

CSC108H Lecture 10

Dan Zingaro

October 1, 2012

Introducing Strings

- ▶ We've used strings when calling the `print` and `input` functions
- ▶ A string is a sequence of characters
- ▶ To indicate that something is a string, we place single- or double-quotes around it
- ▶ Use `+` for string concatenation, `*` for string repetition
 - ▶ These are examples of **overloaded** operators, since they work differently depending on type of operands
- ▶ What happens if we try to use `+` with a string and a number?
 - ▶ Error: `+` doesn't know what to do!
 - ▶ e.g. is `'3' + 4` supposed to be the string `'34'` or the number 7?

ConceptTest

What is the value of `s` after the following code runs?

```
s = 'abc'  
s = 'd' * 3 + s  
s = s + ' ' * 3  
s = s + 'q'
```

- ▶ A. "abcddd q"
- ▶ B. "abcddd''''''q"
- ▶ C. "abcdddq"
- ▶ D. "qdddabc"
- ▶ E. "dddabcq"

Looping through Strings

- ▶ In Python, we can loop over any type of data that is a sequence
- ▶ A string is a sequence of characters
- ▶ The pattern for looping through a string is as follows

```
for char in s:  
    <do something, using char>
```

- ▶ The loop variable in the example above is `char`; it “steps through” each character in the string `s`

Example 1: How Many Vowels?

Write the following function that returns the number of vowels in string `s`. Both uppercase and lowercase vowels should be counted. (The vowels are a, e, i, o, and u.)

```
def num_vowels(s):
```

Example 2: Removing Spaces

Write the following function that returns a string that is the same as `s` except that spaces are removed.

```
def remove_spaces(s):
```

ConcepTest

```
def mystery(s):  
    new_s = ''  
    for c in s:  
        new_s = c + new_s  
    return new_s
```

What is a good description of this function?

- ▶ A. Return a copy of s
- ▶ B. Return the reverse of s
- ▶ C. Return a string consisting of only the final character of s
- ▶ D. Return a string consisting of only the first character of s

ConceptTest

What is the value of `val` after this code executes?

```
val = 0
for i in 'ab':
    for j in 'cd':
        val += 1
```

- ▶ A. 1
- ▶ B. 2
- ▶ C. 4
- ▶ D. 8
- ▶ E. 16

ConceptTest

What is the value of `val` after this code executes?

```
val = 0
for i in 'abc':
    for j in 'def':
        val += 1
```

- ▶ A. 1
- ▶ B. 3
- ▶ C. 6
- ▶ D. 9
- ▶ E. 27