Problem solving on Functional Programming

- At the end of this class, students will be able to-
  - Problems for functional programming

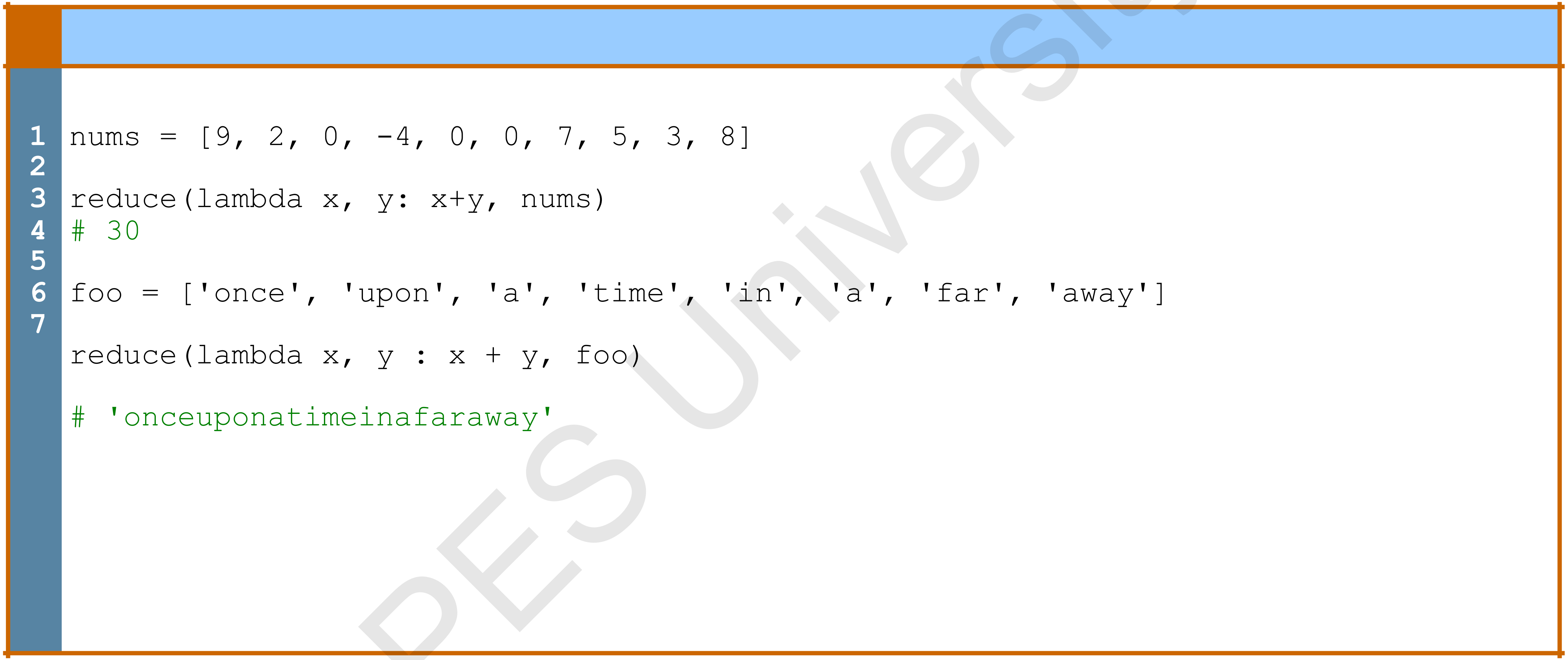
```
map(function, iterable, ...)
```

- Map applies **function** to each element of **iterable** and creates a list of the results
- You can optionally provide more iterables as parameters to map and it will place tuples in the result list
- Map returns an iterator which can be cast to list

```
nums = [0, 4, 7, 2, 1, 0 , 9 , 3, 5, 6, 8, 0, 3]

nums = list(map(lambda x : x % 5, nums))

print(nums)
#[0, 4, 2, 2, 1, 0, 4, 3, 0, 1, 3, 0, 3]

def even (x):
    if (x % 2 == 0):
            return "even"
    else:
            return "odd"

list (map(even, nums))
#['even', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'odd', 'odd', 'even', 'even', 'even', 'odd']
```

