CISS450: Artificial Intelligence
Lecture 13: Files

Yihsiang Liow

Agenda

- We will study files. The file type is also a built-in type.
- But we will study it in a different way. This will be a DIY lecture.

Your Best Friend

- First go to your best friend
- Oh ... who ***is*** your best friend? (At least when you're doing Python).

Experiment #1

- Try:

```
import requests
s = requests.get("http://news.yahoo.com").text
print(s)
f = open("newsyahoocom.html", "w")
f.write(s)
f.close()
```

Experiment #1

- Look at `open(...,"w")`. What does it do?
- Look at `f.write(...)`. What does it do?
- Look at `f.close()`. What does it do?
- Double-click on the saved file.

Experiment #2

- Using what you've learnt from Experiment #1, go to the Python shell and create a file `a.txt` with string `"line 1\nline 2\nline 3\n"` using Python but do ***not*** close the file yet. View the contents of the file. What do you see?
- Now close the file and view the file again. What do you see?
- Check the you've done it correctly. You'll need this for Experiment #3.

Experiment #3

- Try:

```
f = open("a.txt", "r")
s = f.readline()
print(s)
```

- Notice that the file is opened with `"r"` instead of `"w"`. What does `"r"` and `"w"` mean?
- What is the type of `s`?
- Do a few more `f.readline()`s and print. What happens when you reach the EOF (end-of-file)?

Experiment #4

- Using what you learnt from Experiment #5, write a function that prints all the lines of a file given a filename. Use a `while`-loop.

Experiment #5

- For this part, you need the file `a.txt` from Experiment #3
- Try:

```
f = open("a.txt", "a")
f.write("test")
f.close()
```

- Check `a.txt`. What does `"a"` do?

Experiment #6

- Read the documentation. There is a note which says that a file is its own iterator. Under that note, you're given an example of iterating over the text lines of a file using a `for`-loop
- Recall in Experiment #4, you wrote a function which displays that lines of a file using a `while`-loop. Now write another that uses a `for`-loop instead.

Experiment #7

- Try:

```
f = open("a.txt","r")
s = f.readlines()
```

- What is s? What does `readlines` do?
- What do you think `writelines` would do? Try it out right now.

Experiment #8

- Try:

```
f = open("a.txt","r")
s = f.read(2)
```

- What is s? What does `read(2)` do? In general what does `read(n)` do where n is a positive integer?
- Does the argument of `read` have a default? In other words what if I do `f.read()`?

Experiment #9

- This experiment applies to all objects. One of the most powerful features of Python is that it allows you to “look” into the object during runtime. In particular, you have access to all the attributes and methods of objects
- Try:

```
f = open("a.txt","r")
dir(f)
print(f.name)
```

- What does `dir(f)` do? Do the same for any string.

Coda

- Read Python's documentation on files again
- Experiment with `truncate` and `seek`.
- Let me know if you have questions.