

Today's "Menu"

(week 11)

- Announcements
- Strings (and a bit more on Lists)
- Functions vs methods
- Recap and evals
- Questions?

07-Apr-10

COMP 480 - Winter 2010

1

Announcements

- Please note: **last day** to submit any assignments Tuesday April 13 by 6pm – **NO late submission due to end of term!** (no T48 either, sorry)
- Exam period next week: Tuesday – Saturday (Friday?) see web page for more information and hints
- Final class session: review & some Object Oriented Programming terminology + concepts

07-Apr-10

COMP 480 - Winter 2010

2

Strings

- Similar to Lists
 - Sequence of characters
 - Can use ' or "
 - Similar access/methods
 - E.g., index (starts with zero), slices, len, etc
- Major difference: **IMMUTABLE**
 - E.g.


```
name = "Pablo"
name = name + " Neruda"
```

 Creates a new string, it doesn't modify it.

07-Apr-10

COMP 480 - Winter 2010

3

Index

- Usually start with **0** and go to **len-1**
- Sometimes we can have negative index values, unusual/error for most languages, but possible in Python.
- **slices**, use values to extract parts of list/sequence/string. Format [s:e] or [s:e:i] – see page 279 for examples.

07-Apr-10

COMP 480 - Winter 2010

4

Functions vs Methods

- **Functions** are “free standing” “chunks of code” .. No `'.'` used to use/call them
- **Methods** are similar, but firmly associated with an object (more on objects later). Use of `'.'` to call them
- Examples of objects and their methods:
 - Image `im.show()` `show` is a method of Image
 - String `name.upper()` `upper` is a string method

07-Apr-10

COMP 480 - Winter 2010

5

Strings

```
strA = "yogurt makes me hungry"
```

```
print strA
print strA[0]      # common
print strA[-1]     # not very common
print strA[2:5]    # not very common
print strA * 3     # not common
```

07-Apr-10

COMP 480 - Winter 2010

6

Strings

```
for ch in strA:  
    print ch  
# how to print this in reverse?
```

07-Apr-10

COMP 480 - Winter 2010

7

String *methods*

- Remember difference between **methods** and **functions**.
- See p. 285/286 Tables 8-1 and 8-2

07-Apr-10

COMP 480 - Winter 2010

8

Lists

- Most languages use 'arrays' – similar, but limited
- Lists can store any type of data within same list
- **MUTABLE!**
- Uses [] to enclose members, [] is the empty list.
- `listA = ['apple', 3.14, "Igel", 12]` # index 0-3 and of length 4

07-Apr-10

COMP 480 - Winter 2010

9

List

- Careful:

```
listA = [55, 6, 12]
listB = listA
listB[0] = 'Peanuts'
print listA
print listB
```

07-Apr-10

COMP 480 - Winter 2010

10

List

- However:

```
listA = [55, 6, 12]
listB = listA[:]
listB[0] = 'Peanuts'
print listA
print listB
```

07-Apr-10

COMP 480 - Winter 2010

11

List

- However:

```
listA = [55, 6, 12]
listB = listA[:]  
listB[0] = 'Peanuts'  
print listA  
print listB
```

07-Apr-10

COMP 480 - Winter 2010

12

List *methods*

- append
- remove
- insert
- sort
- See table 8.3 p 301 for examples. Consult API for more details and additional information.

07-Apr-10

COMP 480 - Winter 2010

13

Summary/Recap

- Questions?

07-Apr-10

COMP 480 - Winter 2010

14