

Objectives

- String Review
- String representation: ASCII
- String methods
- Broader Issue: Environmental Monitoring

Oct 5, 2007

Sprenkle - CS111

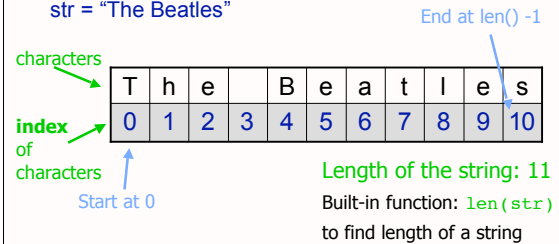
1

Strings

- Actually a *sequence* of characters

➤ Example:

`str = "The Beatles"`



Oct 5, 2007

Sprenkle - CS111

2

Iterating Through a String

- Use a **for** loop to iterate through characters in a string

```
for char in str:  
    print char
```

- Read as "for each character in the string str"

Oct 5, 2007

Sprenkle - CS111

3

Substrings Operator

Literally, **not** optional

- Look at a particular character in the string

➤ Syntax: `string[<integer expression>]`

➤ [Positive values]: index of character

➤ [Negative values]: count backwards from end

- Examples:

➤ `<sequence>[0]` returns the first element/char

➤ `<sequence>[-1]` returns the last element/char

We will deal with sequences beyond strings later.

Oct 5, 2007

Sprenkle - CS111

whilestr.py

4

Substrings Operator

- You can select a substring (zero or more characters) using the `[]` and `:`
- `<sequence>[<start>:<end>]`
 - returns the subsequence from **start** up to and not including **end**
- `<sequence>[<start>:]`
 - returns the subsequence from **start** to the end of the sequence
- `<sequence>[:<end>]`
 - returns the subsequence from the first element up to and not including **end**
- `<sequence>[:]`
 - returns a copy of the entire sequence

Oct 5, 2007

Sprenkle - CS111

5

Checking for substrings

- `search.py`

Oct 5, 2007

Sprenkle - CS111

6

Converting Binary to Decimal

- Accumulator design pattern
- Read in the binary number as a string
 - The starting exponent will be the length of the string-1
- Initialize the result to zero
- For each bit in the binary number
 - Multiply the bit by the appropriate power of 2
 - Add this to the result
 - Reduce the exponent by 1
- Print the result

Oct 5, 2007

Sprenkle - CS111 [binaryToDecimal.py](#) 7

Alternative Solution by Cathy

- Program: `binaryToDecimal2.py`
- Lessons learned
 - Can iterate through string using an integer as the index and the `substring()` operator
 - Developing solutions: if iterating (through whatever) in one direction doesn't work, try the other

Oct 5, 2007

Sprenkle - CS111

8

Converting Decimal to Binary

- Read in the decimal as an integer
- Initialize the result to the empty string
- Repeat until the decimal is 0:
 - `result = str(decimal % 2) + result`
 - `decimal = decimal / 2`
- Print the result

Intuition?

Try out algorithm with 22
Implement algorithm
[decimalToBinary.py](#)

Oct 5, 2007

Sprenkle - CS111 [decimalToBinary.py](#) 9

String Representations

- A string is a sequence of characters
- Each character is stored as a binary number
- **ASCII** (American Standard Code for Information Interchange) is one standard encoding for characters
- One problem with ASCII is that it is based on the English language
 - Cannot represent other types of characters
 - Unicode is a new standard
- Handout: fix second 'Q' and 'R' to lowercase

Oct 5, 2007

Sprenkle - CS111 [ASCII Table Handout](#) 10

ASCII Questions

- Lowercase letters are represented by what range of numbers?
- Uppercase letters are represented by what range of numbers?
- What is the difference between the decimal encoding of 'M' and 'N'?
 - Between 'm' and 'n'?

Oct 5, 2007

Sprenkle - CS111

11

ASCII Questions

- Lowercase letters are represented by what range of numbers?
 - 97--122
- Uppercase letters are represented by what range of numbers?
 - 65--90
- What is the difference between the decimal encoding of 'M' and 'N'?
 - Between 'm' and 'n'?
 - 1

Oct 5, 2007

Sprenkle - CS111

12

Translating to/from ASCII

- Translate a character into its ASCII numeric code using **ord**
 - `ord('a') ==> 97`
- Translate an ASCII numeric code into its character using **chr**
 - `chr(97) ==> 'a'`

Oct 5, 2007

Sprenkle - CS111

[ascii_table.py](#)
[ascii.py](#)

13

Encryption

- Process of encoding information to keep it secure
- One technique: Substitution Cipher
 - Each character in message is replaced by a new character
 - Caesar Cipher
 - Replace with a character X places away
 - Julius Caesar used it to communicate with his generals
 - “Wrap around”
 - Write program(s) to do this in lab

