

Introduction to programming, Lesson 5: functions and numpy

Functions – 2

1. Multiple values

A function can return several values at a time.

```
def division(a, b):
    q = a // b
    r = a % b
    return q, r    #returns two values

q, r = division(22, 5)
print "quotient = ", q, ", remainder = ", r
```

2. Docstring

The docstring allows to understand easily what your function does.

```
def division(a, b):
    '''takes two integers a and b as arguments and
    returns the quotient and the remainder of
    the Euclidean division of a by b
    '''
    q = a // b
    r = a % b
    return q, r
```

In the project, you should have a docstring for all your functions. All the functions from standard Python modules have the docstring. You can view it as follows. (Press q to exit the viewer.)

```
from random import *
help(randint)
```

Numpy

1. Array creation

In the following example, you will first create a Python list, and then a numpy array out of the newly created list.

```
# Import the numpy package as np
import numpy as np

# Create a new list height
height = [1.87, 1.87, 1.82, 1.91, 1.90, 1.85]

# Create a numpy array from height
np_height = np.array(height)
print np_height
```

Arrays can have more than one dimension. Try to create an array out of list of lists, like `[[1,1], [2,4], [3,7]]`.

2. Pyramid of ages: Array creation

Download a file `age-2013.txt` from Moodle, Lesson 5. This file was generated by INSEE and contains demographic statistics for France. First line contains categories, gender and age, separated by commas. Each of the subsequent lines contains a name of the region and the number of inhabitants of each region that belong to each category. Your task is to write a function that reads the file and returns a tuple of four numpy arrays: `arr_categories`, `arr_regions`, `arr_men`, `arr_women`. The types of data for arrays `arr_categories` and `arr_regions` must be `str`, the types of data for arrays `arr_men`, `arr_women` must be `int`.

```
import numpy as np
def read_data(filename):
```

Introduction to programming, Lesson 5: functions and numpy

```
lines = open(filename).readlines()
#Your code goes here
return (arr_categories, arr_regions, arr_men, arr_women)
```

3. Some statistics on arrays

We have many unary operations, such as computing the sum of all the elements in the array.

```
import numpy as np
weight_kg = np.array([81.65, 97.52, 95.25, 92.98, 86.18, 88.45])
print weight_kg.sum() #sum of all values
print weight_kg.max() #min value
print weight_kg.size() #size
```

For two-dimensional arrays, it is possible to compute statistics by rows or by columns.

```
import numpy as np
weight_height = np.array([[180,81.65], [160,97.52], [200,95.25]])
print weight_height.max(axis=0) #max by columns
print weight_height.max(axis=1) #max by rows
```

4. Pyramid of ages: Population by region

Write a function that returns an array `total`. Each value must contain the total number of inhabitants of the corresponding region in float. To convert elements of an array from int to float, use `astype(float)`.

```
def total_by_region(arr_women, arr_men):
    #Your code goes here
    total = total.astype(float)
    return total
```

5. Element-wise operations

Arithmetic operators, math functions, and Boolean operators on arrays apply *element-wise*. A new array is created and filled with the result.

```
import numpy as np
a = np.array( [20,30,40,50] )
b = np.array([0, 1, 2, 3])
print a-b
print a*b
print 10*np.sqrt(a)
print a<35
```

6. Pyramid of ages: statistics by region

Write a function `percentage_by_region` that computes two arrays, percentage of men in each region and percentage of women in each region.

7. Subsetting

Another great feature of numpy arrays is the ability to subset. For instance, if you wanted to know which weights in our `weights_kg` array are above 90, we could quickly subset it to find out.

```
import numpy as np
weight_kg = np.array([81.65, 95.25, 92.98, 86.18,97.52,88.45])
over_90 = weight_kg > 90
print over_90, weight_kg[over_90]
```

8. Pyramid of ages: Percentage of men and women by region

Find the most masculine and the most feminine region using subsetting.

9. Dot product

Imagine you are running a bagel shop. You have several recipes of bagels, and for each you

Introduction to programming, Lesson 5: functions and numpy

store the necessary ingredients in a table (in kg). You also know price of each ingredient: cheese is 15€/kg, ham is 40€/kg, and tomato is 5€/kg. Your task is to compute the price of each bagel.

	Cheese	Ham	Tomato
Bagel 1	0.1	0.1	0.05
Bagel 2	0.05	0.1	0.1
Bagel 3	0.2	0	0.1

Price of Bagel 1 is $15*0.1 + 40*0.1 + 5*0.05 = 5.75$. Price of Bagel 2 is $15*0.05 + 40*0.1 + 5*0.1 = 5.25$. Price of Bagel 3 is $15*0.2 + 40*0 + 5*0.1 = 3.5$. We can write the prices as one array $[5.75, 5.25, 3.5]$. We say that this array is a result of a dot product of the array of ingredients and the array of prices.

```
import numpy as np
ingredients = np.array([[0.1, 0.1, 0.05], [0.05, 0.1, 0.1], [0.2, 0, 0.1]])
prices_ingredients = np.array([15, 40, 5])
prices_bagels = np.dot(ingredients, prices_ingredients)
```

10. Pyramid of ages: Average age by region

Assuming that the age of each person in group 0-19 is 10, in group 20-39 is 30, in group 40-59 is 50, in group 60-74 is 67, and in group 70+ is 80, compute the average age in each region. Find regions with highest and smallest average ages.