

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

### 15. Decisions

Read §2.3.1 and as you read try the commands on page 39

```
x=int(input("Enter a whole number no greater than ten: "))
if x > 10:
    print("You entered a number greater than ten.")
    print("Let me fix that for you.")
    x=10
print("Your number is",x)
```

and the commands at the bottom of page 41

```
x=int(input("Enter a whole number no greater than ten: "))
if x>10:
    print("your number is great than ten.")
elif x>9:
    print("Your number is OK, but you're cutting it close.")
else:
    print("Your number is fine. Move along.")
```

Continue reading §2.3.2 and try the commands

```
x=int(input("Enter a whole number no greater than ten: "))
while x>10:
    print("This is greater than ten. Please try again.")
    x=int(input("Enter a whole number no greater than ten: "))
print("Your number is",x)
```

Continue reading §2.3.3 (The third line needs the correction “is not met” to instead “is met”.) but stop reading on page 45 before you get to the commands `f1=1` and following lines. Put the book aside and try yourself to write a program which determines the Fibonacci numbers. When you succeeded, continue reading on page 45.

Try the commands on top of the page 46

```
f1,f2=1,1
while f1<=1000:
    print(f1)
    f1,f2=f2,f1+f2
```

and try to understand how exactly this program works. If you google “Fibonacci Nature” you get some beautiful examples and explanations for the occurrence of fibonacci sequence in nature.

### 16. Repetitions Sample Program

To practice how repetition commands work, copy into your working directory the sample program

```
~kvollmay/share.dir/pythonsamples.dir/sample_repetitions.py
```

Look at this program and before running this program, lookup online (or remind yourself) the used commands. Still before running this program, predict what exactly the program will print on the screen when you run the program. After your prediction, run the program and compare with your prediction.

## 17. Factorial

Write a program which reads in  $N$  and prints out  $N!$ . (Use a repetition command.)

## 18. Lists & Arrays Intro: Containers

For examples how to use arrays copy into your working directory the sample program

```
~kvollmay/share.dir/pythonsamples.dir/sample_arrays.py
```

To learn more about lists and arrays, scan this sample file and then read in Newman as described in the following steps.

### 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)
```

On page 49 and all following pages of this section, replace the `math` package with the `numpy` package the same way as in last class. For example `sqrt(3.0)` is instead `np.sqrt(3.0)`  
In addition 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

```
import numpy as np
r=[1.0,1.5,2.2]
logr=list(map(np.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])
```

## Arrays

Read §2.4.2 pages 53 and 54. We will also use numpy, but import differently so the commands on page 54 are replaced by

```
import numpy as np
a=np.zeros(4,float)
print(a)
```

and try also

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

for pages 55 and 56 try

```
d=np.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 a python script which includes the line `import numpy as np` to test the python commands

```
e=np.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`