Oct 5, 2007

Sprenkle - CS111

14

String Comparisons

- Same operations as with numbers:
 - `==, !=`
 - `<, <=`
 - `>, >=`

} Alphabetical comparison
- Use in conditions for if statements and while loops

Oct 5, 2007

Sprenkle - CS111

[string_compare.py](#)

15

String Methods

- **Methods**: available operations to perform on strings
 - Slightly different than functions
- Example method: `find(substring)`
 - Finds the index where substring is in string
 - Returns -1 if substring isn't found
- To call a method:
 - `<string>.methodname([arguments])`
 - Example: `filename.find(".py")`

Executed on this string

Oct 5, 2007

Sprenkle - CS111

16

Common String Methods

Method	Operation
<code>center(width)</code>	Returns a copy of string centered within the given number of columns
<code>count(sub[, start [, end]])</code>	Return # of non-overlapping occurrences of substring <code>sub</code> in the string.
<code>endswith(sub), startswith(sub)</code>	Return <code>True</code> iff string ends with/begins with <code>sub</code>
<code>find(sub[, start [, end]])</code>	Return first index where substring <code>sub</code> is found
<code>isalpha(), isdigit(), isspace()</code>	Returns <code>True</code> iff string contains letters/digits/whitespace only
<code>lower(), upper()</code>	Return a copy of string converted to lowercase/uppercase

Oct 5, 2007

Sprenkle - CS111

[string_methods.py](#)

17

String Methods vs. Functions

- Functions: all “input” as arguments/parameters
 - Example: `len` is a built-in function
 - Called as `len(str)`
- Methods: “input” are argument/parameters **and** the string the method was called on
 - Example: `str.upper()`

Oct 5, 2007

Sprenkle - CS111

18

Practice

- Modify binaryToDecimal.py to verify that the entered string contains only numbers
 - How could we make sure that it contains only 0s and 1s?

Oct 5, 2007

Sprenkle - CS111

19

Broader Issues: Environmental Monitoring

- Interdisciplinary projects involving sensor networks
 - Important new-ish CS research area
- Disclaimer:
 - Not a seismologist or a biologist
- Groups

Jennifer
Oliver
Will R

Keith
Maya
Alysen

Cathy
Will L
Matt

Oct 5, 2007

Sprenkle - CS111

20

Discussion

- What are the CS challenges to both projects?
 - Any challenges only applicable to one project?
- How does the environment impact the CS research problems/solutions?
- How did the researchers address these challenges?
 - How would **you** address the challenges?

Oct 5, 2007

Sprenkle - CS111

21

Overview of Challenges: Efficiency

- Some CSists thought that efficiency didn't matter anymore
 - GB of memory, terabytes of storage on machines
- Now: small and embedded devices
 - Need to be efficient!
- Energy in battery powered nodes
- Amount of data stored (when to delete?)
- When, amount of data transferred

Oct 5, 2007

Sprenkle - CS111

22

Overview of Challenges: Reliability

- Data delivery
 - Missing data
 - Connectivity (good signal?)
 - Duplicate data (different sources?)
 - Dead sensor nodes
 - Calibration of data (time synchronization)
- Nodes
 - Withstand extreme weather, conditions
 - Battery life
- Robustness: recover from software failure/malfunction or bad data

Oct 5, 2007

Sprenkle - CS111

23

Overview of Challenges

- Testing
 - Accurately simulate conditions (which will vary widely over long periods of time)
- Different goals from domain scientists
 - CS: push boundaries of sensor networks
 - Example: Improve reliability of data to 95%
 - Seismologists: need 100% reliable data

Oct 5, 2007

Sprenkle - CS111

24

Overview of Solutions: Efficiency

- Energy in battery powered nodes
 - Solar-powered batteries
 - Only transmit if new data
- Amount of data stored (when to delete?)
 - Notify all when data gets to base station
- When, amount of data transferred
 - ZebraNet: only transmit if new data
 - Only transmit if zebra gives data to base
 - Volcano: only when “interesting” data

Oct 5, 2007

Sprenkle - CS111

25

Overview of Solutions: Reliability

- Data delivery
 - Redundancy of data -- verify/validate it is correct
 - Only send to zebras with history of reporting back to base station
- Nodes
 - Weather proofing
 - Batteries: solar-panels to recharge

Oct 5, 2007

Sprenkle - CS111

26

Overview of Solutions: Testing

- Novel simulations!
- Emulate environment/scenarios on computer
- Emulate zebras with horses
- Push software to make sure it “recovers” appropriately from errors or bad information

Oct 5, 2007

Sprenkle - CS111

27