	 return original.concat(newItem);
	 // Add your code above this line
	}
	var first = [1, 2, 3];
	var second = [4, 5];
	nonMutatingPush(first, second);

15) Using the reduce method to analyse data

OLD CODE	NEW CODE

// The global variable
var watchList = [
  {
    "Title": "Inception",
    "Year": "2010",
    "Rated": "PG-13",
    "Released": "16 Jul 2010",
    "Runtime": "148 min",
    "Genre": "Action, Adventure, Crime",
    "Director": "Christopher Nolan",
    "Writer": "Christopher Nolan",
   . . . . . .
  }
];
 	
function getRating(watchList){
  // Only change code below thi
	// the global variable
	var watchList = [
	 {
	 Title: "Inception",
	 Year: "2010",
	 Rated: "PG-13",
	 Released: "16 Jul 2010",
	 Runtime: "148 min",
	 Genre: "Action, Adventure, Crime",
	 Director: "Christopher Nolan",
	 Writer: "Christopher Nolan",
	   . . . . . .
	 }
	];
	function getRating(watchList) {
	 // Add your code below this line
	 const averageRating = watchList.reduce(({ sum, count }, {

s line
  var averageRating;
 	
  // Only change code above this line
  return averageRating;
}
console.log(getRating(watchList));
	Director: dir, imdbRating: rating }, idx, arr) => {
	 if (dir === 'Christopher Nolan') {
	 count++;
	 sum += Number(rating);
	 }
	 return idx === arr.length - 1
	 ? sum / count
	 : { sum, count };
	 }, { sum: 0, count: 0 });
	 // Add your code above this line
	 return averageRating;
	}
	console.log(getRating(watchList));

16) Using higher order functions - map/filter/reduce to solve complex problems

OLD CODE	NEW CODE

const squareList = (arr) => {
  // Only change code below this line
  return arr;
  // Only change code above this line
};
 	
const squaredIntegers = squareList([-3, 4.8, 5, 3, -3.2]);
console.log(squaredIntegers);
	const squareList = arr => {
	 return arr.reduce((sqrIntegers, num) => {
	 return Number.isInteger(num) && num > 0
	 ? sqrIntegers.concat(num * num)
	 : sqrIntegers;
	 }, []);
	};

17) Sorting an array alphabetically using the sort method

OLD CODE	NEW CODE

function alphabeticalOrder(arr) {
  // Only change code below thi
	function alphabeticalOrder(arr)
	{
	 // Add your code below this lin

s line
 	
// Only change code above this line
}
alphabeticalOrder(["a", "d", "c", "a", "z", "g"]);
	e
	return arr.sort(function(a, b) {
	return a === b ? 0 : a < b ? -1 : 1;
	});
	// Add your code above this line
	}
	alphabeticalOrder(["a", "d", "c", "a", "z", "g"]);

18) Returning a sorted array without changing the original

OLD CODE	NEW CODE

var globalArray = [5, 6, 3, 2, 9];
function nonMutatingSort(arr) {
// Only change code below this line
 	
// Only change code above this line
}
nonMutatingSort(globalArray);
	var globalArray = [5, 6, 3, 2, 9];
	function nonMutatingSort(arr) {
	// Add your code below this line
	return [].concat(arr).sort(function(a, b) {
	return a - b;
	});
	// Add your code above this line
	}
	nonMutatingSort(globalArray);

19) Splitting a string into an array using the split method

OLD CODE	NEW CODE

function splitify(str) {
// Only change code below this line
 	
// Only change code above this line
}
splitify("Hello World,I-am code");
	function splitify(str) {
	// Add your code below this line
	return str.split(/\W/);
	// Add your code above this line
	}
	splitify("Hello World,I-am code");

20) Combining an array into a string using the join method

OLD CODE	NEW CODE

function sentensify(str) {
  // Only change code below this line
  // Only change code above this line
}
sentensify("May-the-force-be-with-you");
	function sentensify(str) {
	// Add your code below this line
	 return str.split(/\W/).join(" ");
	// Add your code above this line
	}
	sentensify("May-the-force-be-with-you");

21) Converting strings to URL slugs

OLD CODE	NEW CODE

// Only change code below this line
function urlSlug(title) {
 	
}
// Only change code above this line
	// the global variable
	var globalTitle = "Winter Is Coming";
	// Add your code below this line
	function urlSlug(title) {
	 return title
	 .toLowerCase()
	 .trim()
	 .split(/\s+/)
	 .join("-");
	}
	// Add your code above this line
	var winterComing = urlSlug(globalTitle); // Should be "winter-is-coming"

22) Using the every method to check that every element in the array meets a criteria

OLD CODE	NEW CODE

function checkPositive(arr) {
  // Only change code below this line
 	
  // Only change code above thi
	function checkPositive(arr) {
	 // Add your code below this line
	 return arr.every(function(value) {

s line
}
checkPositive([1, 2, 3, -4, 5]);
	return value > 0;
	});
	// Add your code above this line
	}
	checkPositive([1, 2, 3, -4, 5]);

23) Using the some method to check that any elements in the array meet a criteria

OLD CODE	NEW CODE

function checkPositive(arr) {
// Only change code below this line
 	
// Only change code above this line
}
checkPositive([1, 2, 3, -4, 5]);
	function checkPositive(arr) {
	return arr.some(elem => elem > 0);
	}
	checkPositive([1, 2, 3, -4, 5]);

24) Introduction to currying and partial application

OLD CODE	NEW CODE

function add(x) {
// Only change code below this line
 	
// Only change code above this line
}
add(10)(20)(30);
	function add(x) {
	// Add your code below this line
	return y => z => x + y + z;
	// Add your code above this line
	}
	add(10)(20)(30);