

File I/O

Reading Files

Demo 1

What does the following Processing code do?

```
line_num = 1 # current line of the file being read
f = open("cs_majors.txt", "r")
for line in f:
    cs_majors = line.rstrip("\n")
    cs_majors = cs_majors.split(",")

    num_majors = int(cs_majors.pop(1))
    fill(255, 255, 0)
    ellipse(100*line_num, 150, num_majors, num_majors)

    years = cs_majors.pop(0)
    fill(255, 255, 255)
    text(years, 100*line_num, 50)

    line_num = line_num + 1
f.close()
```

Answer: The program reads data from `cs_majors.txt`, specifically the number of computer science (CS) majors per academic year, where each academic year and its number of majors are stored on a single line. It presents the years as text above circles whose diametres are equal to the number of majors for that academic year.

Exercise 1

Sprites are pre-rendered drawings that represent some object. Suppose we have a file `sprite.txt` containing instructions for drawing a sprite using Processing commands. Each line begins with "fill" or "ellipse" (Processing commands) followed by its arguments on the same line.

Example `sprite.txt` data:

```
fill,100,50,25
ellipse,100,100,50,100
```

Write Processing code which reads the commands and their arguments line-by-line from the tabular file and draws the resulting sprite onto the canvas.

Solution

```
# CMPT 140: File I/O
# Topic(s): Read File

# make canvas dims bigger to see the whole sprite
size(200,200)

# read sprite info from files
f = open("sprite.txt","r")
for line in f:

    # clean up line data
    data = line.rstrip("\n")
    data = data.split(",")

    # set fill colour or draw ellipse based on first list data item
    if data[0] == "fill":
        fill(int(data[1]),int(data[2]),int(data[3]))
    elif data[0] == "ellipse":
        ellipse(int(data[1]),int(data[2]),int(data[3]),int(data[4]))

f.close()
```

This code can be found in `cmpt140-ch15-py/cmpt140_ch15_read_sprite/cmpt140_ch15_read_sprite.pyde`

The tabular `sprite.txt` is a bit unconventional in that each line has a different number of data items based on whether it is a `fill` or `ellipse` call (three or four arguments respectively). Nevertheless, the arguments in every column are all of the same type and delimited by commas; processing the data from the file is a similar procedure regardless of the function called.

Writing Files

Demo 2

Suppose we have a list `dots` representing the integer locations of points in a connect-the-dots drawing. Every two data items in the list represents the (x,y) coordinate for one of the points.

Write Processing code which writes the puzzle's dot coordinates, one (x,y) pairing per line, to tabular file `puzzle1.txt`.

Example `dots`:

```
dots = [ 10,10, 50,50, 50,100, 10,150 ]
```

Expected `puzzle1.txt` data:

```
10,10
50,50
50,100
10,150
```

Solution

```
# CMPT 140: File I/O
# Topic(s): Write File

# example (x,y) locations for connect-the-dots puzzle
dots = [10,10, 50,50, 50,100, 10,150, 150,150, 100,100, 100,50, 150,10]

# write puzzle to file
f = open("puzzle1.txt","w")
for d in range(0,len(dots),2):
    dot_x = str(dots[d])
    dot_y = str(dots[d+1])
    f.write( dot_x + "," + dot_y + "\n")
f.close()
```

This code can be found in `cmpt140-ch15-py/cmpt140_ch15_write_dots/cmpt140_ch15_write_dots.pyde`

The reason we can't use the `join()` method (as described in the readings) directly on a slice of `dots` is because the method requires its arguments be strings (`dots` contains integers).

Exercise 2

Suppose list `markers` contains information for drawing a set of circular map markers. Every four data items in the list represent a single marker's colour (string), its x-coordinate (integer), its y-coordinate (integer), and its radius (integer).

Write Processing code which writes all marker descriptions to tabular file `markers.txt` such that each marker's description is on its own line.

Example `markers`:

```
markers = [ "red",120,400,30, "green",350,350,50 ]
```

Expected `markers.txt` data:

```
"red",120,400,30
"green",350,350,50
```

Solution

```
# CMPT 140: File I/O
# Topic(s): Write File

# example circular markers' (colour,x-coord,y-coord,radius) infos
markers = [ "green",400,50,15, "green",200,200,15,
            "green",350,350,50, "green",260,130,50,
            "red",60,30,15, "red",120,400,30,
            "red",125,132,30, "red",265,225,50 ]

# write every marker's description to its own line
f = open("markers.txt","w")
for m in range(0,len(markers),4):
    str_marker = markers[m]
    for n in range(m+1,m+4):
        str_marker = str_marker + "," + str(markers[n])
    f.write( str_marker+"\n" )
f.close()
```

This code can be found in `cmpt140-ch15-py/cmpt140_ch15_write_markers/cmpt140_ch15_write_markers.pyde`