Python:
Map — Filter — Reduce

CS 5010
Prof. Nada Basit

Traditional Function / Method

- In Python, a traditional method header begins with the key word “`def`” then the method name along with any parameters within parenthesis, followed by a colon “ : ”
- The body of this method begins on the next line and is indented to the right

-
  ```
  def square(number):
     ‘‘‘ Returns the square of number ’’’
     return  number * number
  ```

Anonymous Function Using Lambda

- Can create functions “on the fly” by writing anonymous functions
- Why anonymous?
  - It doesn’t have a name
  - Based upon the lambda form
- Functions are objects – an anonymous function can only consist of an expression
- How to use it?   Simply use the variable name

```
square = lambda x: x*x # parameter : expression
print(square(2))   # to use, use the variable name
and pass a parameter
```

Anonymous Function & Map Function

- One place an anonymous function is used is with the map higher-order function (*more on higher-order functions next*)

- Example:

```
numbers = [1,2,3,4]
print(numbers)
numbsq = list(map(lambda x: x*x, numbers))
print(numbsq)  # Output:  [1, 4, 9, 16]
```

- Comment: save time by not having to define square function – don't "on the fly" using lambda

Higher-order Functions

- **Map**
  - ○ Takes a function and applies it to ALL of the elements of a given sequence
- **Filter**
  - ○ Takes a function, often a Boolean function, and returns only those elements of the sequence that meet the criteria (that are TRUE when passed to the function)
- **Reduce**
  - ○ Applies a function on two arguments cumulatively to the items of an iterable object (a sequence such as a list) so as to reduce the iterable object to a single value
  - ○ Need: `import functools`

Map Example

Given these 3 functions:

```
def square(number):
   return number * number

def sum(x,y):
   return x + y
```

```
def even(number):
   if number % 2 == 0:
      return True
   else:
      return False
```

```
numbers = [1,2,3]
print(numbers)    # output:  [1, 2, 3]
# mapping the square function:
numberssq = list(map(square, numbers))
print(numberssq)  # output: [1, 4, 9]
```

Filter Example

Given these 3 functions:

```
def square(number):
    return number * number

def sum(x,y):
    return x + y
```

```
def even(number):
    if number % 2 == 0:
        return True
    else:
        return False
```

```
numbers = list(range(1,11))  # numbers 1-10 in a list
print(numbers)# output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
# filtering using the even function:
evens = list(filter(even, numbers))
print(evens)        # output: [2, 4, 6, 8, 10]
```

Reduce Example

Given these 3 functions:

```
def square(number):
    return number * number

def sum(x,y):
    return x + y
```

```
def even(number):
    if number % 2 == 0:
        return True
    else:
        return False
```

```
import functools
numbers = list(range(1,11)) # numbers 1-10 in a list
print(numbers) # output: [1, 2, 3, …, 10]
# reducing using the sum function:
sum = functools.reduce(sum, numbers)
print("The sum of the range is " + str(sum)) # 55
```

pyScript15.py

- Look at pyScript15.py for another example