CS150 - Basic Data Types (pt 2)

Note Title 1/27/2012

(Ch 2)

Announcements

- HW due tomorrow
- HW2 will be posted today
- Help session for turing next Tuesday at 12:30 additional one at 5pm in 121 Ritter Hall

Last time: Lists

```
   mylist = list()
or thelist = []
```

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: .

```
print students
print students [0]
print students[1], 'is taking cs50'
```

Strings

word = str()

Constructor is str(), but this makes a 'blank' string

More often:

```
words = 'Knock knock'
response = "Who's there?"
```

use either

```
statement = "Knock knock\nWho's there?"
```

newline

String functions

Tons of them - p. 56-57

Examples

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

Careful - not lists!

```
word[0] = 'J'
```

Can slice:

```
    alphabet = 'abcdefghij"
    abc = alphabet[0:3]

song = 'Hungry like the wolf"
'y' in song
'ike' in song
song.index('l')
```

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

$z = x + y$

($x + y$ are unchanged)

$1\overset{7}{\cancel{5}}63$

Tuples

An immutable version of lists

not [ ]

skyBlue = (136, 207, 236)

Can't be changes

Functions: any non-mutating ones from the list class

eg
- len
- in
- [ ]

Booleans : True & False

'Goodbye' < 'Hello' → True

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

Ex:

```
a = 2
b = 3
c = 5

a < b
a < 3    and   8 > c

a == b
a != b   and   b != c
```

x	y	not x	x and y	x or y
T	T	F	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	F

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:

```
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)