CS 150 - Basic Data Types (pt 2)

(Ch 2)

Announcements

- HW due tomorrow
- HW2 will be posted today
- Help session for tutoring next Tuesday at 12:30 in 121 Ritter Hall
Last time: Lists

\[
\text{mylist} = \text{list(1)}
\]
\[
\text{or } \text{the other } = [13]
\]

Methods:
- append
- pop
- insert
- remove
- sort
- reverse
- range
Now:

Note: We lost all our work last time!

→ Need to do a script, like our picture.py

Open a new py file.
Input + Output

In scripts, doesn’t echo back like at python prompt.

Need to use `print`:

```python
print students[0]
print students[1], 'is taking cs/50'
```
Strings
word = str() <
Constructor is str() but this makes a blank string
More often:
words = 'Knock knock'
response = "Who's there?"
statement = "Knock knock In Who's there?"
newline
String functions
Tons of them - p. 56-57
Examples

```python
word = "Hello"
len(word)
word[2]
```

Careful - not lists!

```python
word[0] = 'J'
```
Can slice:

```python
alphabet = 'abcdefg hij
abc = alphabet[0:3]

song = 'Hungry like the wolf'
'y' in song
'ike' in song
Song. index(['i'])
```
Others

'hi' + '!'  
'over' + 'load'

'Goodbye' < 'Hello'

'Hello' == 'hello'

word = 'Hello'
informal = word.lower()
screaming = word.upper()
(print these)
Lists & Strings

request = 'eggs and milk and cheese'
request.split()

request.split('and')
request.split(' and ')

guests = ['Alice', 'Bob', 'Eve']

Conjunction = 'and'
Conjunction.join(guests)
Immutable versus Mutable

A mutable object is one which can be changed after it is created.

A immutable object cannot.

Examples: mutable: lists

immutable: string, ints

\[ z = x + y \] (x + y give unchanged) \[ 7 \neq 3 \]
**Tuples**

An immutable version of lists

skyBlue = (136, 207, 236)

Can't be changes

Functions: any non-mutable ones from the list class

eg: -1 + (12)
Booleans: True & False

'Goodbye' < 'Hello' → True

Operations: and, or, not, or, >=, !=

Ex: a = 2
    b = 3
    c = 5

a < b
a < 3 and b > c

a == b
a != b and b != c
<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>not $x$</th>
<th>$x$ and $y$</th>
<th>$x$ or $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
</tbody>
</table>
Input:
To prompt for user input from a file, use `raw_input`

Ex:
```
print "Hello, what is your name?"
name = raw_input()
print 'Hi, name, !'
```
```
fix () (use + to avoid spaces: name+!)
```
Input

By default, raw_input gets a string:

```python
age = raw_input('How old are you?')
print 'Next year, you will be ', age + 1
```

Solution: cast to an integer

(just like int/float examples)