`reduce(`**`function, iterable`**`[,`**`initializer`**`])`

- Reduce will apply **function** to each element in **iterable** along with the sum so far and create a cumulative sum of the results
- **function** must take two parameters
- If initializer is provided, initializer will stand as the first argument in the sum
- Unfortunately in python 3 reduce() requires an import statement
  - `from functools import reduce`



```
nums = [9, 2, 0, -4, 0, 0, 7, 5, 3, 8]

reduce(lambda x, y: x+y, nums)
# 30

foo = ['once', 'upon', 'a', 'time', 'in', 'a', 'far', 'away']

reduce(lambda x, y : x + y, foo)

# 'onceuponatimeinafaraway'
```

```
numlists = [[1, 2, 3], [4, 5], [6, 7, 8, 9]]
reduce(lambda a, b: a + b, numlists, [])
# [1, 2, 3, 4, 5, 6, 7, 8, 9]

nums = [1, 2, 3, 4, 5, 6, 7, 8]
nums = list(reduce(lambda x, y : (x, y), nums))
print(nums)        #(((((((1, 2), 3), 4), 5), 6), 7), 8)
```

Goal: given a list of numbers I want to find the average of those numbers in a few lines using `reduce()`

For Loop Method:

- sum up every element of the list
- divide the sum by the length of the list

Solution

```
nums = [92, 27, 63, 43, 88, 8, 38, 91, 47, 74, 18, 16,
        29, 21, 60, 27, 62, 59, 86, 56]

sum = reduce(lambda x, y : x + y, nums) / len(nums)
```

Framework for processing huge datasets on certain kinds of distributable problems

Map Step:

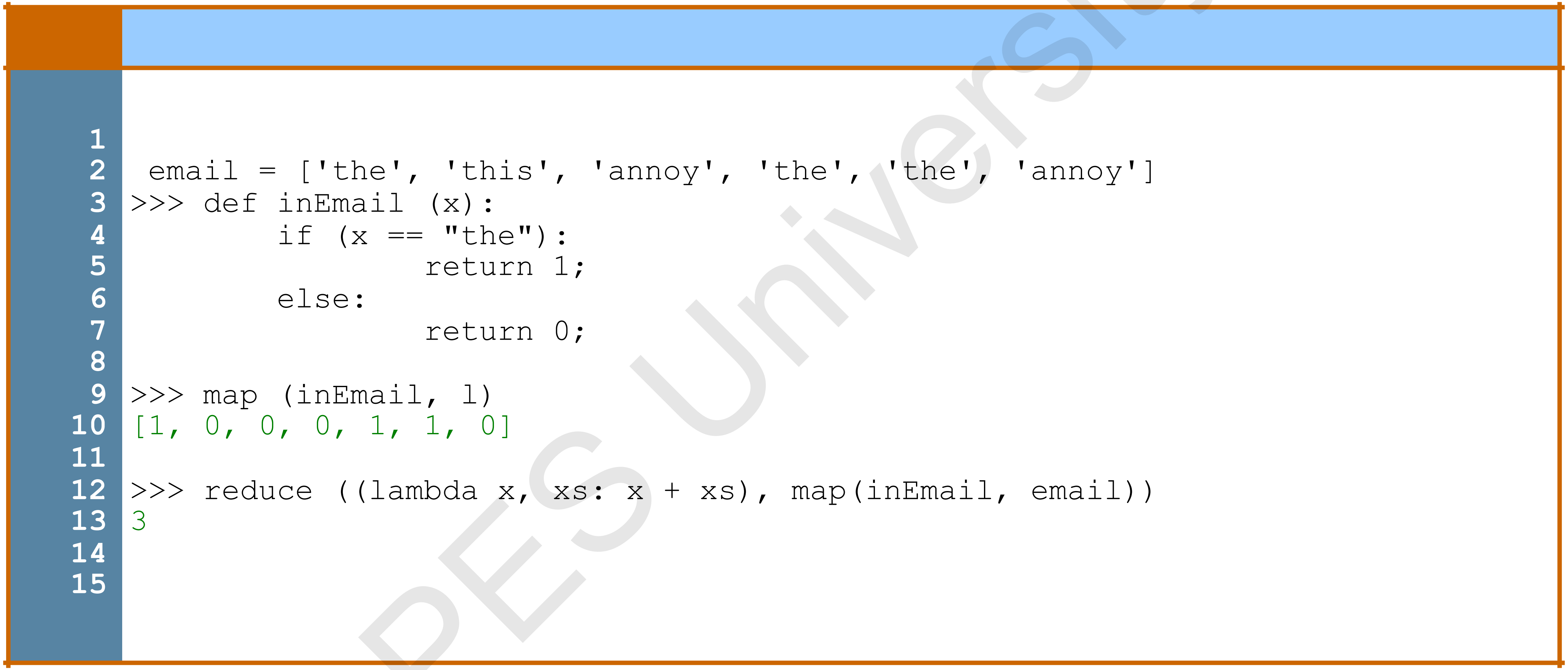
- master node takes the input, chops it up into smaller sub-problems, and distributes those to worker nodes.
- worker node may chop its work into yet small pieces and redistribute again

