COMP6080
Web Front-End Programming

Javascript (General)
Map, Reduce, Filter

Map, Reduce, Filter

- **Map**: creates a new list with the results of calling a provided function on every element in the given list
- **Reduce**: executes a **reducer** function (that you provide) on each member of the array resulting in a single output value
- **Filter**: creates a new array with all elements that pass the test implemented by the provided function

Map

Map: creates a new array with the results of calling a provided function on every element in the calling array

map.js

```
const items = [1, 2, 3];

const makeSquare = (i) => {
  return {
    item: i,
    square: i*i,
  };
};

const itemsAndSquares = items.map(makeSquare);

console.log(itemsAndSquares);
```

Reduce

Reduce: executes a **reducer** function (that you provide) on each member of the array resulting in a single output value

reduce.js

```javascript
const items = [1, 2, 3];

const getSum = (total, num) => {
  return total + num;
};

const totalItems = items.reduce(getSum, 0);

console.log(totalItems);
```

Filter

Filter: creates a new array with all elements that pass the test implemented by the provided function

filter.py

```
const items = [1, 2, 3];

const onlyEven = (i) => {
  if (i % 2 == 0) return true;
  return false;
};

const evenItems = items.filter(onlyEven);

console.log(evenItems);
```

Combined

allthree.py

```python
from functools import reduce

if __name__ == '__main__':
    marks = [ 39, 43.2, 48.6, 24, 33.6 ] # Marks out of 60
    normalised_marks = map(lambda m: 100*m/60, marks)
    passing_marks = list(filter(lambda m: m >= 50, normalised_marks))
    total = reduce(lambda a, b: a + b, passing_marks)
    average = total/len(passing_marks)
    print(average)
```