

ES6

ARRAYS

Filtering even/odd numbers

Use *filter* to compute a subset of array

```
let all = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
let even = all.filter((i) => { return i % 2 === 0 })
```

```
let odd = all.filter(i => i % 2 !== 0)
```

Note syntax difference/equivalence!

Return *true* to keep, *false* to skip item

Filter with Old Function Notation

ES6 Arrow functions are less verbose

```
let all = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

ES6:

```
let even = all.filter(i => i % 2 === 0)
```

ES5:

```
let even = all.filter(function (i) {return i % 2 === 0})
```

Careful if you need additional lines!

- ES6:

```
let even = all.filter(i => {  
    console.log(i)  
    return i % 2 === 0})
```

Implicit returns only work if it's one line. You'll need explicit return if body has more than one line

- ES5:

```
let even = all.filter(function (i) {  
    console.log(i)  
    return i % 2 === 0})
```

Mapping

- Use *map* to collate results into new array

```
let all = [1, 2, 3, 4]
```

```
let square = all.map(i => i * i)
```

```
// square = [1, 4, 9, 16]
```

Finding & Filtering

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

```
array.find(x => x > 3)
```

// => 4

```
array.findIndex(x => x > 3)
```

// => 2

```
array.filter(x => x > 3)
```

// => [4, 5]

STRING INTERPOLATION

Jose Annunziato

Use interpolation instead of concatenation

```
const customer = { name: "Alice" }
```

```
const card = { amount: 7, product: "Bar", unitprice: 42 }
```

```
let message = `Hello ${customer.name},
```

```
  want to buy ${card.amount} ${card.product} for
```

```
  a total of ${card.amount * card.unitprice} bucks?`
```

```
console.log(message)
```