

Python workshop

Week 5: Dictionaries and tuples

barbera@van-schaik.org



Overview of this workshop series

- Week 1: Writing your first program
- Week 2: Make choices and reuse code
- Week 3: Loops and strings
- Week 4: Files and lists
- Week 5: Dictionaries and tuples

Acknowledgments: Structure of the workshop follows the book
“Python for informatics” by Charles Severance.
Several examples are from this book or the accompanying slides.

Last weeks exercises

Assignment 1

- How many lines with the word “true” and how many with “false” in plato.txt?
- Hints
 - Open file
 - Make two variables to count “true” and “false”
 - Use the string method: find



assignment1.py

Assignment 2

- Open the file “hobbies.txt” and print the names of the persons
- Hints:
 - Open file
 - “split” the lines
 - Get the right column and print it

assignment2.py



Bonus exercise – guessing game

- Let the user think of a number between 1 and 1000
- The computer makes guesses
- The user gives hints: higher/lower (or h/l)

One solution: let computer guess all the numbers between 1 and 1000 (not so efficient)

How would you solve this?

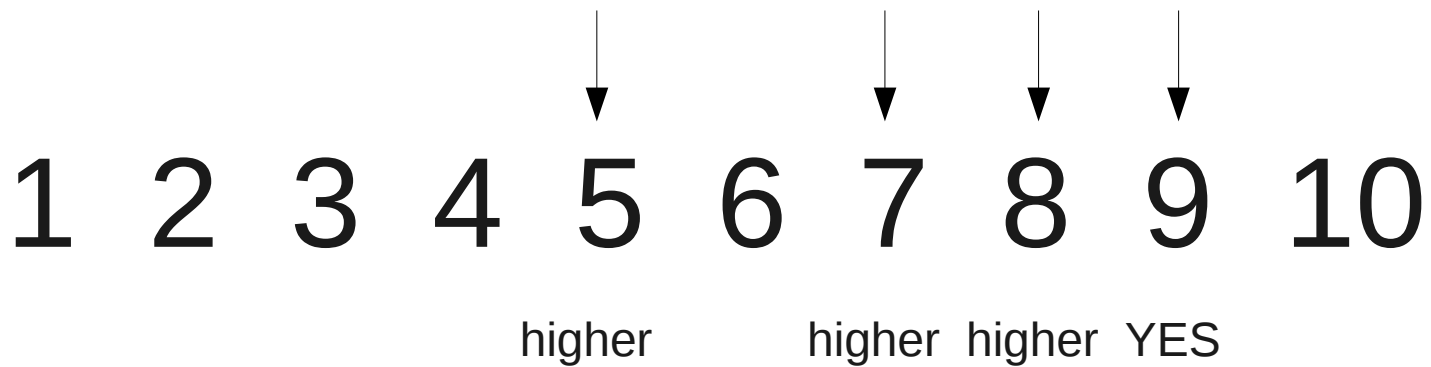
With code or as a concept

guessing-game3.py



Guessing game - concept

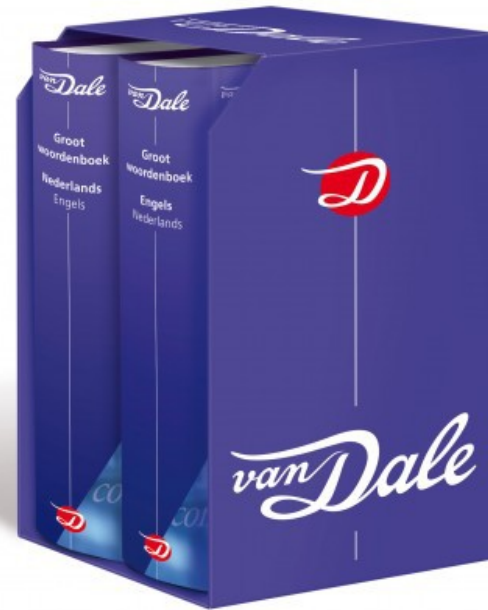
User thinks of number 9



Dictionaries

What is a dictionary?

