

IN-CLASS WORK: PYTHON — LISTS, ARRAYS & REPETITIONS

1. Lists & Arrays Intro: Containers

Read §2.4.1 and as practice try Read §2.4. intro on pages 46 and 47 of Newman book.

2. Lists

Read §2.4.1 and as practice try for page 48 the following commands:

```
r=[1,1,2,3,8]
print(r)
```

and also

```
x=1.0
y=1.5
z=-2.2
r=[x,y,z]
print(r)
y=2.0
print(r)
r=[x,y,z]
print(r)
```

for pages 49 and 50 try the commands

```
a=[3,2.5,4,"hi",-3]
print(a)
print(a[0],a[1],a[3])
a[1]=7.3
print(a)
```

and for page 51 try

```
from math import log,exp
r=[1.0,1.5,2.2]
logr=list(map(log,r))
print(logr)
```

and for pages 51 and 52 test

```
r=[1.0,1.5,2.2,9.1]
print(r)
r.append(2.8)
print(r)
r.pop()
print(r)
r.pop(1)
print(r)
print(r[2])
```

3. Arrays

Read §2.4.2 pages 53 and 54. We will use again, `scipy`, which includes `numpy`, so the commands on page 54 are replaced by

```
import scipy as sp
a=sp.zeros(4,float)
print(a)
```

and try also

```
b=sp.zeros([2,3],int)
print(b)
```

for pages 55 and 56 try

```
d=sp.array([[1,2,3],[4,5,6]])
print(d)
d[1,2]=30
print(d)
```

Continue reading, so read §2.4.3. Notice that `loadtxt` allows us to read data from a file, so we do not have to type in by hand each value. Get a feel for how it works, by using in your linux terminal `gedit` (or any other editor you like) to make the file named `values.txt` with the content as the four last lines on page 57. Then you can use `jupyter` (or a python script which includes the line `import scipy as sp`) to test the python commands

```
e=sp.loadtxt("values.txt",float)
print(e)
```

In case this made you curious about how to also write into a file of a specified name, you may have a look at `~kvollmay/share.dir/pythonsamples.dir/sample_inout.py`

In case we are short on time, you may skip §2.4.4 and §2.4.5. In case you have plenty of time, read §2.4.4 and §2.4.5. No need yet for testing the commands.

4. For Loops

In §2.5 you learn another tool to repeat the same set of commands over and over again. While you read §2.5 try on page 67 the commands

```
r=[1,3,4]
for n in r:
    print(n)
    print(2*n)
    print("")
print("Finished")
```

continue reading pages 68 and 69 and for page 70 try

```
for n in range(5):
    print(n**2)
```

Continue reading §2.5 until page 73.