# Python Sequences

*Strings, Lists, and Files*

# Reading: Chapter 5 from Zelle text

# Strings

- String data type
- 1-D array o characters
- Text in between quotation marks
  - E.g. “valid string”
  - Simple quotation also accepted, i.e. ‘valid string’

# String operations

- Input
  - The input() built in function
  - E.g. name = input(“Input your name: “ )
- Output
  - The print() built in function
  - E.g. print(name)
- Indexing
  - Access to any character using integer index
  - Indexing from right using negative indices
  - Indexing returns a string with a single character

# String operations

Python Shell

File Edit Shell Debug Options Windows Help

```
Python 3.2.2 (default, Sep  4 2011, 09:51:08) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> name = input("Input your name: ")
Input your name: Voicu
>>> name
'Voicu'
>>> print(name)
Voicu
>>> name[0]
'V'
>>> name[-1]
'u'
>>>
```

Ln: 13 Col: 4

# String operations

- Slicing
  - Defines substrings within string
  - Consider a string variable called sentence 'CS 17700 is offered in spring 2012'
    - sentence[0:3] is ‘CS ‘
    - sentence[9:] is 'is offered in spring 2012‘
    - sentence[:8] is ‘CS 17700’
- Concatenation
  - ‘123’ + ‘456’ = ‘123456’
- Repetition
  - ‘123’*3 = ‘123123123’

# String operations

| Operator | Meaning |
|---|---|
| + | Concatenation |
| * | Repetition |
| <string>[ ] | Indexing |
| <string>[ : ] | Slicing |
| len(<string>) | Length |
| for <var> in <string> | Iteration through characters |

Table 5.1: Python string operations

# Finding the encoding of a character

- Characters are encoded using 8 bits
- Built in function ord() returns encoding of character
  - the numerical value used to represent it
  - ord(‘A) is 65, ord(‘B) is 66, ord(‘a’) is 97

# iClicker

What is the result of this operation

‘Hono’+’lu’*2+’ is far away’

A. ‘Honolulu is far away’

B. ‘HonoluHonolu is far away’

C. ‘Honolu*2 is far away’

D. All of the above

E. None of the above

# Sequences

- All operations described for strings apply to sequences in general
- This includes arrays of numbers
  - If A is [1, 2, 3, 4, 3, 3, 1], A[1:4] is [2, 3, 4]

# String methods

- Methods are functions specific to a data structure, i.e. an object
- Methods are invoked with a . following the variable name
- Table 5.2 lists strings methods

# String methods

| Function | Meaning |
|---|---|
| `s.capitalize()` | Copy of `s` with only the first character capitalized. |
| `s.center(width)` | Copy of `s` centered in a field of given `width`. |
| `s.count(sub)` | Count the number of occurrences of `sub` in `s`. |
| `s.find(sub)` | Find the first position where `sub` occurs in `s`. |
| `s.join(list)` | Concatenate `list` into a string, using `s` as separator. |
| `s.ljust(width)` | Like `center`, but `s` is left-justified. |
| `s.lower()` | Copy of `s` in all lowercase characters. |
| `s.lstrip()` | Copy of `s` with leading white space removed. |
| `s.replace(oldsub,newsub)` | Replace all occurrences of `oldsub` in `s` with `newsub`. |
| `s.rfind(sub)` | Like `find`, but returns the rightmost position. |
| `s.rjust(width)` | Like `center`, but `s` is right-justified. |
| `s.rstrip()` | Copy of `s` with trailing white space removed. |
| `s.split()` | Split `s` into a list of substrings (see text). |
| `s.title()` | Copy of `s` with first character of each word capitalized. |
| `s.upper()` | Copy of `s` with all characters converted to uppercase. |

Table 5.2: Some string methods

# List methods

- See section 11.2.2
  - append(x), sort(), reverse(), index(x), insert(i, x)
  - count(x), remove(x), pop(i)

| Method | Meaning |
|---|---|
| <list>.append(x) | Add element x to end of list. |
| <list>.sort() | Sort (order) the list. |
| <list>.reverse() | Reverse the list. |
| <list>.index(x) | Returns index of first occurrence of x. |
| <list>.insert(i,x) | Insert x into list at index i. |
| <list>.count(x) | Returns the number of occurrences of x in list. |
| <list>.remove(x) | Deletes the first occurrence of x in list. |
| <list>.pop(i) | Deletes the ith element of the list and returns its value. |

# Files

- Need for large and persistent memory
  - many applications exceed main memory capacity
  - data needs to be "remembered" after computer is turned off
- Text files
  - store text, can be opened with word processor, can be read by humans
- Binary files
  - store data, can only be opened by special programs, cannot be read directly
  - e.g. an image file with extension .jpg, an executable file with extension .exe, etc.

# File Processing

- Create or open, read or write, close
- For text files:
  - read(), readline(), readlines()
  - print() with last parameter the file where to write

# File Processing

Python Shell

File Edit Shell Debug Options Windows Help

```
>>> file = open("CS177/Intro/SortMin.py", "r")
>>> lines = file.readlines()
>>> lines[0]
'# function (i.e. sub-algorithm) for finding index of minimum element\n'
>>> lines[1]
'        # starting at index i0\n'
>>> lines
['# function (i.e. sub-algorithm) for finding index of minimum element\n', '        # starti
ng at index i0\n', 'def MinIndex(A, i0):\n', '    n = len(A)\n', '    imin = i0\n', '    for
i in range(i0+1, n):\n', '        if A[i] < A[imin]:\n', '            imin = i\n', '    retu
rn imin\n', '\n', '# function for sorting array by finding minimum repeatedly\n', 'def SortM
in(A):\n', '    n = len(A)\n', '    for i in range(0, n-1):\n', '\n', '        #find index o
f minimum, starting at i\n', '        imin = MinIndex(A, i)\n', '        \n', '        # swa
p A[i] and A[imin]\n', '        tmp = A[i]\n', '        A[i] = A[imin]\n', '        A[imin]
= tmp\n', '\n', '    return A\n', '\n']
>>> |
```

Ln: 65 Col: 4