- Key-value pairs
 - Een → One
 - Twee → Two
 - Drie → Three



```
>>> n12en = dict()  
>>> print n12en  
{}
```

Put data in dictionary

```
>>> nl2en = dict()
>>> nl2en["een"] = "one"
>>> nl2en["twee"] = "two"
>>> nl2en["drie"] = "three"
>>> print nl2en
```



```
>>> altdict = {'een':'one', 'twee':'two', 'drie':'three'}
>>> print altdict
```



Order is not the same as entered
Lists are ordered, dictionaries are not

Get data out

```
>>> print nl2en["twee"] # shows value
```

```
>>> print nl2en["hallo"] # does not exist: error
```

```
>>> for key in nl2en:
```

```
>>>     print key, nl2en[key]
```



Functions and operations

```
>>> len(n12en) # get nr of key-value pairs
```

```
>>> 'een' in n12en # True
```

```
>>> 'one' in n12en # False
```

```
>>> 'one' in n12en.values() # True
```

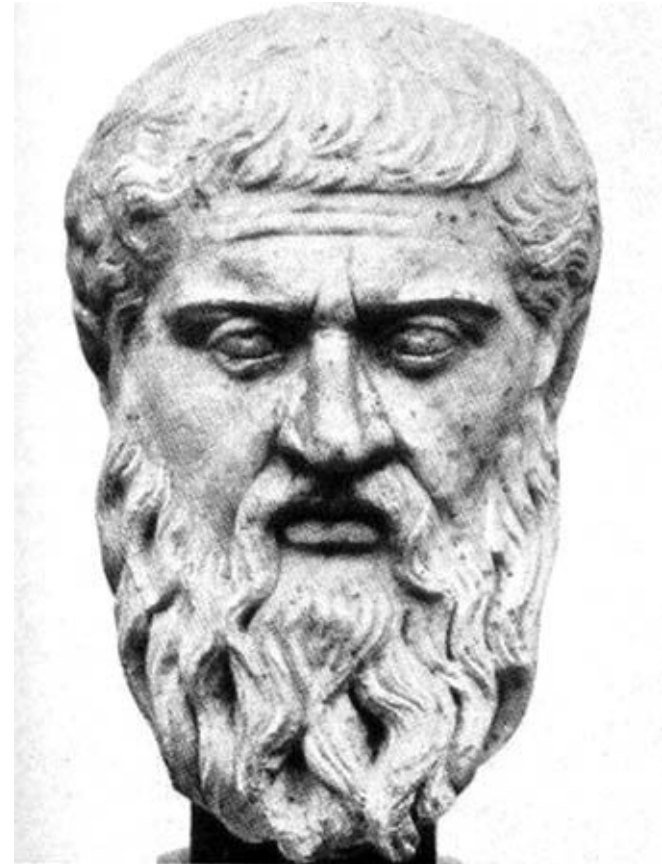
<https://docs.python.org/2.7/library/stdtypes.html#typesmapping>

Count stuff

```
import sys

myfile = "plato.txt"

try:
    fh = open(myfile, "r")
except:
    sys.exit("cannot open file")
```



Count stuff

```
# Go through file, get the words and count the words
d = dict()
for line in fh:
    line = line.rstrip() # remove newline
    line = line.lower() # make everything lowercase
    words = line.split() # split the line on tab or space

    for word in words:
        d[word] = d.get(word, 0) + 1
```

count-words.py

Count stuff

```
print "Show 10 words and their frequency"
i = 0
for key in d:
    if i >= 10:
        break

    print key, d[key]
    i += 1
```

Count stuff

```
print "Show the TOP 10 words and their frequency"
i = 0
for key in sorted(d, key=d.get, reverse=True):
    if i >= 10:
        break

    print key, d[key]
    i += 1
```


Tuples

What is a tuple?

