

Jenni Diaz

14.4 Exercise

1. ES5 Function that takes a variable number of arguments:

```
function filterOutOdds()  
{  
  var nums = Array.prototype.slice.call(arguments);  
  return nums.filter(function(num)  
  
    {return num % 2 === 0});  
}
```

2. ES2015 Function that takes a variable number of arguments:

```
const filterOutOdds = (...args) => args.filter(v => v % 2 === 0)
```

3. findMin:

```
const findMin = (...args) => Math.min(...args)
```

```
findMin(1,4,12,-3) // -3
```

```
findMin(1,-1) // -1
```

```
findMin(3,1) // 1
```

4. mergeObjects:

```
const mergeObjects = (obj1, obj2) => ({...obj1, ...obj2})
```

```
mergeObjects({a:1, b:2}, {c:3, d:4}) // {a:1, b:2, c:3, d:4}
```

5. doubleAndReturnArgs:

```
const doubleAndReturnArgs = (arr, ...args) => [...arr, ...args.map(v => v *2)]
```

```
doubleAndReturnArgs([1,2,3],4,4) // [1,2,3,8,8]
```

```
doubleAndReturnArgs([2],10,4) // [2, 20, 8]
```

Slice and Dice:

```
const removeRandom = items =>
```

```
{  
  let idx = Math.floor(Math.random() * items.length);  
  return [...items.slice(0, idx), ...items.slice(idx + 1)];}
```

```
const extend = (array1, array2) =>
```

```
{  
  return [...array1, ...array2];  
}
```

```
const addKeyVal = (obj, key, val) =>
```

```
{  
  let newObj = { ...obj }  
  newObj[key] = val;  
  return newObj;}
```

```
const removeKey = (obj, key) =>
```

```
{  
  let newObj = { ...obj }  
  delete newObj[key]  
  return newObj;  
}
```

```
let newObj = { ...obj }  
newObj[key] = val;  
return newObj;
```