Reduce Step:

- master node then takes the answers to all the sub-problems and combines them in a way to get the output

Problem: Given an email how do you tell if it is spam?

- Count occurrences of certain words. If they occur too frequently the email is spam.



```
 email = ['the', 'this', 'annoy', 'the', 'the', 'annoy']
>>> def inEmail (x):
        if (x == "the"):
                return 1;
        else:
                return 0;

>>> map (inEmail, l)
[1, 0, 0, 0, 1, 1, 0]

>>> reduce ((lambda x, xs: x + xs), map(inEmail, email))
3
```

- find the smallest string from the list of strings

```
string = input("Enter a string : ").split()
print(string)
small = min(string, key = lambda x : len(x))
print("The smallest string is ",small)
```

Output:

```
Enter a string : aaaaa aaaa bbb cc
['aaaaa', 'aaaa', 'bbb', 'cc']
The smallest string is  cc
```

- find all strings ending with a given word

```
string = input("Enter a string : ").split(" ")
print(string)
letter = input("Enter a letter: ")
l = list(filter(lambda x: x.endswith(letter), string))
print("Strings ending with suffix", letter, " are ", l)
```

Output:

```
Enter a string : aaao bbbo ccco asdd ghhdff hjj
['aaao', 'bbbo', 'ccco', 'asdd', 'ghhdff', 'hjj']
Enter a letter: o
Strings ending with suffix o  are  ['aaao', 'bbbo', 'ccco']
```

- find the average length of the string - use reduce to find the total length.

```
import functools
string = input("Enter a string : ").split(" ")
print(string)
length = list(map(len,string))
avg = (functools.reduce(lambda x,y: x + y ,length)) /len(length)
print("Average length of a string is ", avg)
```

Output:

```
Enter a string : hello how are you
['hello', 'how', 'are', 'you']
Average length of a string is  3.5
```

- Consider the list
- details=[(“Python”, “A”,(23,34,19)),

```
              (“Chemistry”, “B”,(12,23,25)),
              (“Maths”, “C”, (15,34,12))
             ]
```

- Sort the list on the basis of subject name

```
details=[("Python", "A",(23,34,19)), ("Chemistry", "B",(12,23,25)),  ("Maths", "C",
(15,34,12)) ]
subject = sorted(details)
print("Student list is sorted based on 1st field, Subject:\n", subject)
details.sort(reverse=True)
print("Student list is sorted based on 1st field, Subject:\n", details)
```

Output:

```
Student list is sorted based on 1st field, Subject:
[('Chemistry', 'B', (12, 23, 25)), ('Maths', 'C', (15, 34, 12)), ('Python', 'A', (23, 34, 19))]
Student list is sorted based on 1st field, Subject:
[('Python', 'A', (23, 34, 19)), ('Maths', 'C', (15, 34, 12)), ('Chemistry', 'B', (12, 23, 25))]
```

- Sort the list on the basis of highest total marks for the section

```
details=[("Python", “B",(23,34,19)), (" Python ", “C",(12,23,25)),  (" Python ", “A",
(15,34,12)) ]
highest_total= sorted(details, reverse = True, key = lambda t: sum(t[2]))
print("details is sorted based on total of marks in descending order:\n", highest_total)
```

```
Output:
details is sorted based on total of marks in descending order:
[('Python', 'B', (23, 34, 19)), ('Python', 'A', (15, 34, 12)), ('Python', 'C', (12, 23, 25))]
```

Examples

- Using the filter function, define a filterPositive function that takes a list of numbers and returns a list of its positive elements.

- Using the filter function, define a filterSameLength function that takes a string st and a list of strings and returns all the strings in the list that have the same length as st.