- A tuple is like a list, but immutable

```
>>> t = tuple()
>>> print t
()
```



Tuples



```
>>> t = 'a', 'b', 'c', 'd', 'e' # tuple is comma separated
>>> t = ('a', 'b', 'c', 'd', 'e') # clearer way to do this
>>> tup = ('a',) # single value needs to have a comma
```

Tuple assignments

- Need to assign all values at once
- It is possible to replace an entire tuple

```
>>> t = ('a', 'b', 'c', 'd', 'e')
```

```
>>> t[0] = 'bla' # not possible
```

```
>>> t.append('bla') # also not possible
```

```
>>> t = ('bla',) + t # this is possible
```

Compare tuples



```
>>> (0, 1, 2) < (0, 3, 4) # True
```

```
>>> (0, 1, 2000000) < (0, 3, 4) # True?
```

```
>>> (10, 1, 2) < (0, 3, 4) # False
```

It starts at the first elements and stops once it is not True anymore



<



Sort on first element, in case of a tie try second element

```
txt = 'but soft what light in yonder window breaks'
```

```
words = txt.split()
```

```
t = list()
```

```
for word in words:
```

```
    t.append((len(word), word))
```

```
t.sort(reverse=True)
```

```
res = list()
```

```
for length, word in t:
```

```
    res.append(word)
```

```
print res
```

This is a tuple in a list



The comma-separated *length, word*
is a tuple as well

Example from book
tuple-sort.py

Multiple assignments at once

```
>>> x, y = 'hello', 'there'  
>>> print x  
>>> print y
```



```
>>> x, y = y, x # swap values
```

```
>>> mail = 'barbera@van-schaik.org'
```

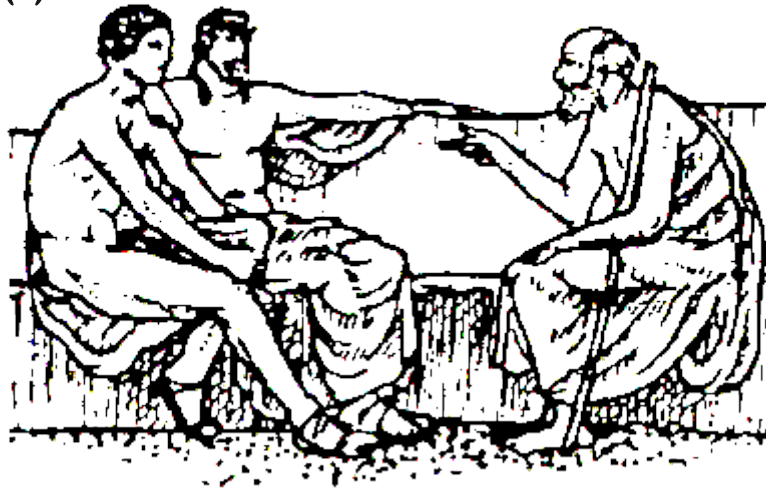
```
>>> name, domain = mail.split('@')
```



Dictionaries and tuples

```
>>> d = { 'socrates': 10, 'gorgias': 5, 'polus': 2}
>>> t = d.items()
>>> print t

>>> t.sort()
>>> print t
```



Multiple assignments in a script

```
for key, value in d.items():
    print key, value
```

Strings, lists, dictionaries, tuples

- String: immutable, iterate over characters

```
>>> s = "hooray"
```

- List: mutable, ordered, index is a number

```
>>> l = ['aap', 'noot', 'mies']
```

