Programming Skills
Lecture 5

Files, dicts

Planning

Date	Remark
May 31	Hand out practice exam
June 7	Hand out take-home exam AND Q&A about practice exam
June 23	Deadline take-home exam
June 30	Written exam

Recap

- Strings
- Lists
- Tuples

Script order

- Import modules

- Build (using functions)
  - From specific to general

- Invoke the functions

Alias vs Clone

```
list1 == list2

list1 is list2
```

Strings (cont’d)

(not in chapters)

Escape characters

```
“this is a double quote: \””

“this is a back slash: \\”
```

Important escape characters

\t	Tab
\n	New line
\r	Return

String formatting

Embedding variables in your string, with nice formatting:

```
formattedString.format(variable)

formattedString.format(variable,
  variable)
```

String formatting

```
“Name: {}, address:
  {}”.format(Aname, anAddress)

“Age = {:1f} years”.format(age)

“face_{0>3d}.jpg”.format(trialnr)

“mean = {data[0]}, sd =
  {data[1]}”.format(data)
```

String formatting

```
“{1} is your {0}?”.format(“quest”,
 “What”)

“I {verb} the {object} off the
 {place}”.format(verb = “took”, object
 = “coffee”, place = “desk”)

“{0}, oh {0}! Wherefore art thou
 {0}?”.format(“Romeo”)

“
```

String formatting (alt)

```
“%s is your %s?” % (“quest”, “What”)
```

```
“pi is %1.2f” % (3.14159)
```

Literature: String formatting cookbook

https://mkaz.tech/python-string-format.html

Handling files

File path

- Files are located in folders/directories
- Find a file using its **path** from the **root**
  - Windows example:

    C:\Program Files\Cool Program\myfile.py
  - Linux/Mac example:

    /Users/Thijs/Desktop/myfile.py

Paths in python

- If you don’t specify a path, python will look for a file in its current working directory

- In Windows: use escape characters in strings – due to the back slash (‘C:\\program files\\’)

- You can always use Linux/Mac path notation

Reading a file

```
myfile = open(‘file.txt’, ‘r’)

# do something here

myfile.close()
```

Reading a file

```
myfile = open(‘file.txt’, ‘r’)

for line in myfile:
    print(line)

myfile.close()
```

Writing a file

```
Myfile = open(‘file.txt’, ‘w’)

for line in somedata:
    myfile.write(line)

myfile.close()
```

Writing a file

```
myfile.write(line)
```

- Line must be a string
- Remember string formatting:

```
line = “{0}, {1},
  {2}”.format(trialnr, response,
  latency)
```

Appending to a file

- Same as writing to a file, but with:

```
myfile = open(‘file.txt’, ‘a’)
```

- If file doesn’t exist, it will create that file.

Dictionaries

Dict

```
{“raphael”: “red”, “leonardo”:
  “blue”}
```

- Key-value pairs
- Look up value with a key: `dict[key]`
- Anything can be a key
- Remove key-value pair with `del`

Dict methods

`.keys()`

Returns a list of the dictionary keys

`.values()`

Returns a list of the dictionary values

`.copy()`

Returns a clone of the dictionary

`.get(key, alt)`

Returns the value of a key, or, if it does not exist, alt

More on dicts

- You can iterate through dicts with `for`

  (will iterate through keys)

- You can check whether key is in dict with:

```
key in dict
```

Recap

- String formatting
- Files
- Dicts

This week’s homework

- Read e-book chapters Files, Dictionaries
- Solve these problems:
  - Ch Files: 1, 2, 5
  - Ch Dicts: 1, 2, Counting Letters, 5
- Bring hard-copy of these problems:
  - Ch Files: 2
  - Ch Dicts: 2, Counting Letters