

CS1 Lecture 23

Mar. 10, 2017

- HW5 due this afternoon, 5pm
- HW4 scores have been posted
- Overall course scores so far and approximate mapping to letter grades

Last time

- HW5 Q1
- Optional/default/keyword arguments
- Ch 19: conditional expressions and list comprehensions, generator expressions, any/all

Today

- HW5 questions/hints
- What are we doing the rest of the semester?
- Simple image processing examples

HW 5 Q1 – last minute suggestions

- I've seen quite a few people have lines like:
 - `line.strip(",;:?!")` not very useful. Why?
 - `line.replace('.', '').replace(',', '').replace(';', '').replace(':', '').replace('?', '').replace('!', '')`
.... (and a bunch more 'replace's for other punctuation)

I don't really see value in this. I recommend the split, then strip approach. Simpler and words fine.
- Test your code little bit by little bit on small file – e.g. 10 sms-es. Many people still write too much code before testing basic things.
 - print the “cleaned” lines (e.g. lines after split/strip)
 - print the dictionaries
 - convince yourself things are correct – analyze the small test file “by hand” – look at it and calculate ...
- Last time I presented default/keyword arguments. I added one to my HW5 Q1 solution – let's me print top 30 words bigger than any specified length

Chapter 19: list comprehensions

Python provides another shorthand- **list comprehensions** – concise expressions for constructing lists.

Consider common pattern:

```
result = []  
for item in someList:  
    result.append(someFunc(item))
```

Can do this in one line with list comprehension:

```
result = [someFunc(item) for item in someList]
```

“apply someFunc to each item in someList and gather the results in a new list”

Note: some of you are forgetting the []'s (which then yields a generator expression, which we discussed a little bit last time)

Chapter 19: list comprehensions

Example related to both Q2 and Q3.

Given matrix of numbers, write an expression that uses a list comprehension and returns True if all elements of the 2nd column are positive.

```
>>> L = [[1, 12, 3, 4, 5], [5, -1, 4, 1, 1], [0, -3, 4, 0, 9]]
```

```
>>> all([row[2] > 0 for row in L])
```

As a more general function:

```
def columnIsPositive(matrix, i):  
    return all([row[i]>0 for row in matrix])
```

After Spring Break

Tentative schedule:

- Ch 15-18: Classes and objects / object oriented programming (1 week)
- Ch 21 (plus material from other sources): running-time analysis, searching, sorting, and other algorithms, randomization (2 weeks)
- GUIs and graphing/charting/visualization (1 week)
- Exam 2
- Accessing web data using public APIs, json, etc. (1-2 weeks)
- Limits of computing – what's hard, what's impossible

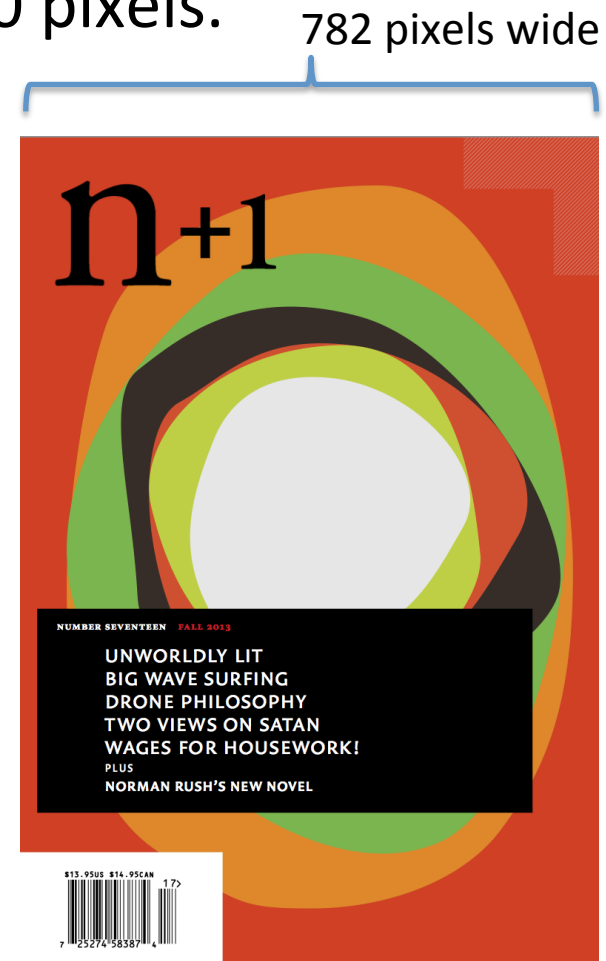
Example: simple image editing

- images are represented as a rectangular matrix of pixels. Inspect properties of an image on your computer and you'll see the dimension expressed as, e.g. 600 x 400 pixels. This means 400 rows of 600 pixels.

▼ More Info:

Dimensions: 782 × 1124
Color space: RGB
Color profile: Color LCD
Alpha channel: Yes
Last opened: Today 10:58 AM

1124
pixels
tall



Simple image editing

- A common way to represent color is using combination of three components: red, green, blue. Called RGB color.
 - Use a tuple, with first component expressing red, second expressing green, third blue.
 - Some systems use 0.0-→1.0 for range. “Standard” or common is integers 0-→255. You’ll hear things like “24 bit color” or “8 bits per channel”
 - In 0-→255 system:
 - (0,0,0) is black
 - (0,0,255) is bright blue
 - (255,255,255) is white
 - (255,255, 0) is combination of “full” red and “full” green, which is ?? (I always forget ...)
 - Easy to find interactive color charts/tools on the web ...
 - Note that there are other color representations/systems – RGB isn’t the only one. CMYK, HSV, ...

Simple image editing

- Not important for this example but something to think about...
- How much storage does it take to represent an image?
- For instance given what I said on previous page, how much memory (bytes/megabytes) does it take to store a 1024 x 1024 RGB image?
- Now look at the properties of some photographs stored on your computer – compare pixel dimensions and disk storage space used. Does the numbers make sense to you? Can you explain the relationship?
- Similarly, how much space might a video – say a 2-hour movie – require? (HD video is 1920 x 1080) When you download a movie, how big is file? Do the numbers make sense?

Simple image editing

- To write a simple image editing program in Python, you only need to understand
 - basics of rgb color
 - basics of image coordinates
 - how to retrieve color values from image - `getpixel`
 - how to set color values in image - `setpixel`

Set pixel at pos=(column,row) on image, with color=(r,g,b).

```
def setpixel(image, pos, color):
```

```
    r,g,b = color
```

```
    column,row = pos
```

```
    image.put("#%02x%02x%02x" % (r,g,b), (column, row))
```

Simple image editing

```
# Return tuple (r, g, b) representing color at pixel at pos=(column,row)
def getpixel(image, pos):
    column, row = pos
    stringvalue = image.get(column,row)
    colortuple = tuple(map(int, stringvalue.split()))
    return(colortuple)
```

- look at `simpleImageManip.py`
 - Easy to make grayscale, negative, flipped, redder images
- for fun, `simpleSteganography.py` –
 - easy to hide secret messages in images

Have a great Spring Break!