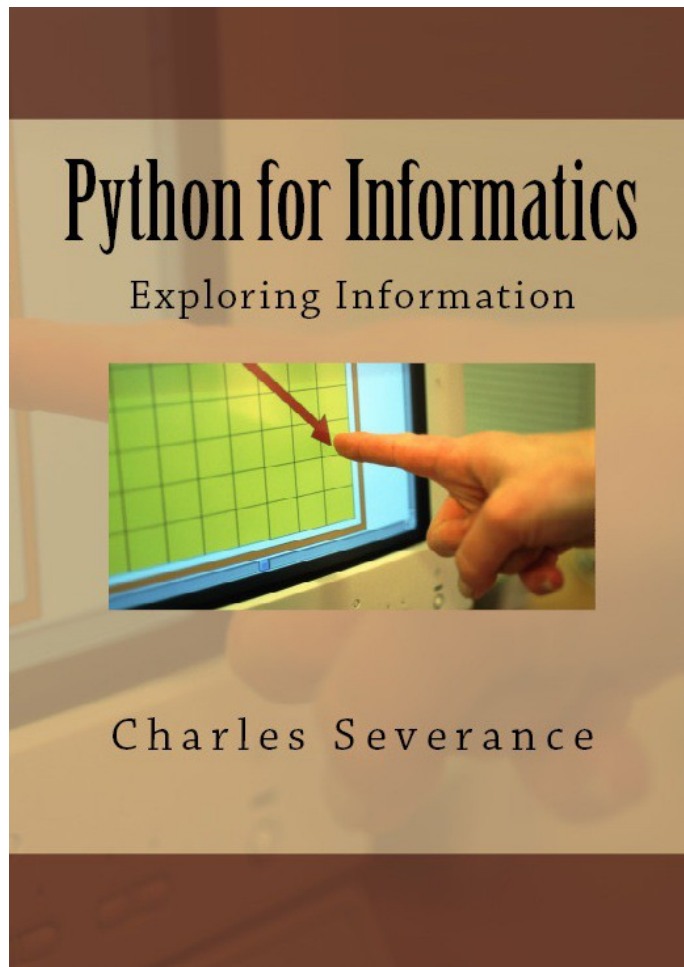
- Dictionary: key-value pairs, mutable, unordered

```
>>> d = {'aap':10, 'noot':5, 'mies':25}
```

- Tuple: immutable, ordered, index is a number

```
>>> t = ('a', 'b', 'c', 'd')
```


Dictionaries and tuples



Chapter 9 and 10 contain more comments on these topics... and exercises

Big data, cloud and the internet of things

Yep, all the buzz words

Big data



- DNA sequencing produces 9 GB data
- Analyzing this takes several days
- Want to speed this up
- Want to track the progress

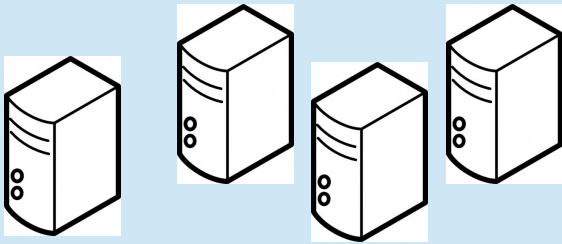
Cloud Computing?



Computers can fly now?

quickmeme.com

Computations and monitoring



Several machines that
do the computations in parallel

Each machine reports its status
RUNNING/FAILED/FINISHED



One machine for monitoring
• Gets status updates from all machines

Monitoring, technical stuff

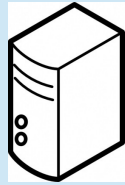
Machines with job

- Runs a python CGIHTTPServer
- set-status.py ip:1.2.3.4 status:RUNNING message:"Doing stuff"
- Presents status of the running job in json format

Monitor

- List with ip-adresses of machine
- Python script gets json file from each machine
- Prints a list with statuses

IoT – monitoring on smartwatch



Machine for monitoring

- Gets status updates from all machines
- Prints all statuses
- **Presents all statuses in a format my watch understands**

On the watch

- An app that “talks” with the monitoring machine
- Blinks and vibrates when all computations are complete



End of this course

