COMP 141

Strings III

Rhodes College

1

Announcements

Reminders:

- Program 6 - due Sunday (April 5th)
- Midterm 2 – Wednesday, April 8th
  - Practice Problems on Course Website
  - Solutions to Practice Problems on Moodle

2

Accessing Characters Review

Strings are stored character by character.
Each character in a string is numbered by its position:

0	1	2	3	4	5	6	7
"C"	"o"	"m"	"p"	"u"	"t"	"e"	"r"

The numbers shown here above the characters are called ***indices*** (singular: index) or ***positions***.

3

Negative Indices

Negative indexing can be used.
Particularly useful for getting characters near the end of a string.

0	1	2	3	4	5	6	7
-8	-7	-6	-5	-4	-3	-2	-1
"C"	"o"	"m"	"p"	"u"	"t"	"e"	"r"

```
s[2] is the same as s[-6] both refer to "m"

To find last letter in string use: s[-1]
```


4

String Indices

- Two ways to use square brackets
  - 1 number inside -> gives you 1 character of a string
    - s[0] gives you the first character in s
    - If s = "Computer", s[0] gives you 'C'
  - 2 numbers inside (separated by a colon) -> gives you a ***substring*** or string ***slice***

5

5

String Slicing

- **Slice:** span of items taken from a sequence, known as *substring*
  - Slicing format: `string[start : end]`
    - Expression will return a string containing a copy of the characters from `start` up to, but not including, `end`
    - If `start` not specified, 0 is used for start index
    - If `end` not specified, `len(string)` is used for end index
  - Slicing expressions can include a step value and negative indexes relative to end of string

6

6

More String Methods

Table 9-3 Search and replace methods

Method	Description
`endswith(substring)`	The *substring* argument is a string. The method returns true if the string ends with *substring*.
`find(substring)`	The *substring* argument is a string. The method returns the lowest index in the string where *substring* is found. If *substring* is not found, the method returns −1.
`replace(old, new)`	The *old* and *new* arguments are both strings. The method returns a copy of the string with all instances of *old* replaced by *new*.
`startswith(substring)`	The *substring* argument is a string. The method returns true if the string starts with *substring*.

7

7

Using the `find` method

```
def main():
    filename = "First Last_assignsubmission_file_lastname_firstname_prg6.py"
    print(renameFile(filename))

def renameFile(fileName):
    ind = fileName.find("file_")
    fileName = fileName[ind+5:]
    return fileName

main()
```

Output:
`lastname_firstname_prg6.py`

8

8

Testing, Searching, and Manipulating Strings

- **You can use the `in` operator to determine whether one string is contained in another string**
  - General format: *`string1`* `in` *`string2`*
    - *`string1`* and *`string2`* can be string literals or variables referencing strings
- **Similarly you can use the `not in` operator to determine whether one string is not contained in another string**

9

9

In-Class Lab

Lab is on course website

Solutions to some of the problems are in the Box folder – strings3lab.py

10

10